

CP323607

# Reverse Engineering with Imported Data in Fusion 360

Jeff Strater  
Sr Software Engineer, Autodesk, Inc

Phil Eichmiller  
Sr Quality Assurance Engineer, Autodesk, Inc

## Learning Objectives

Following this class, an attendee should:

1. Understand how to reverse engineer designs from a variety of imported data sources
2. Produce reverse engineered designs that are accurate, native, reliable Fusion data
3. Understand how to efficiently reverse engineer with mesh data, organic and analytic/precise/
4. Understand how to work with other imported data (pictures, drawings, etc)

## Description

This class is for you if you are a Fusion user who regularly works with imported data, with the primary goal of reverse engineering a design from that imported data. We will explore some techniques to use data such as meshes, curves, and images from other design systems to create a basis for creating a reliable, native, parametric Fusion design that represents that data as closely as possible.

## Speakers

### Jeff Strater



Jeff is a Senior Software Architect in the Fusion 360 team. I've been with Fusion since the very beginning (before parametrics, before XRef...). My focus is on general modeling/sketching. Before that, I was a developer and architect on Inventor, also before R1. So, I'm a long-time CAD guy. When not working with Fusion or its customers, I like to run, cycle, hike, and read science fiction.

### Phil Eichmiller



Phil is a Senior Software Quality Assurance Engineer for Autodesk, on the Fusion 360 team. In addition to testing and designing with Fusion 360, he enjoys working with the Fusion online community, presenting, and teaching Fusion 360. He used Inventor software as a product designer for 16 years, and he's been teaching Inventor for the CAD program at Portland Community College for the last 10 years. Roller Derby is his favorite pastime, especially watching his daughters, who are both derby stars in Portland, Oregon.

## Table of Contents

<b>REVERSE ENGINEERING WITH IMPORTED DATA IN FUSION 360.....</b>	<b>1</b>
<b>1 INTRODUCTION .....</b>	<b>2</b>
<b>2 REVERSE ENGINEERING PARTS FROM IMPORTED MESH DATA .....</b>	<b>3</b>
2.1 BREP VS MESH DATA .....	3
2.1.1 <i>Organic vs Precise meshes</i> .....	7
2.1.2 <i>Triangle vs Quad meshes</i> .....	9
2.1.3 <i>Mesh models in Fusion</i> .....	9
2.2 WHAT NOT TO DO .....	9
2.3 ORGANIC MESH REVERSE ENGINEERING.....	11
2.3.1 <i>Quad mesh to T-Splines workflow</i> .....	12
2.3.2 <i>Organic triangle mesh to T-Splines workflow</i> .....	16
2.3.3 <i>Manual reverse engineering from organic triangle meshes</i> .....	19
2.4 PRECISE (PRISMATIC/RECTILINEAR) MESH REVERSE ENGINEERING .....	24
2.4.1 <i>Scanned meshes vs “clean” meshes</i> .....	24
2.4.2 <i>“Dirty” mesh models from scan data</i> .....	24
2.4.3 <i>“Clean” models from CAD data</i> .....	34
2.5 REVERSE ENGINEERING TECHNIQUES FOR “PRECISE” TRIANGLE MESHES.....	34
2.5.2 <i>The Fit Curves to Mesh Section command</i> .....	38
2.5.3 <i>Create the parametric features</i> .....	41
2.6 LOOKING TOWARD THE FUTURE.....	46
<b>3 REVERSE ENGINEERING FROM IMAGE DATA .....</b>	<b>48</b>
3.1 INSERT ATTACHED CANVAS .....	50
3.2 CALIBRATE .....	50
3.3 CREATING SKETCHES AND FEATURES TO REVERSE ENGINEER FROM CANVAS DATA.....	52
3.3.1 <i>Creating 2D Sketches</i> .....	52
3.3.2 <i>Creating 3D sketches</i> .....	53
3.3.3 <i>Creating solid geometry</i> .....	56
3.4 SUMMARY OF THIS WORKFLOW.....	61
<b>4 REVERSE ENGINEERING SHEET METAL PARTS.....</b>	<b>61</b>
4.1 SHEET METAL FROM FLAT PATTERN .....	62
4.1.1 <i>Import DXF into Fusion</i> .....	62
4.1.2 <i>Create a Flange</i> .....	63
4.1.3 <i>Create Bends</i> .....	64
4.2 FROM A FOLDED 3D PART .....	67
4.2.1 <i>Import the model</i> .....	67
4.2.2 <i>Convert to parametric design</i> .....	68
4.2.3 <i>Convert to a sheet metal component</i> .....	69

# 1 Introduction

In Mechanical Design, we don't often have the luxury of starting a design completely from a blank sheet of paper, using a single CAD tool for the entire design. Often (most of the time,

actually), we are starting a design as a variation from an existing one, or using parts or whole subsystems from another design. Sometimes, we are trying to retrofit a new subsystem onto a machine that is so old that there are no electronic CAD designs for it, and we may not even have printed drawings for it. Sometimes, we are even trying to build a CAD model of parts of a competitor's product. In fact, the dictionary definition of reverse engineering is this:

## re·verse en·gi·neer·ing

*noun*

noun: **reverse engineering**

the reproduction of another manufacturer's product following detailed examination of its construction or composition.

The input to this process can take a variety of forms. With the decreased cost, increased reliability and usability of 3D scanners, and the development of photogrammetry, it is now practical to do a 3D scan of a part, or take a series of photos from lots of angles and get a mesh model of that part. Perhaps you have old drawings to work from. Maybe just a few photos of the part from the top, side, and front. Sites like [Thiniverse.com](http://Thiniverse.com) also can be a source of (mostly mesh) data to start the reverse engineering process. With the rise of content-sharing sites like [GrabCAD](http://GrabCAD), that are specially oriented around CAD data, you may have a CAD model, but in a different CAD tool, and you want to bring those models into Fusion 360 and modify them there.

If that describes any part of your day-to-day workflow, this class is aimed at you. We will discuss ways to deal with a variety of imported data, with the goal being to rebuild a native version of the design in Fusion 360.

## 2 Reverse engineering parts from imported mesh data

Fusion 360 is not a mesh modeler. There are lots of products on the market, inside and outside of Autodesk that are primarily mesh modelers. This includes:

- 3D Studio Max: [3ds-max](#)
- Maya: [maya](#)
- Meshmixer: [meshmixer](#)
- Blender: [blender](#)
- Etc...

While Fusion does have a mesh editing workspace, its primary body type can be considered to be solid or surface Boundary Representation models, or "BRep" for short. As Lars Christensen says: "All other body types want to become BRep solids when they grow up". Solid BRep models are needed to make drawings, simulate the design, and mostly for CAM, as well.

### 2.1 BRep vs Mesh data

BRep models tend to be very precise, mathematical descriptions of the geometry. They are "infinitely accurate", meaning that, no matter how close you zoom into the model, the description of a cylinder is still a cylinder. Mesh models, on the other hand, tend to be "finite resolution approximations" of model geometry, built of small, usually triangular, facets. Here is an image to show the difference:

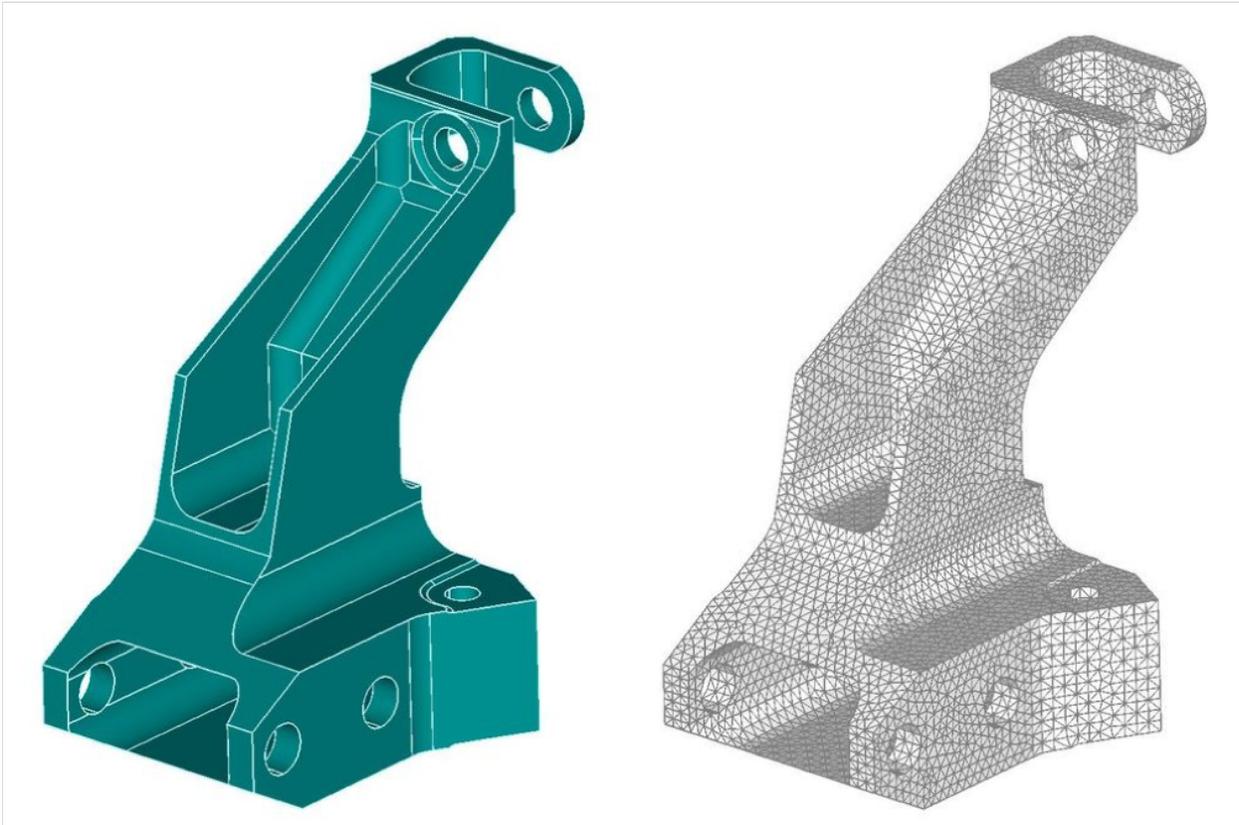


FIGURE 1 - BREP VS. MESH REPRESENTATIONS

These are representations of the same part, but the image on the left is a BRep representation, while the one on the right is a mesh representation. No matter how close you zoom into the BRep representation, it will look like this, with precisely defined edges linear, cylindrical holes and fillet faces, and planar faces where planes exist. However, if you zoom in closely to the model on the right, you will see that what looks like a cylinder is really a series of planar triangles. What looks like a circular edge is really a kind of polygon.

Think about the difference between a PNG and an SVG for 2D graphics. If you zoom in to a PNG image, the closer you get the more pixelated it becomes. The “S” in SVG stands for “scalable”, meaning a line is defined mathematically with two endpoints. With that definition, no matter how closely you zoom into the line, it can be represented precisely.

Here is a PNG image of a bird:



*FIGURE 2 - PNG IMAGE*

And, an SVG version of the same bird:



*FIGURE 3 - SVG IMAGE*

If you zoom in on the PNG version, you get:



FIGURE 4 - ZOOMED IN PNG IMAGE

And on the SVG, you get:



FIGURE 5 - ZOOMED IN SVG IMAGE

The SVG in this analogy corresponds to the BRep – it is infinitely scalable. The PNG corresponds to the PNG – the resolution is fixed.

So, our goal in working with imported mesh data is to be able to build a BRep version of the model that can guarantee this mathematical precision.

**Parametric vs Direct models.** You all are probably aware of the difference between these two design types in Fusion. For the purposes of this class, let's assume that the goal is a fully parametric representation of the design, because that is, arguably, the "most editable" representation. Having sketches you can go back and edit will make the resulting design more amenable to any changes that need to happen. So, we will focus on producing fully parametric, BRep models as our primary goal.

### 2.1.1 Organic vs Precise meshes

There are lots of types of models which are often represented with mesh models. Two of the basic types are, what we will call in this class: “organic” meshes, and “precise” meshes. Examples of organic meshes include: characters in a video game or movie, rounded, highly stylized surfaces such as shampoo bottles, video game controllers, computer mice, or other primarily curvy shapes, like those that appear in nature. Here are some examples of those:

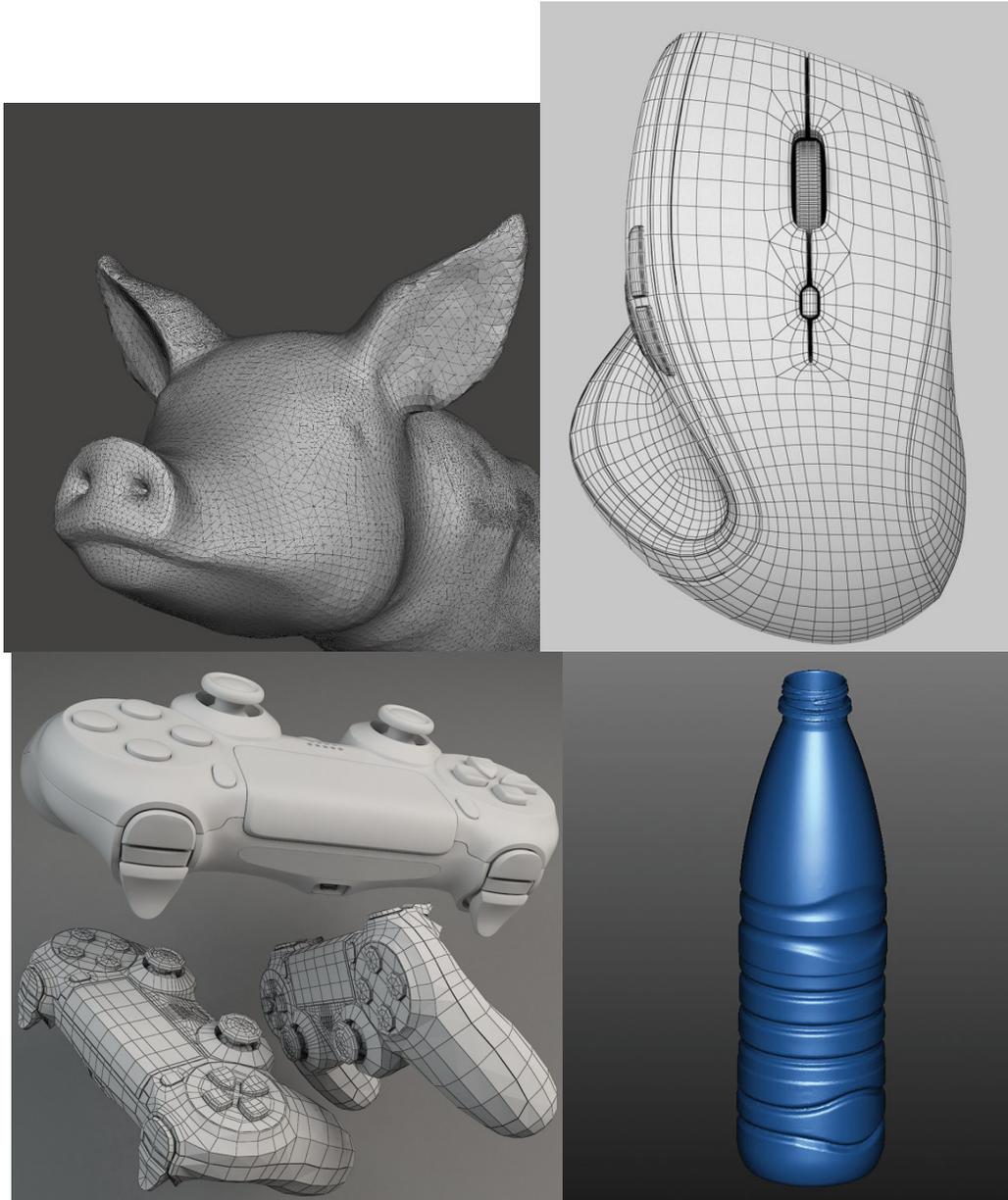
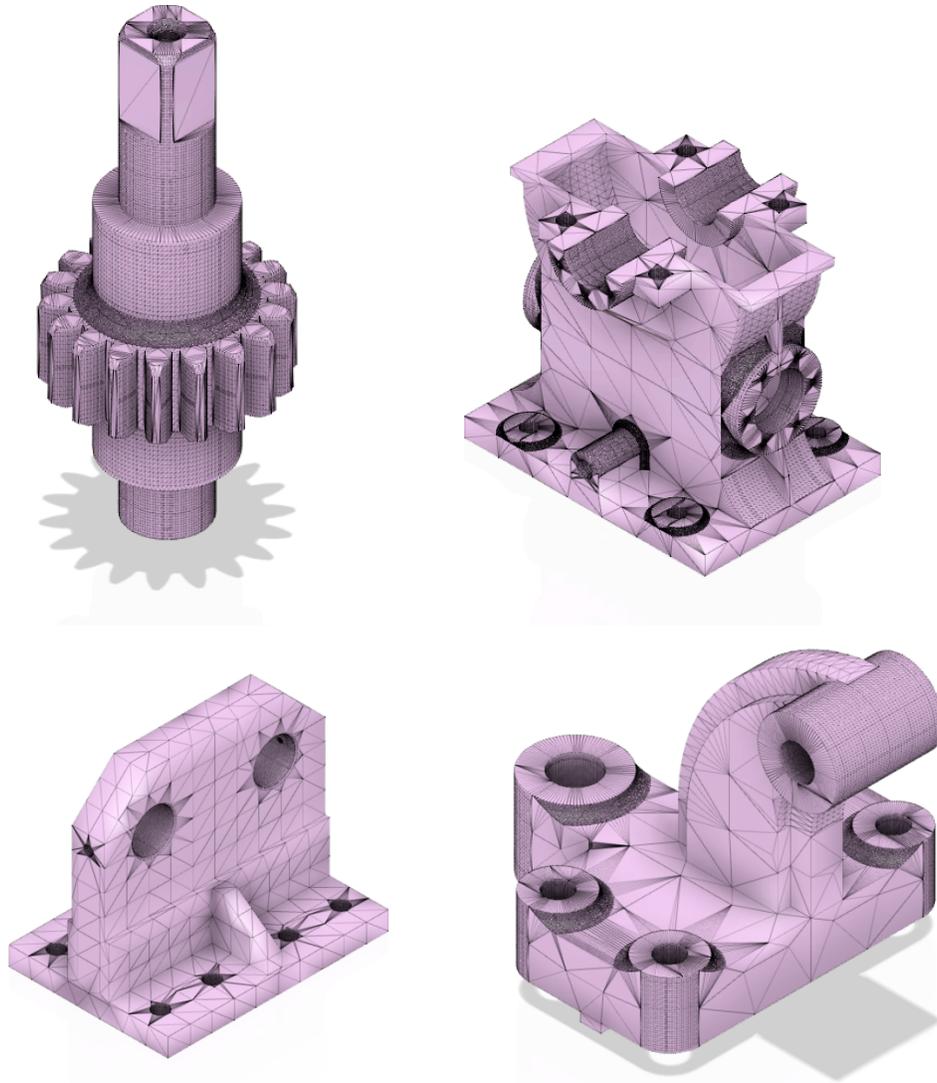


FIGURE 6 - EXAMPLES OF ORGANIC MESHES

The other type of mesh, which for lack of a better term, we will call “precise” meshes. Other possible terms for this class of mesh models include “rectilinear”, “analytic”, “mechanical”, etc. This is a mesh model of shapes commonly used in mechanical designs. These models tend to

have lots of analytic surfaces – planes, cylinders, cones, torii, spheres. Here are some examples of precise meshes:



*FIGURE 7 - EXAMPLES OF PRECISE MESHES*

Precise meshes are often those generated as meshes from a BRep, in some CAD tool. Why do people choose to represent these kinds of models as meshes? Good question. Mesh is a kind of “least common denominator” 3D format. All computer graphics systems use triangle meshes to draw 3D geometry. All 3D printers, also, can consume triangle mesh geometry. Because of this common-format aspect, customers know that almost any system will be able to consume a mesh model. Additionally, because mesh is inherently a “dumb” format, any intellectual property will be hidden in an exported mesh. Finally, the popularity of sites such as [www.thingiverse.com](http://www.thingiverse.com), which use mesh as their native format, have boosted the popularity of mesh as a neutral format.

We’ll cover both kinds of mesh for our discussion. Each has separate strategies that can be used to reverse engineer CAD BRep designs.

### 2.1.2 Triangle vs Quad meshes

There are two primary types of mesh geometry in the industry: Triangle meshes and Quad meshes. The difference in representation is quite simple: This describes the number of sides of each facet in the mesh. A triangle mesh has, well, triangles, with 3 sides. A quad mesh has 4-sided facets. Quad meshes can be easily converted to triangle meshes. Triangle meshes can sometimes be converted, though not so easily, to quad meshes. Quad meshes are often used in sub-division surface systems, such as Fusion's T-Splines.

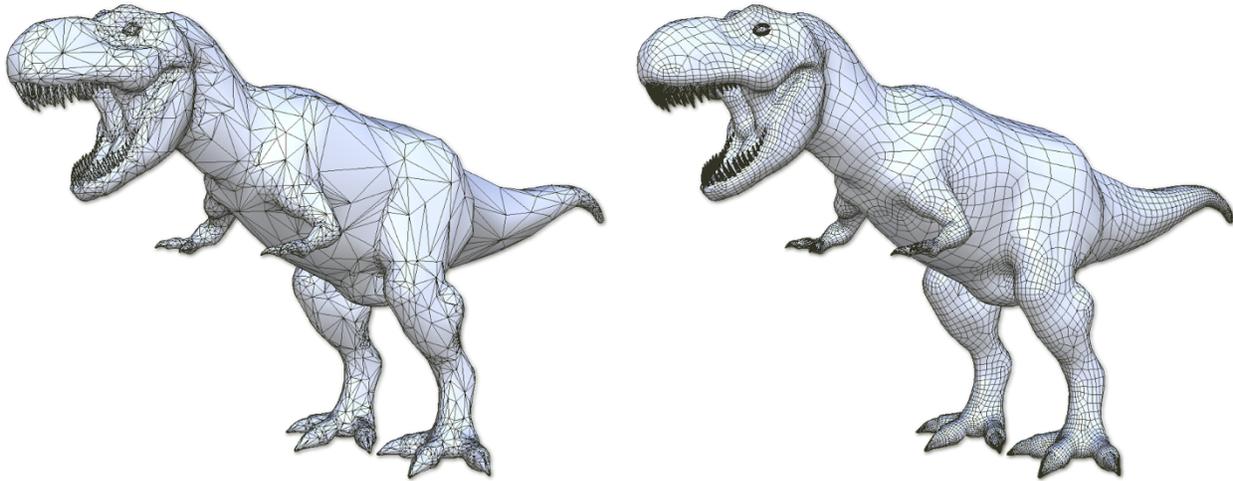


FIGURE 8 - TRIANGLE VS QUAD MESHES

### 2.1.3 Mesh models in Fusion

As we said earlier, Fusion does have a Mesh editing workspace. However, its capabilities are somewhat limited. The good news here is that they are limited to exactly the operations that are needed for this workflow. We designed it with that workflow in mind.

## 2.2 What NOT to do

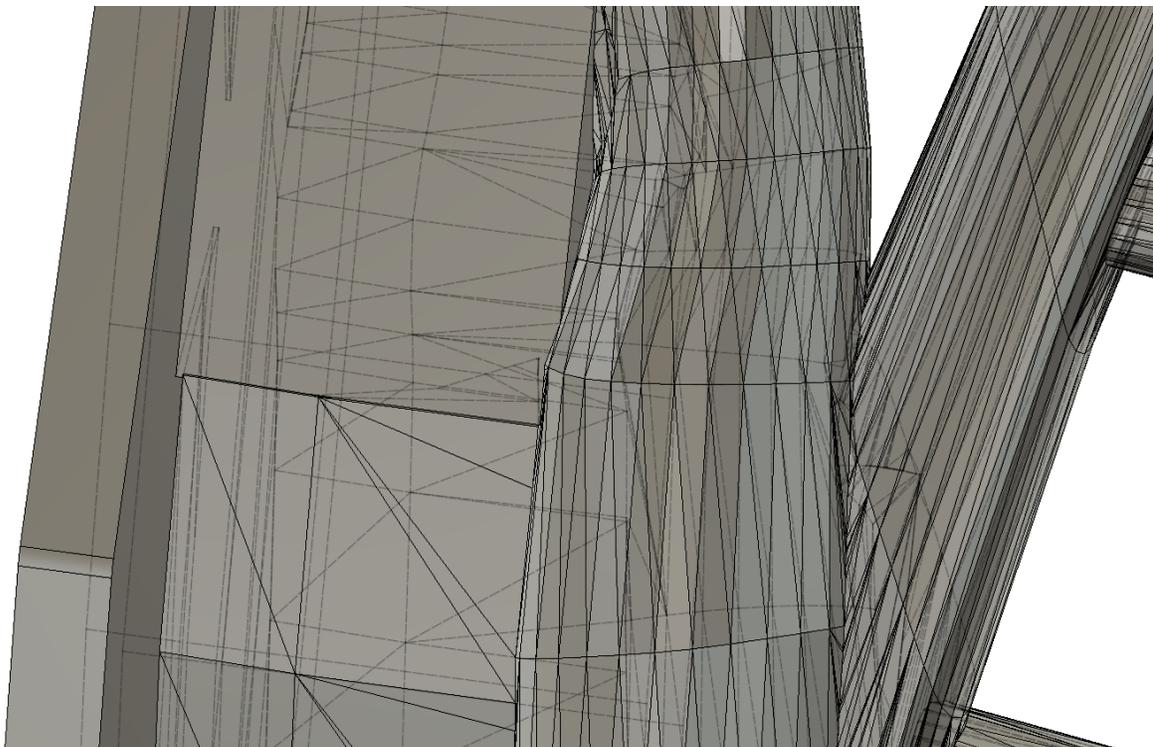
If you are working with mesh data, you might be tempted by a feature which exists in Fusion today: Convert Mesh to BRep:



FIGURE 9 - MESH TO BREP COMMAND

However, don't be tempted by this promising-sounding command name. This is a "use only in case of emergency" command, and here is why: This command is a very simple conversion. It does what its name implies, it does produce a BRep (a solid if the mesh is closed, a surface if the mesh is open). However, what it does is to produce a BRep that has a single face for each mesh facet. Think about that for a minute. That is almost always NOT what you want. The resulting BRep is not good for very much. Planes are not planes, cylinders are not cylinders, etc. Plus, the resulting BRep is extremely heavy. Operations will be slow on this body, will often produce errors, and will likely not produce the results you are looking for, even if they do succeed.

For example, this came from a customer. This customer asks: "Hi, I am trying to develop a 'fillet' between two multi faceted features (at the moment there is a stress concentration in this area and I am looking to remove it). How is the best way to go about this?". This is the model supplied:



*FIGURE 10 - EXAMPLE OF BAD MODEL FROM MESH TO BREP*

It should be pretty obvious that Fusion will not produce a good fillet surface from this garbage input.

Fusion hides this Convert Mesh to BRep command pretty effectively – it is only available in a Direct Modeling design, and only available as a context menu item, not in the tool bar. That reflects some of the concerns around the use of this command. Take its hidden nature as a hint. Like Gary Larson says:

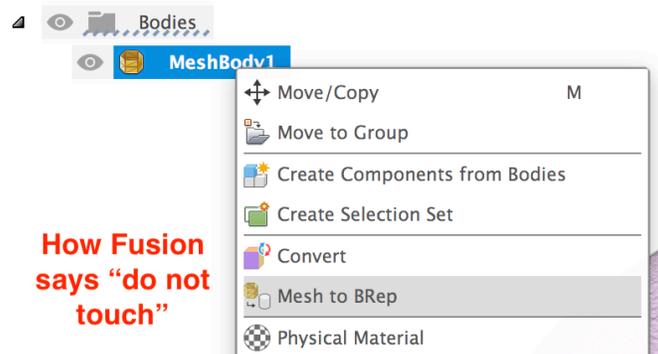
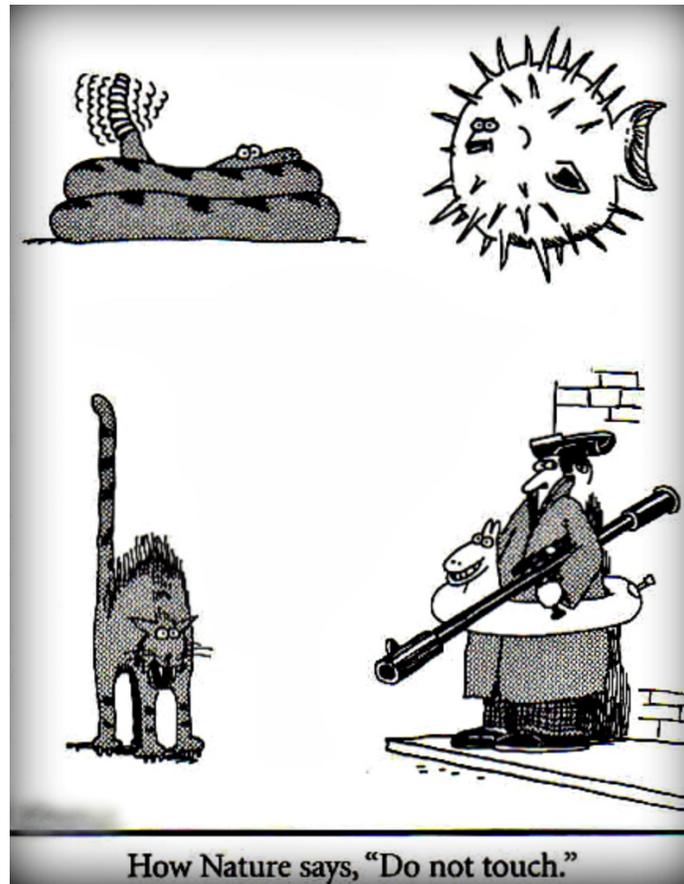


FIGURE 11 - STAY AWAY FROM MESH TO BREP COMMAND!

You can use this approach if you are only trying to do some very basic things with the model: Make a drawing, for instance, because drawings require a solid BRep (although the drawing dimensions will not even be accurate for this type of model, as they will be off by the accuracy of the faceting).

### 2.3 Organic mesh reverse engineering

This workflow is one that Fusion 360, as it sits today, is not particularly good at. The good news is that we're working on that, but can't say too much about it just yet. So, if you want to reverse engineer a model from an organic mesh, there are only a couple of options available:

### 2.3.1 Quad mesh to T-Splines workflow

A Quad mesh is one where each mesh facet has 4 sides. If you are lucky enough to have a quad mesh available to you (not all mesh formats support quad meshes – STL does not, OBJ does), and you have an “organic” model, you can convert those meshes directly into T-Splines (Sculpt) bodies. And, because T-Splines can be directly converted into BRep, you’re home free.

Steps:

- Make sure you have turned off “Triangulate mesh polygons” preference in the Mesh tab:

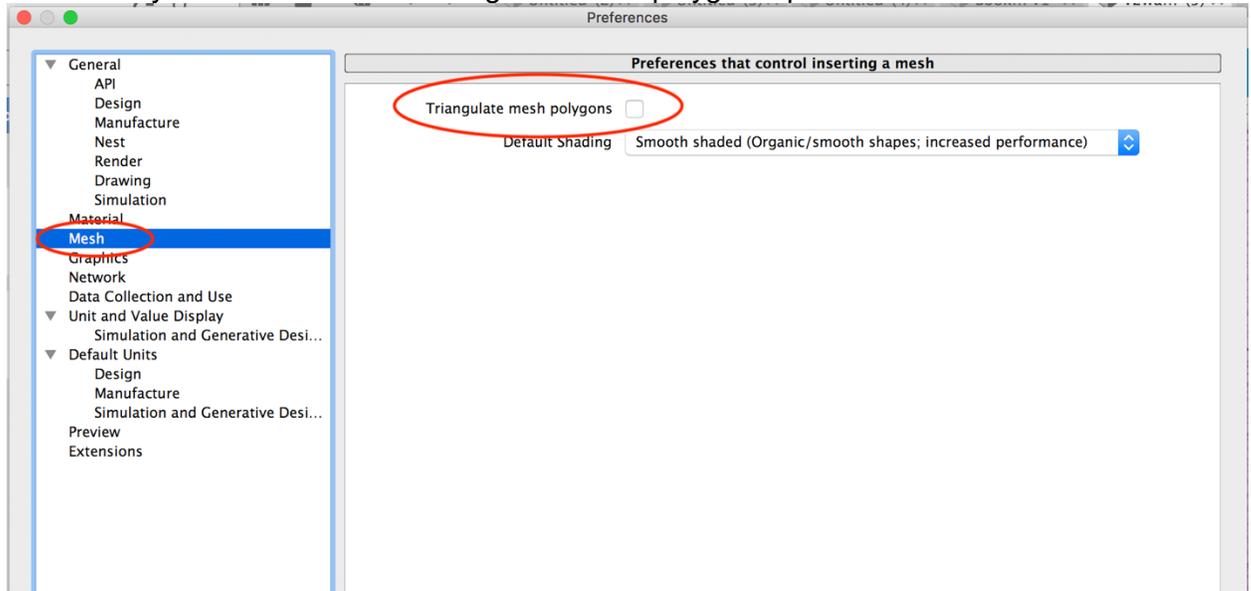


FIGURE 12 - TRIANGULATE MESH OPTION

- This option exists because Fusion’s Mesh workspace can only edit triangular meshes. So, if you are importing a quad mesh to edit, it must be converted to triangles first. But, for this workflow, we do not want triangles
- Import the OBJ or other quad mesh format:

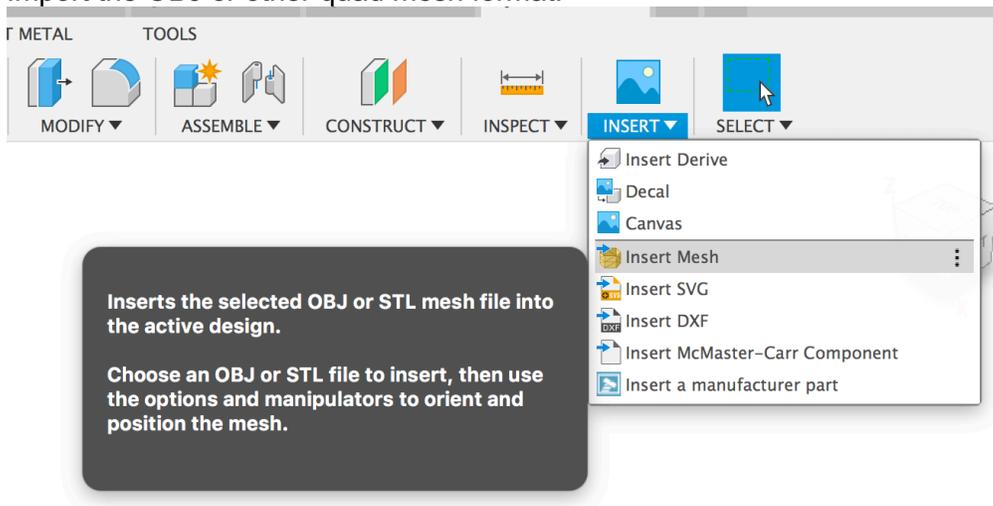


FIGURE 13 - INSERT MESH COMMAND

- select the mesh, position as desired:

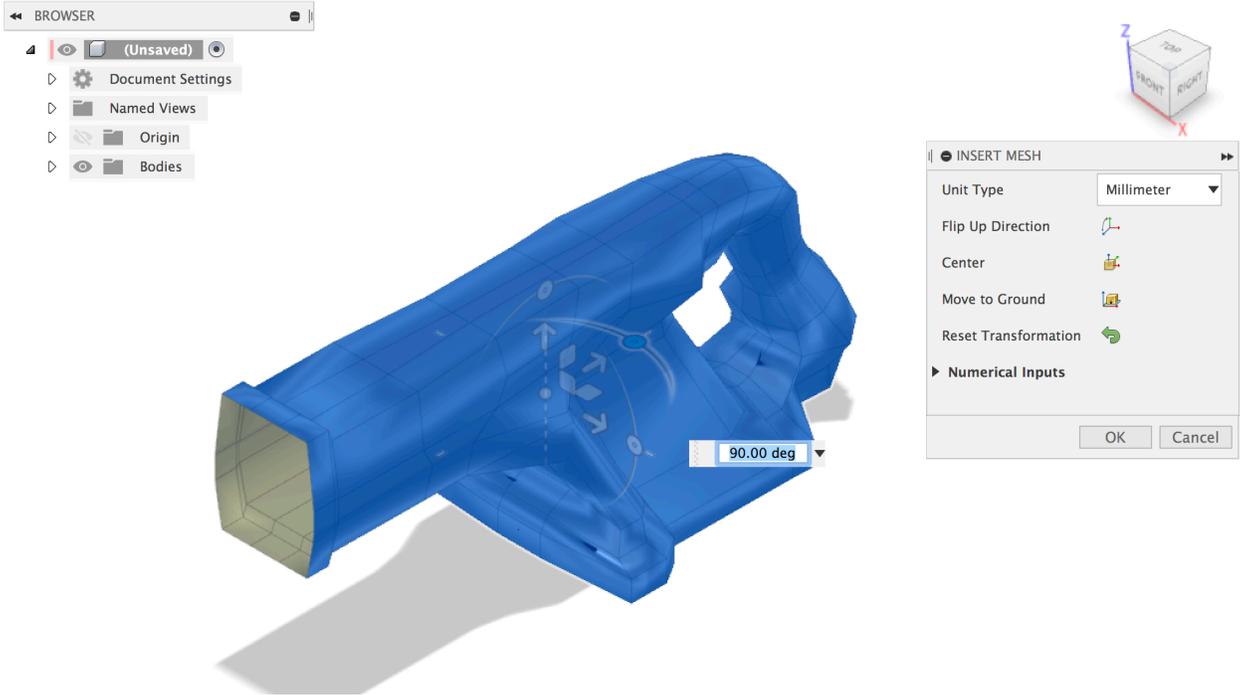


FIGURE 14 - INSERT QUAD MESH

- in the browser, right click on the mesh body, and choose “Convert”:

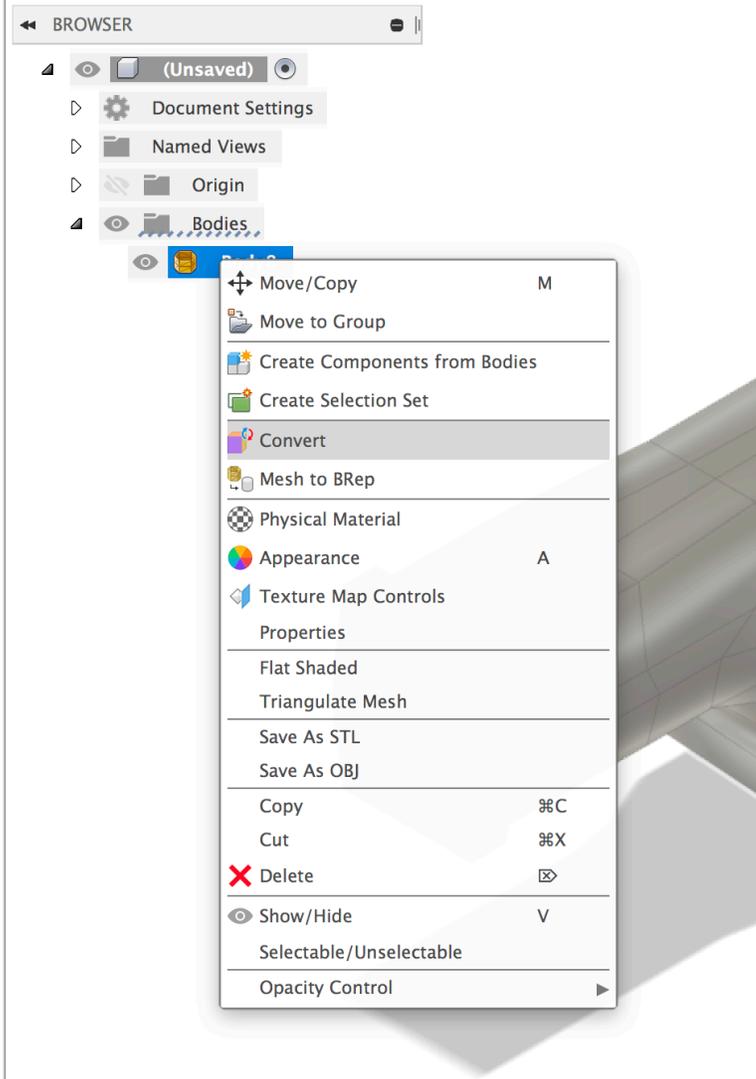


FIGURE 15 - CONVERT MENU

- Choose “Quad Mesh to T-Spline”:

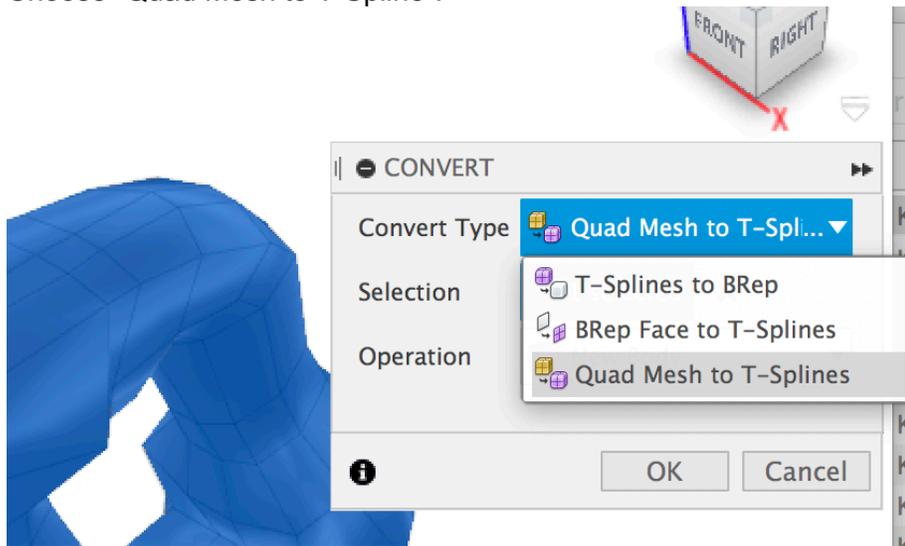


FIGURE 16 - QUAD MESH TO T-SPLINES

- And the result is a T-Spline (Sculpt) body:

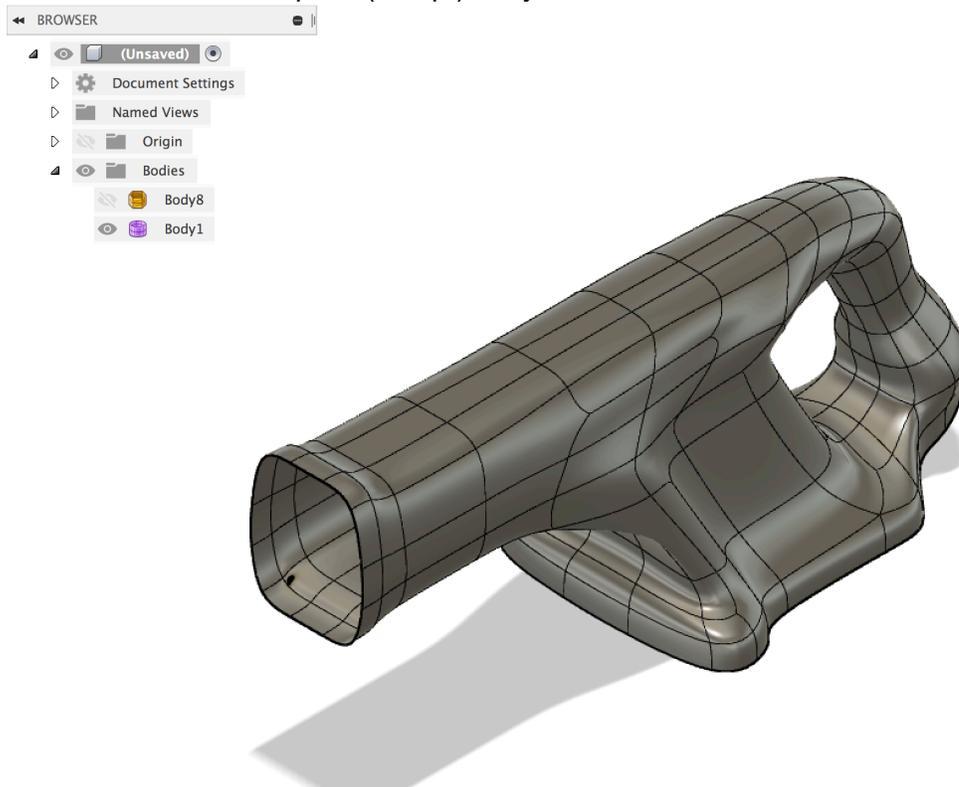


FIGURE 17 - T-SPLINE RESULT

Here is a screencast showing this operation: [Quad mesh to T-Spline](#). Note that this screencast does not contain audio

### 2.3.2 Organic triangle mesh to T-Splines workflow

However, if you only have triangle meshes, then it will be a bit more work. Basically, you will have to convert the triangles into quads, then use the previous method to convert the quads to T-Splines.

Forum post: [mesh-body-plane-cut-problem](#)

Youtube video: [working with organic meshes in F360](#)

Software used:

- Meshmixer (uses the same mesh kernel as Fusion) ([Meshmixer](#))
- InstantMeshes ([instant meshes](#))
- Blender ([Blender](#))

Steps:

1. Import to Meshmixer
2. Clean up in MM - erase/fill, reduce, reorient, etc - optional
3. Export to OBJ
4. Import the OBJ into Instant Meshes for triangle -> quad conversion
  - a. Set target facet count, specify quads
5. Export quad mesh as OBJ
6. Import quad mesh into Fusion
  - a. Have to turn off triangulation in Preferences
7. Convert to TSpline
8. May need Repair Body in Fusion – optional
9. Convert to BRep (exit Sculpt mode)

Here is a GIF showing the critical step here: converting from a triangle mesh to a quad mesh in InstantMeshes

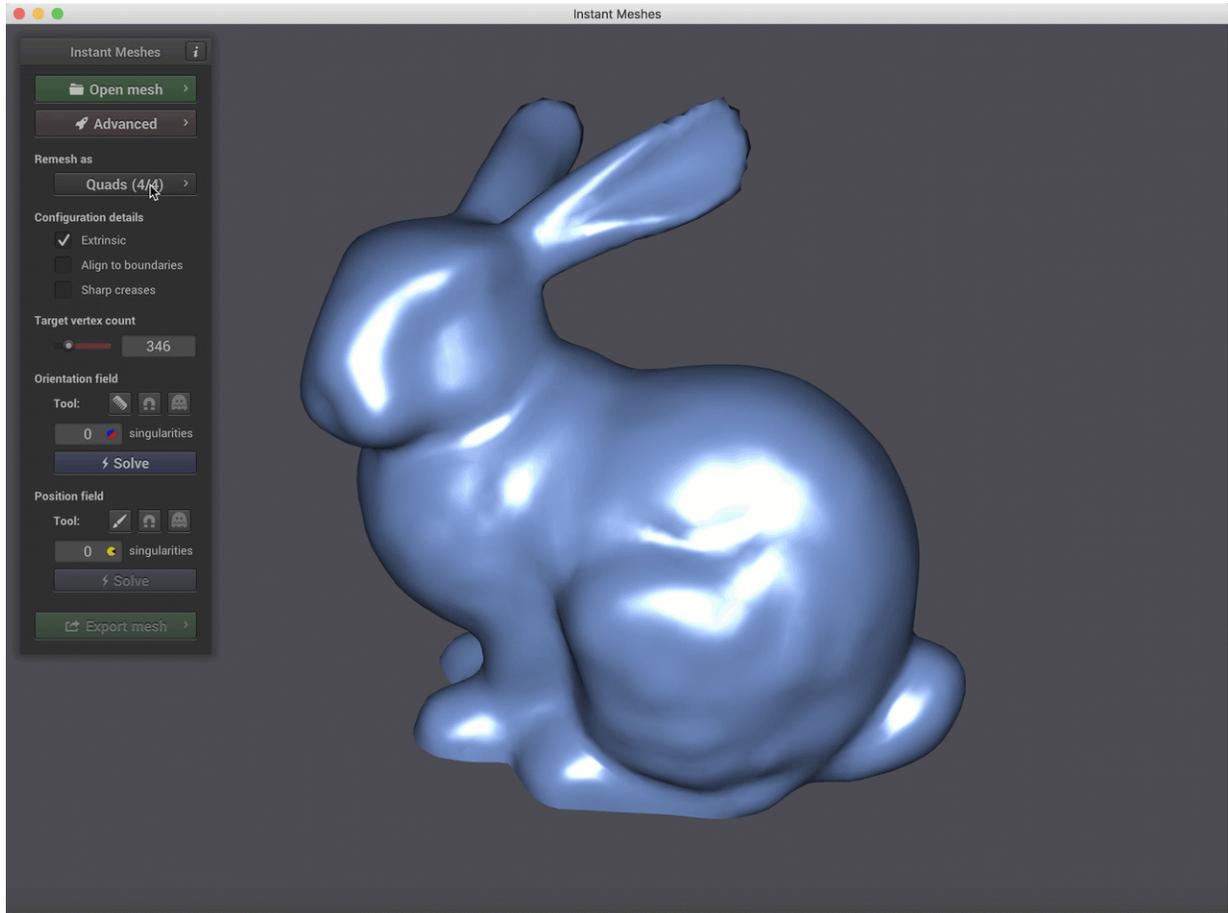


FIGURE 18 – INSTANT MESHES

However, as we hinted earlier, there is hope on this front. The Fusion team has been working hard to add some proprietary Autodesk technology into Fusion that will do all 8 of those steps in one command. The GIF below is just a prototype, but shows the idea:

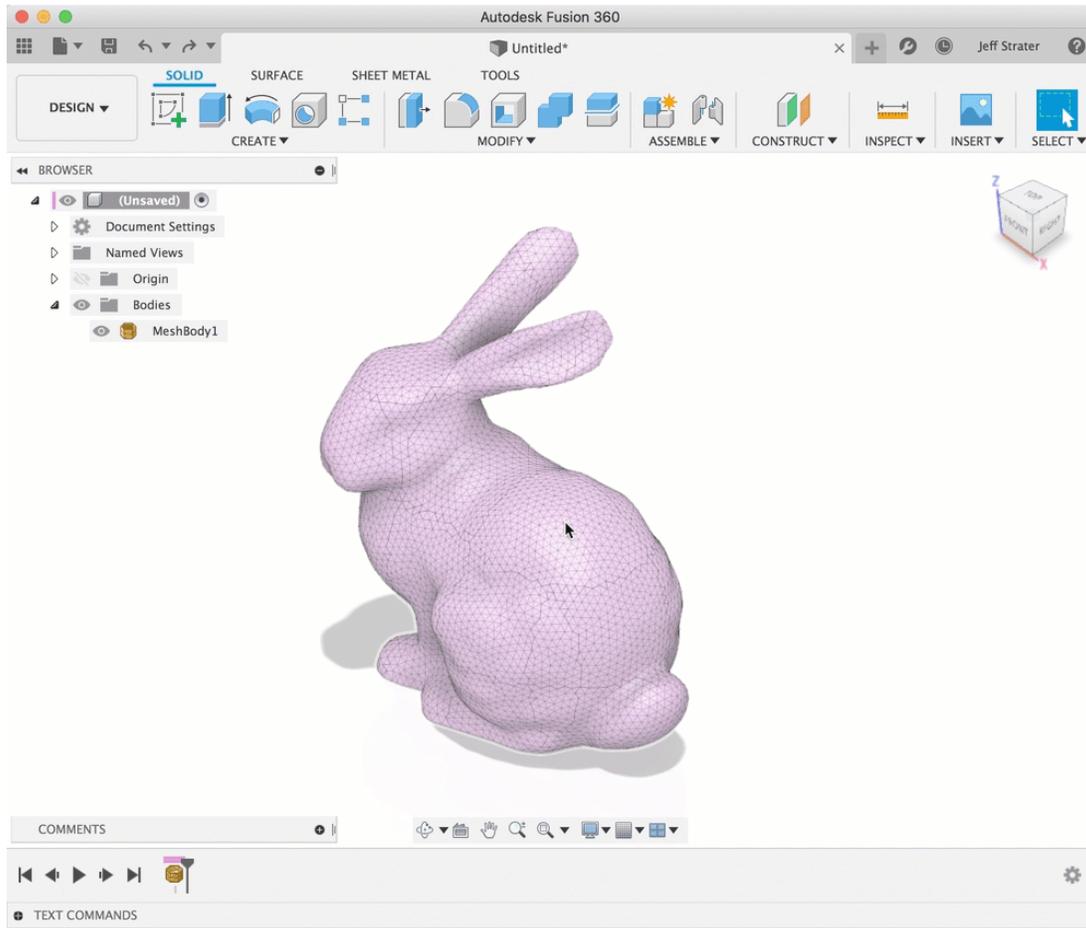


FIGURE 19 - ORGANIC MESH TO BREP IN FUSION

### 2.3.3 Manual reverse engineering from organic triangle meshes

Another useful technique for converting organic meshes into useful BReps, especially if you just need a few surfaces, is the “Pull” method. The Sculpt workspace has the ability to cause T-Spline vertices to be “pulled” to a mesh model. The Edit Form command has a “Object Snap” mode:

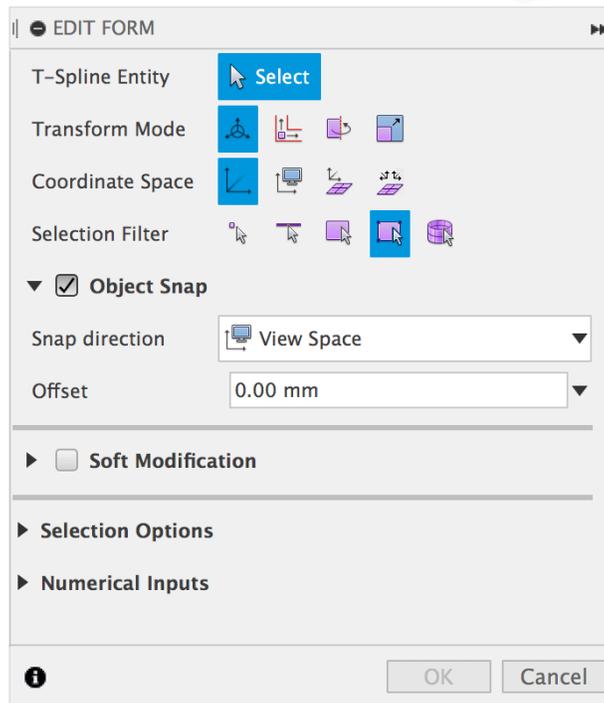


FIGURE 20 - EDIT FORM SHOWING OBJECT SNAP

In Object Snap mode, when you move a T-Spline vertex, it will “snap” to the nearest mesh or BRep model. This can allow you to easily create a T-Spline face or surface which maps to an imported organic mesh. You can choose View direction snap, or World Space direction snap. There is no UI in this command, however, to choose the body to snap to. Fusion will just snap to the closest mesh or BRep body. It’s a bit manual, but if you want fine control, this can be effective.

The second way to use this workflow is the Pull command. This works on the same basic principle, vertices of the T-Spline control frame are snapped to a mesh or BRep body. However, the Pull command operates on an entire T-Spline body at one time. It’s kind of like a vacuum-forming machine for T-Splines. As a result, there is not a lot of precise control over where the Pull snaps each point to. This can produce some hilariously bad results. For this method, it is best to get the T-Spline as close as possible to the desired shape, then use Pull to take it the final step.

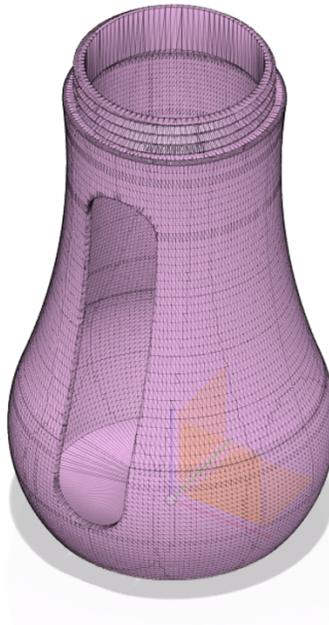
These methods can be used together very effectively. Use Pull to get it close, and Object Snap in Edit Form to allow you to edit the T-Spline while keeping the points on the mesh.

#### Caveats:

- Like all T-Spline operations, Pull or Object Snap is not associative. If you change the mesh, the T-Spline will not follow – you'll have to manually update it
- In the Pull command, you can control whether to snap the control frame vertices or the resulting surfaces to the mesh. In the Object Snap command, the “snap” is only of the control frame vertex. Either way, if you've used T-Splines in Fusion at all, you will know that the resulting T-Spline surface is not guaranteed to be completely aligned with the mesh. You will see the resulting surface sometimes weave in and out of the mesh body. The practical effect of this is that this is not a precise alignment. I find that adding an offset will help get the best result.

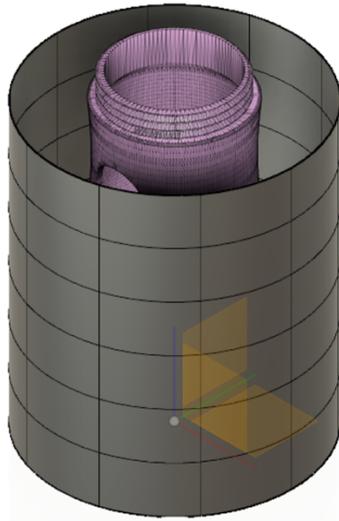
Here is a simple example of this process:

Initial mesh:



*FIGURE 21 - MANUAL T-SPLINE CREATION STARTING POINT*

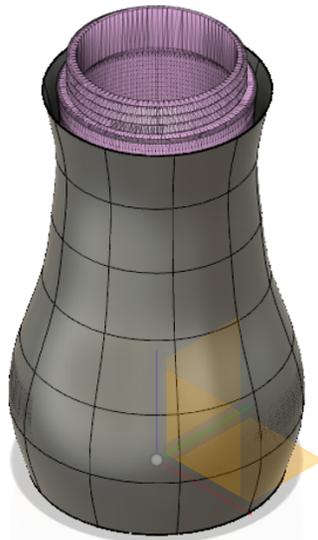
Create a Form feature, and in Sculpt mode, create a cylinder of approximately the same height, and slightly larger than the shape. Add enough length and radial sections for the detail you need:



*FIGURE 22 - MANUAL T-SPLINE CREATION CYLINDER*

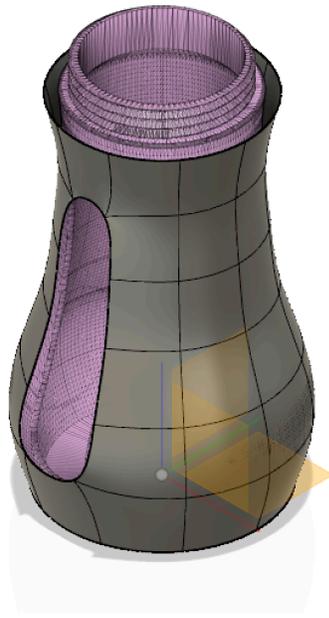
In this case, it is necessary to add enough radial faces so that one set of faces roughly lines up with the open slot in the bottle.

Next, I like to make some rough adjustments to the shape so that it is somewhat close to the shape. I find that Pull works better when the shape is approximately near the mesh shape:



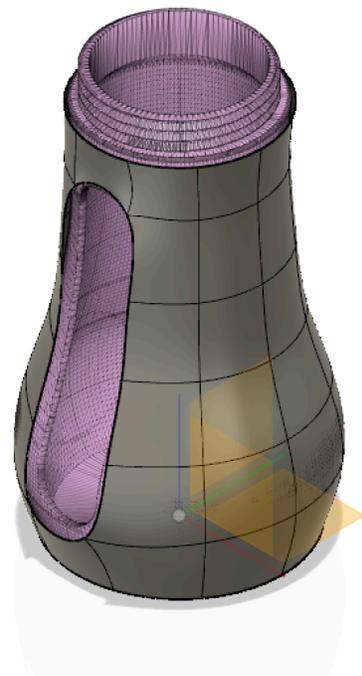
*FIGURE 23 - MANUAL T-SPLINE CREATION ROUGH ADJUSTMENT*

In this case, you need to account for the cutout, so I just deleted a set of faces:



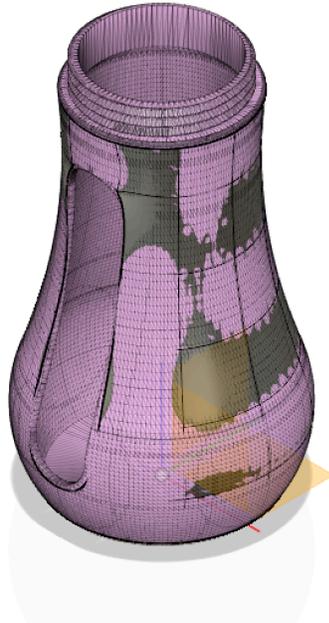
*FIGURE 24 - MANUAL T-SPLINE CREATION CUTOUT*

I made some more tweaks to get the slot close to the mesh slot:



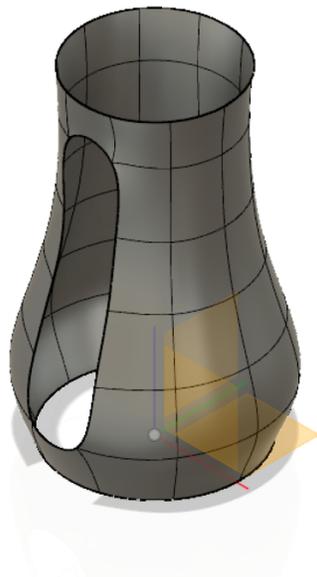
*FIGURE 25 - MANUAL T-SPLINE CREATION FINE ADJUSTMENT*

Then, do the Pull. In this case, I just left settings at their default:



*FIGURE 26 - MANUAL T-SPLINE CREATION PULL RESULTS*

Here is the T-Spline result:



*FIGURE 27 - MANUAL T-SPLINE CREATION TSPLINE RESULTS*

Here is a screencast of this process: [using TSpline Pull command with a mesh body](#). Note: This screencast does not contain audio

## 2.4 Precise (prismatic/rectilinear) mesh reverse engineering

Precise models are the primary focus of this class, since they are inherently more “mechanical” than are organic models, and this type of model is one that is Fusion’s main target market.

### 2.4.1 Scanned meshes vs “clean” meshes

For this workflow, even within the “precise” mesh category, there are variations in the accuracy of the representation of the design.

### 2.4.2 “Dirty” mesh models from scan data

Mesh models derived from a 3D scan, or from photogrammetry, will be very dirty in nature. These mesh models will contain imprecise representations of the surfaces of the model, with lots of deviations from the “true” surface. They will also contain extra geometry that is not in the model. This can be bits of the table on which the scan is done, fasteners in the mesh that you don’t want in the final result, or other glitches introduced by the scanning process), and will often have missing information as well: gaps in the model where the scanner can’t quite see, etc. These meshes will need to be cleaned up before they are usable for reverse engineering.

Here are some examples of mesh cleanup tools that are available in Fusion:

Starting mesh:



*FIGURE 28 - MESH CLEANUP STARTING POINT*

First, we will reduce the mesh. This will help both smooth out the mesh to some extent, but also will make the mesh easier to work with, save faster, etc:

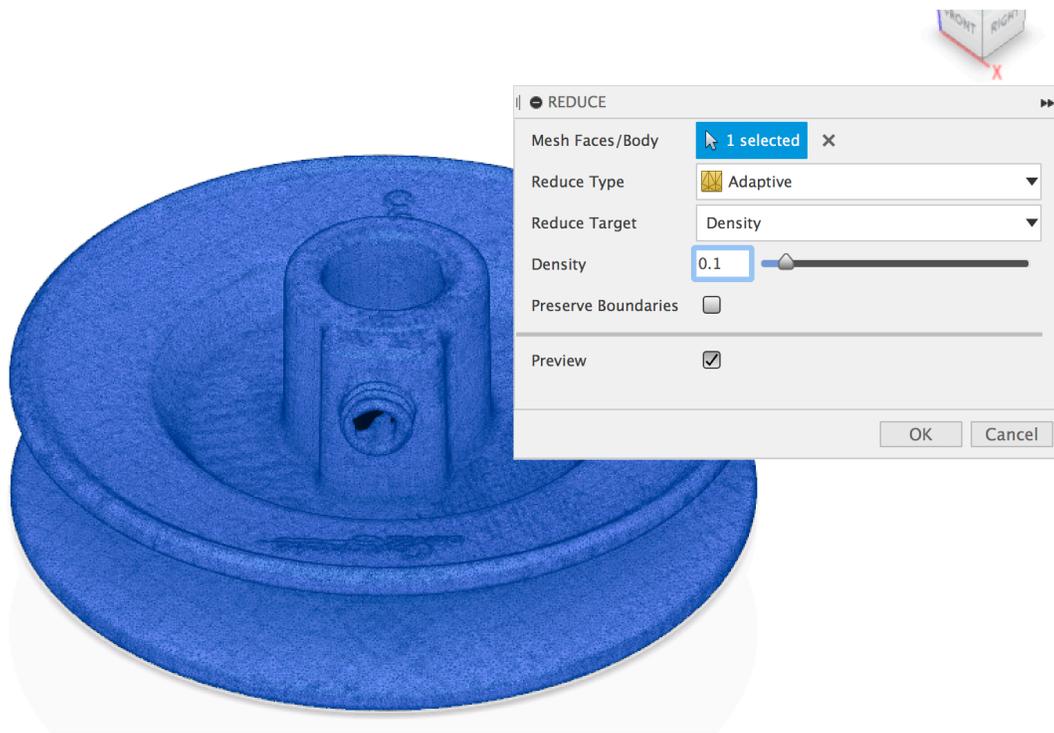


FIGURE 29 - MESH CLEANUP REDUCE DIALOG

And after the Reduce:

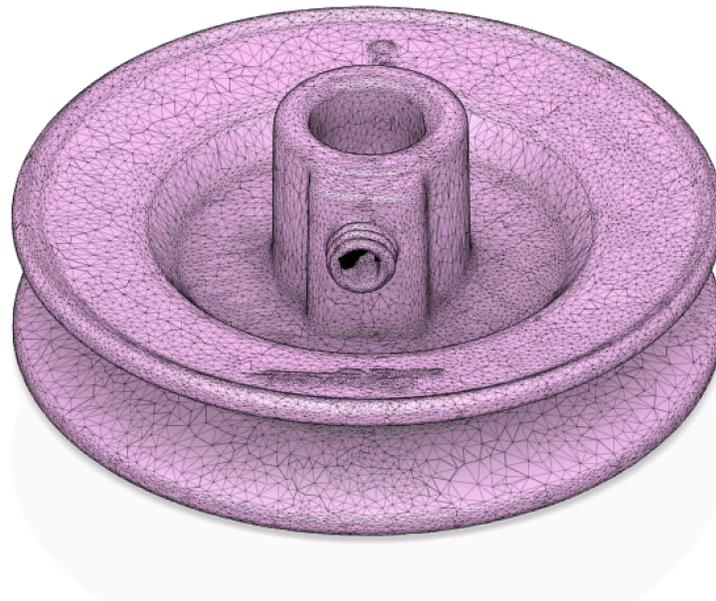


FIGURE 30 - MESH CLEANUP REDUCE RESULTS

Next, we want to remove the embossed text on the front and back of this pulley. First, we want to select the triangles of the emboss and the surrounding triangles, using the Mesh Palette:

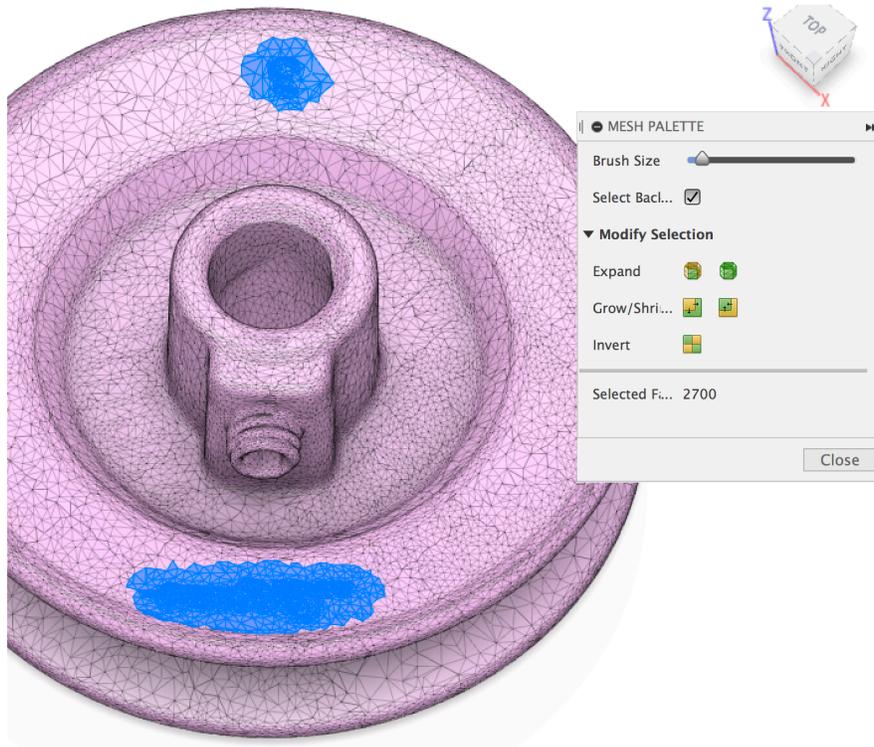


FIGURE 31 - MESH CLEANUP EMOSS SELECTION

First, we'll use the Smooth tool. This tool will just smooth out the raised lettering:

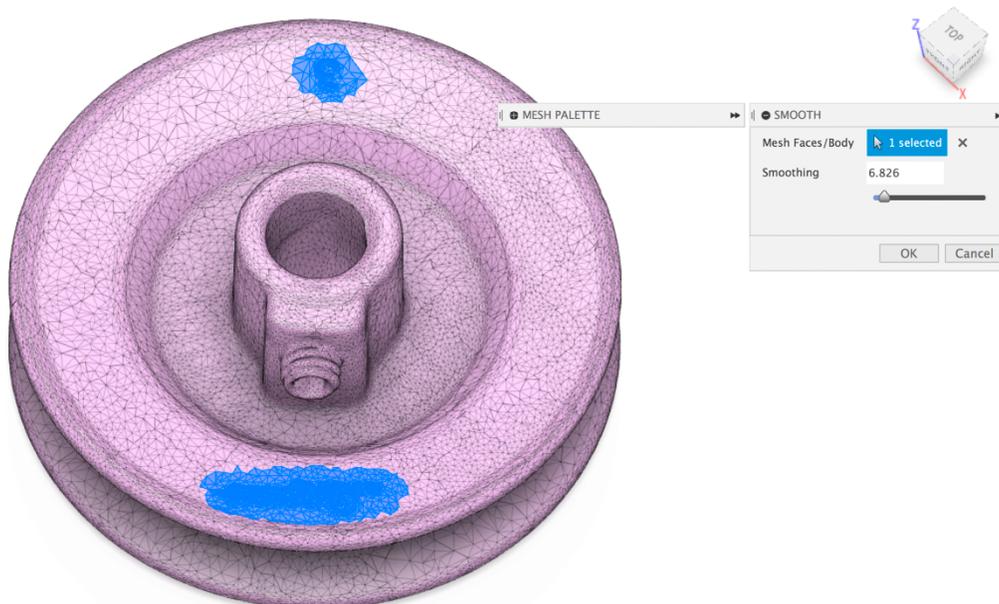


FIGURE 32 - MESH CLEANUP SMOOTH TOOL

After, if you zoom in close, you can see that the text height has been removed:

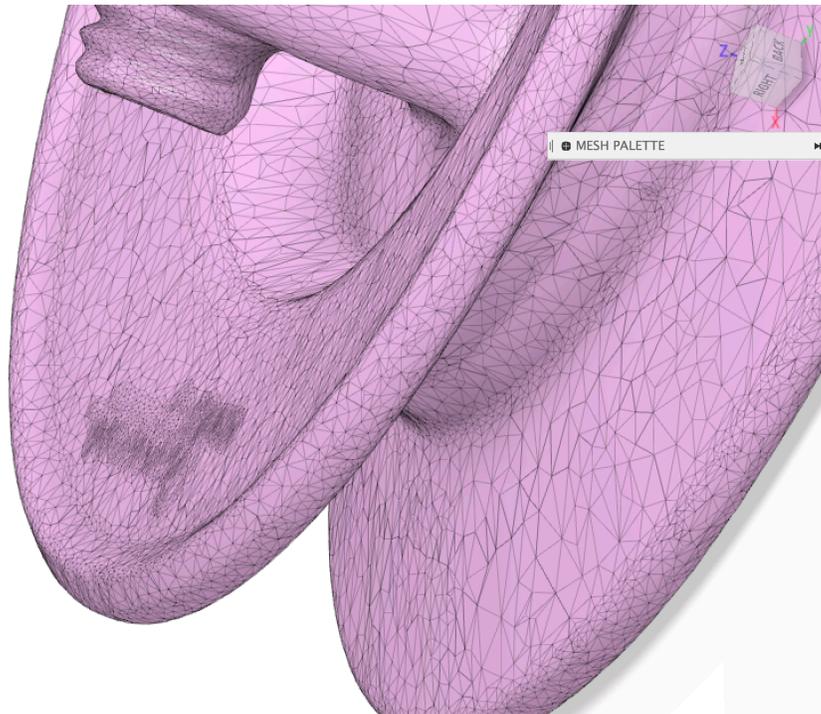


FIGURE 33 - MESH CLEANUP SMOOTH RESULTS

You can see, though, that the triangles from the outline of the text still exist. Another approach is to use the Erase and Fill tool. Again, select the triangles around the text, start the Erase and Fill tool, and set the inputs to achieve your goals:

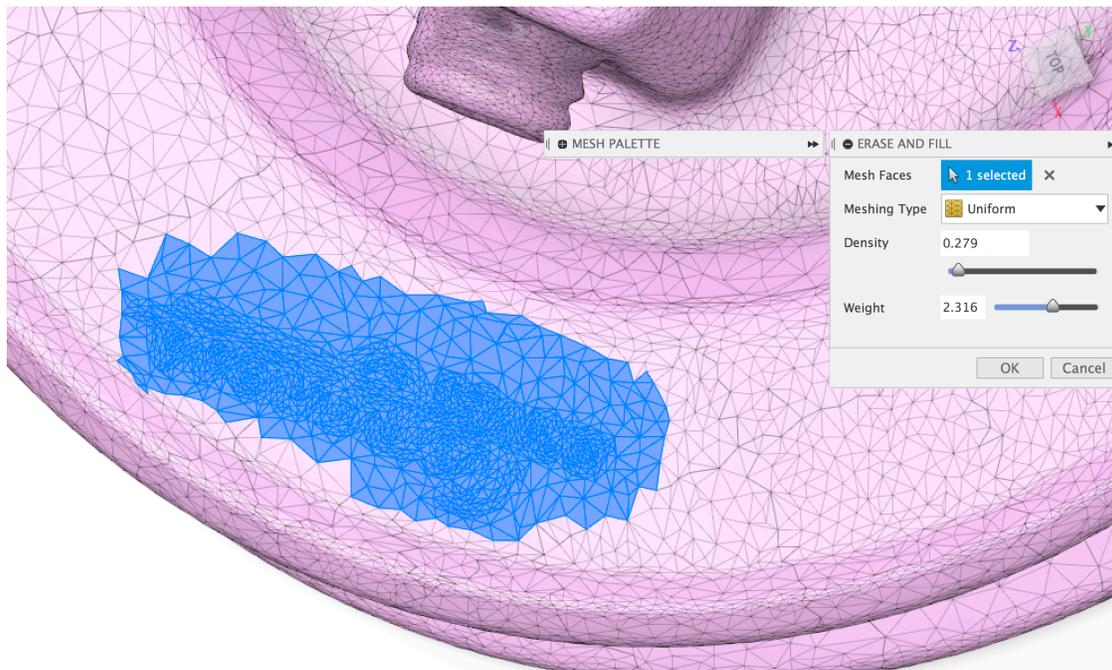
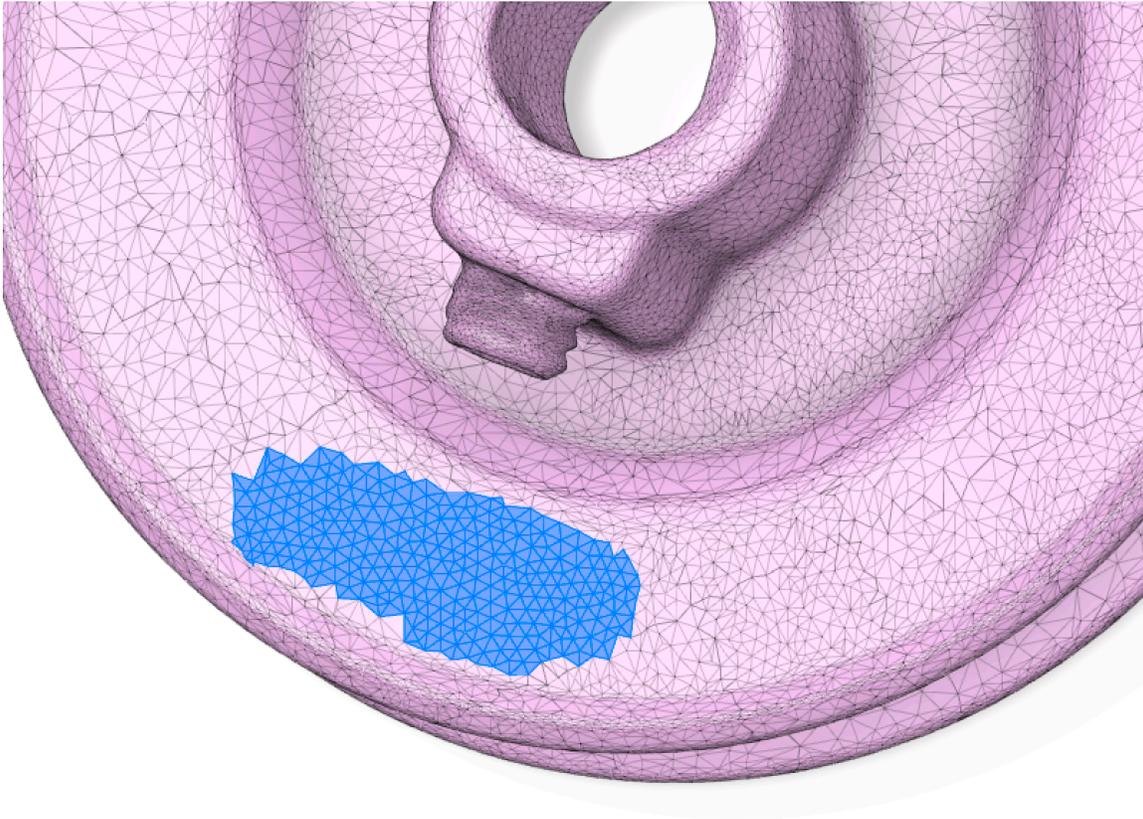


FIGURE 34 - MESH CLEANUP ERASE AND FILL

Here is the result:



*FIGURE 35 - MESH CLEANUP ERASE AND FILL RESULTS*

Use whichever tool gets you the results that most closely match your goals.

Next, we'll delete this part of the mesh:

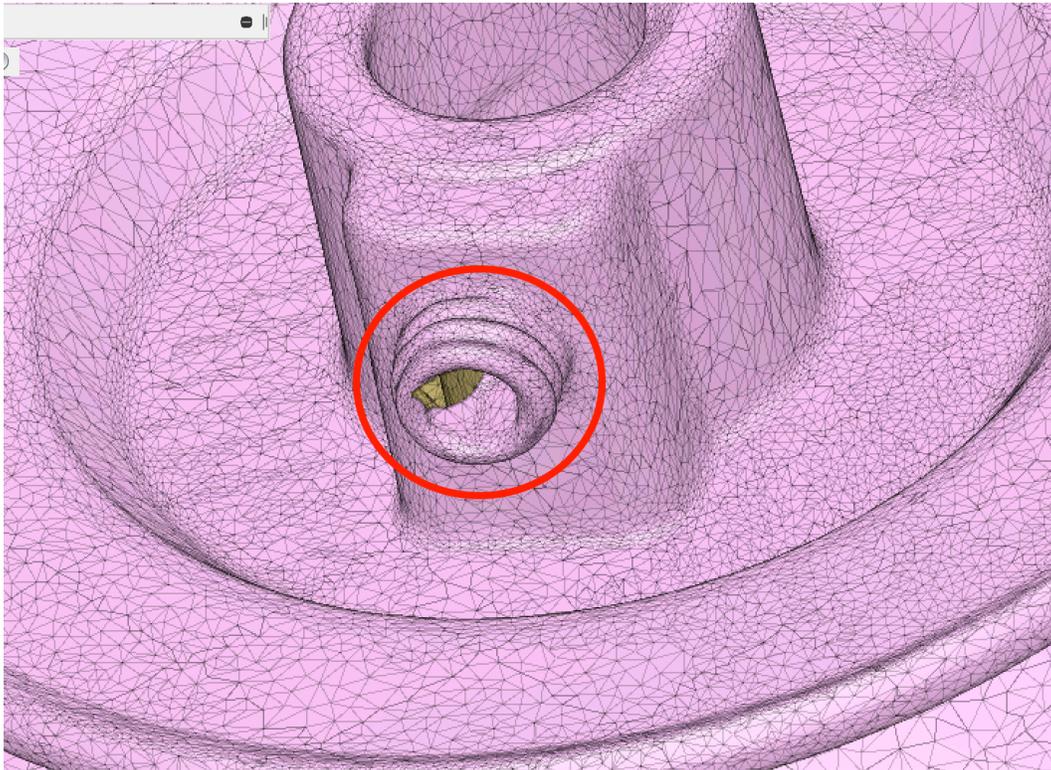


FIGURE 36 - MESH CLEANUP GEOMETRY TO REMOVE

This is clearly a set screw that should be a separate part, but was left in for the scan. So, we want to remove it before proceeding. There are a couple of approaches here. If you are really patient, you can select just the facets from this part of the model. Some tricks to know:

- Use the “Brush size” setting in the Mesh panel to control the size of the selection tool
- Fusion mesh selections (unlike other selections in Fusion) are transacted. You can undo/redo selections. This is very handy if you accidentally select too many facets
- Shift + drag will add to the selection
- Option (Mac)/ Alt (Windows) + drag will de-select facets
- Ctrl (both) + drag will toggle selection

This is as close as I got to the right selection set:

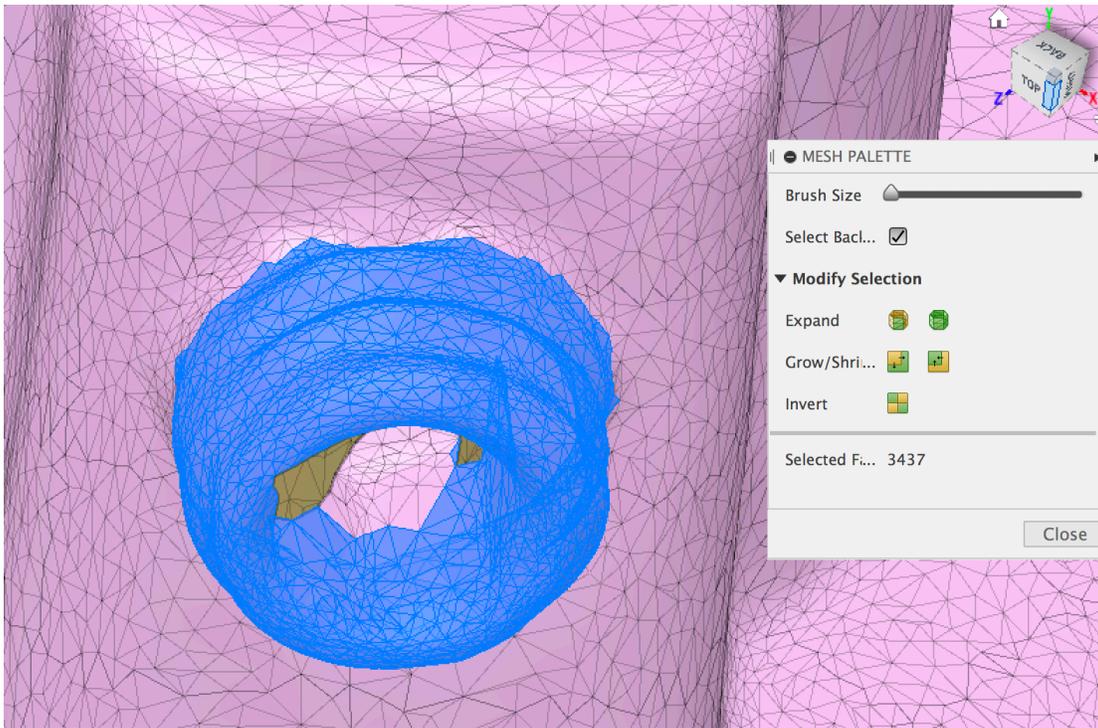


FIGURE 37 - MESH CLEANUP FACET SELECTION

Then, you can just delete these facets:

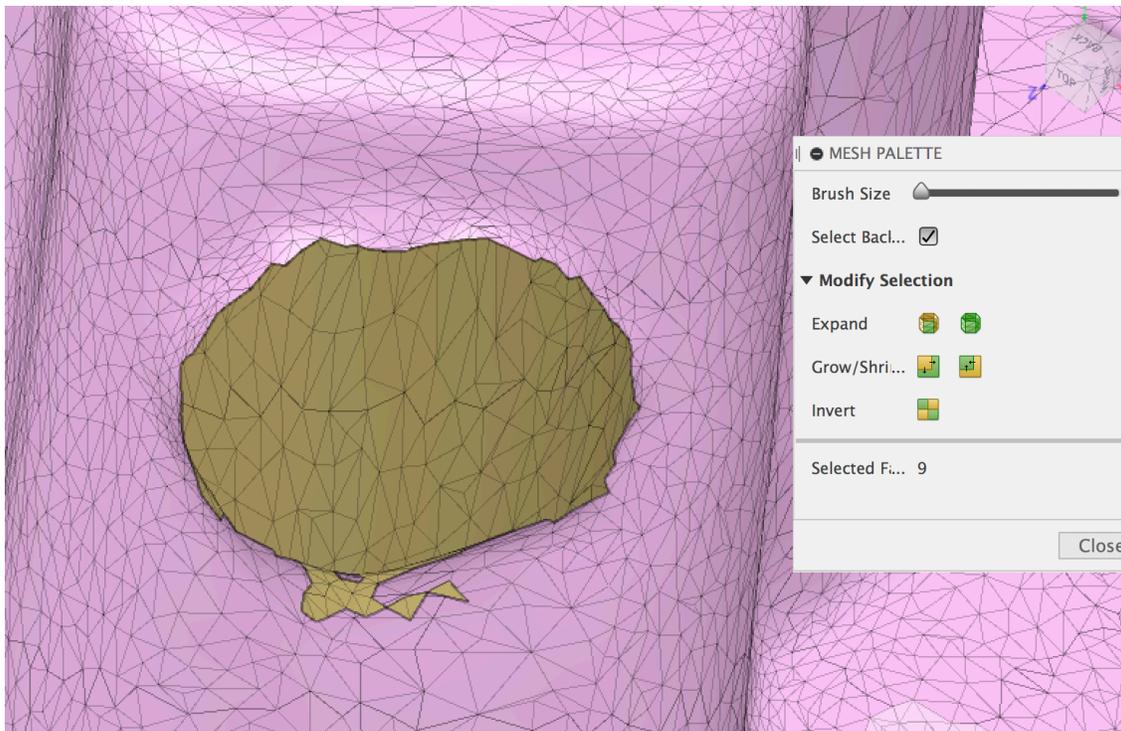


FIGURE 38 - MESH CLEANUP DELETE FACETS

You can see that I've selected a few too many facets. Erase and Fill can fix those. One trick: double click with the cursor over a boundary will select the whole open loop:

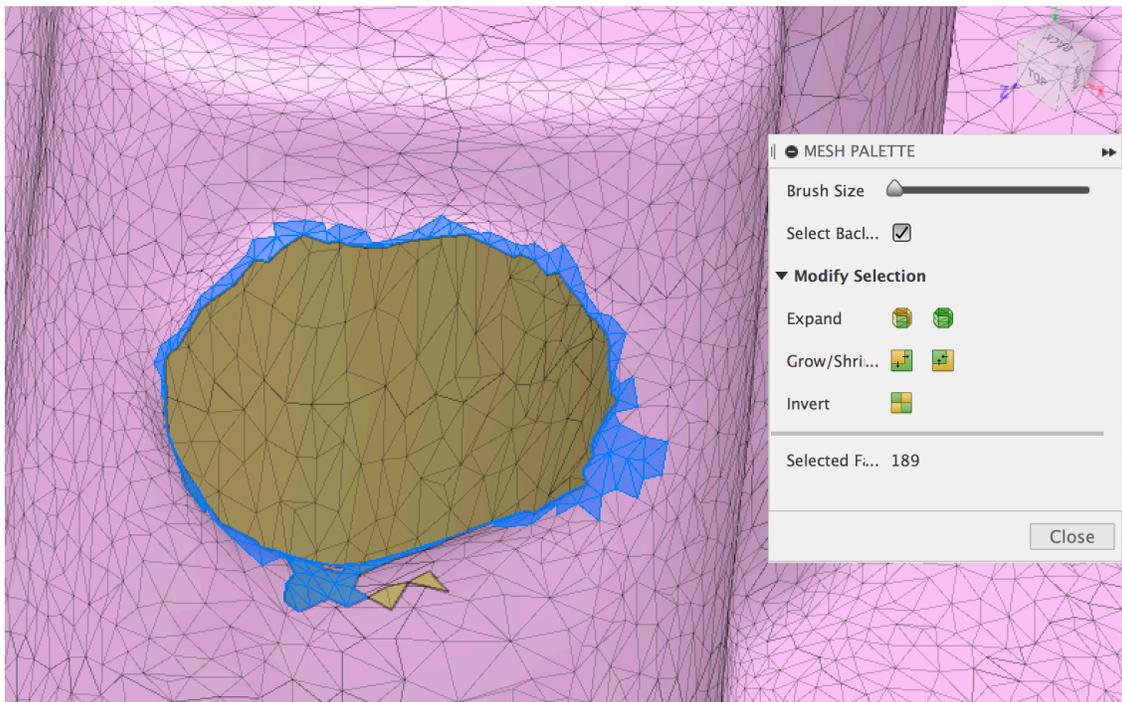
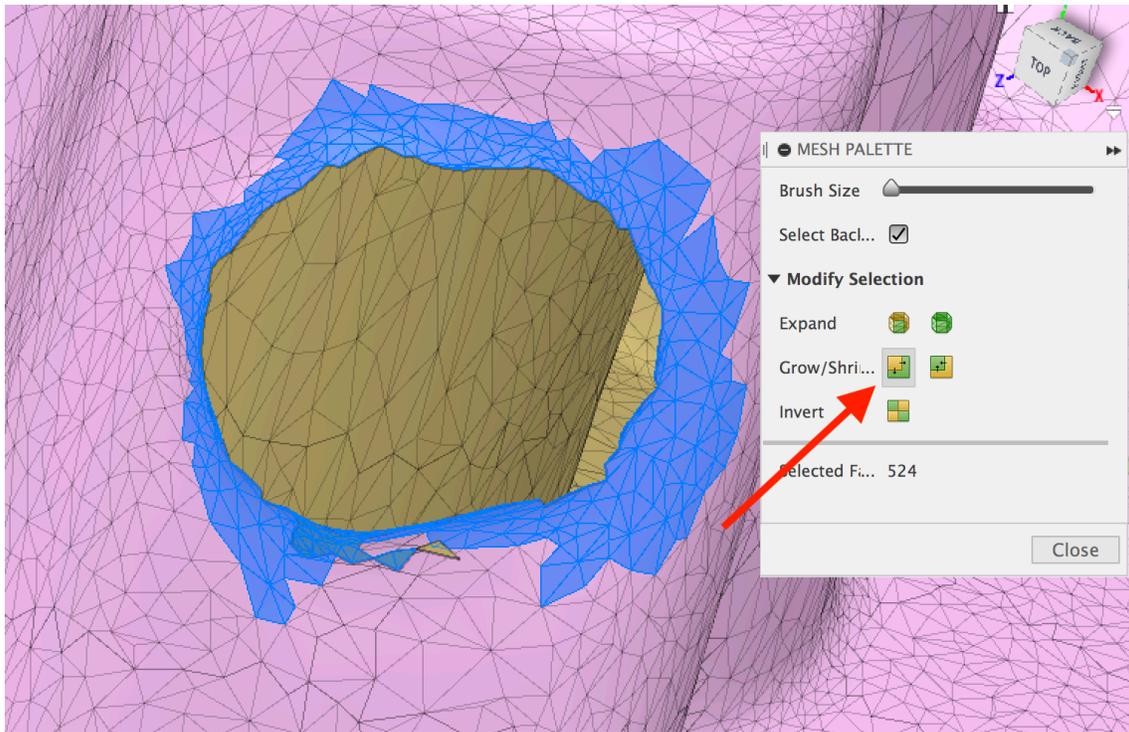


FIGURE 39 - MESH CLEANUP HOLE BOUNDARY SELECTION

And, Erase and Fill will fill in the hole. Another trick: Use Grow Selection in the Mesh Palette to increase this selection, which can help produce a smoother Fill



*FIGURE 40 - MESH CLEANUP GROW SELECTION*

Another useful hint: Turn off “Select Back Faces” when selecting open area boundaries to prevent accidentally selecting faces from the other side of the model (the gold-colored) facets:

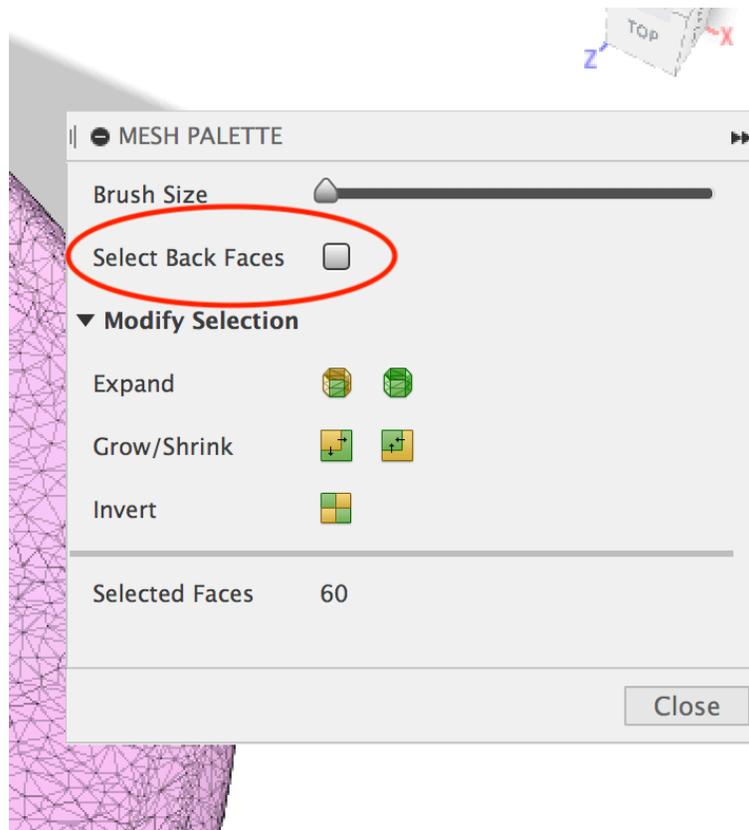


FIGURE 41 - MESH CLEANUP SELECT BACK FACES

Anyway, with enough patience, you can get a nice, smooth filled area:

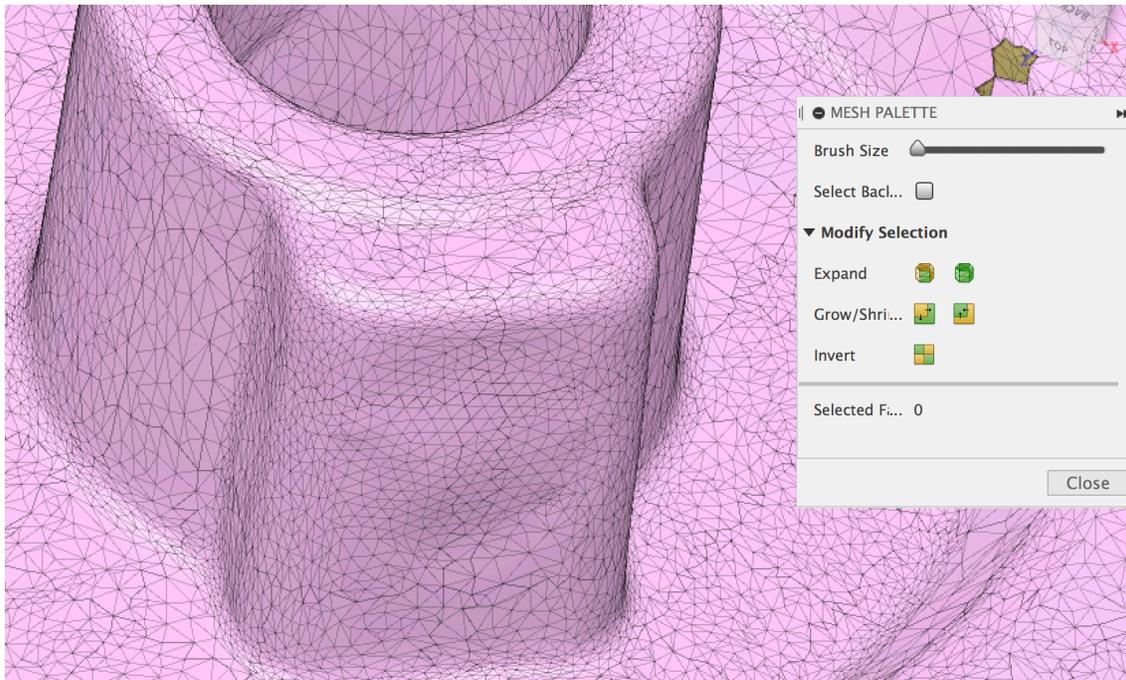


FIGURE 42 - MESH CLEANUP GEOMETRY REMOVAL RESULTS

There are lots of other mesh cleanup techniques available, including:

- Smooth to take out bumps from surfaces
- Remesh to reduce number of triangles
- Plane cut to trim off excess geometry from, say, the scanning table, and produce a nice flat planar mesh section

Here is a screencast of the cleanup process: [mesh cleanup](#). Note, this screencast has no audio.

### 2.4.3 “Clean” models from CAD data

This type of mesh contains precise representations of the surfaces of a part. These meshes are almost always produced as mesh output from a more precise format (save as OBJ from Solidworks, Onshape, etc). These models exist primarily because mesh representations are the “least common denominator” format for 3D geometry. Computer graphics systems use triangle meshes as their primary data representation, and all 3D printers speak mesh as well. So, often, 3D data is shared as triangle meshes. Another reason, I think, that mesh is used to share 3D data is that it strips away any sort of IP from the model. Our job here is to put that IP back into the model.

Clean, Precise meshes can usually skip the cleanup phase above.

## 2.5 Reverse engineering techniques for “Precise” triangle meshes

In Fusion, the main method of generating a design from either a cleaned up scanned mesh or a clean mesh is the Create Mesh Section Sketch command. This command lets you position a sketch plane within a mesh, and generate a special type of sketch primitive called a Mesh

Section. The hard part of this process, and one we can't really help you with is finding the right sections to generate.

**This requires a plan.** You need to look at the design, and as you would with a design you are starting from scratch, plan how you would model this shape. Are you planning to use Extrude, Revolve, Sweep, or Loft? Then, you need to think about which sketches you would need for those various features. For Extrude or Revolve, you just need one sketch, for Sweep you will need a profile sketch and a path sketch, each will have their own section sketch, and for loft you will need two or more profile sketches, and possibly some rail sketches as well. It is critical to the accuracy of your final model that you get these sketches in the right place and orientation. Off by a couple of millimeters, or misaligned by a few degrees, and you will not generate a very precise representation of the model. You may want to define some work planes or axes, and re-orient the mesh so that it is precisely aligned with these work geometries. You may also want to pre-position workplanes at significant locations in your mesh before even creating the mesh section sketches.

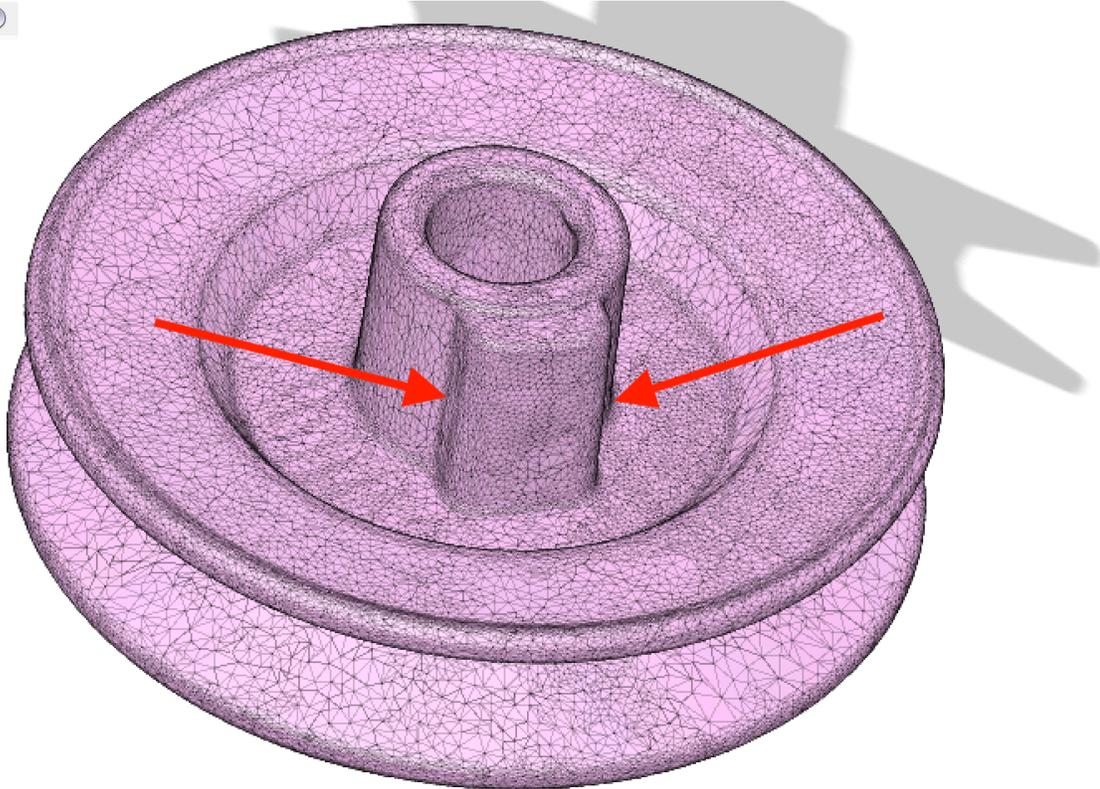
Note that this command just creates a sketch feature in the timeline. You must still edit the sketch before working further with this section.

#### **2.5.1.1 A word about R.U.L.E #1 here**

If you have not all heard of this rule, I encourage you to check it out by going to the source: [R.U.L.E #1](#). This convention focuses on Parametric Fusion designs that contain local components. In this style of design, it is a good convention to make sure that a component is created and active before creating any of the sketches, work geometry, canvases, etc. This is primarily to make sure that these items are owned by the component, so that they are transformed as the component is transformed, instanced with the component, etc. If you are creating your reverse-engineered design one part at a time, and assembling it bottom-up, this rule does not apply to you. You should just create all the geometry in the root of the component, including sketches, bodies, etc.

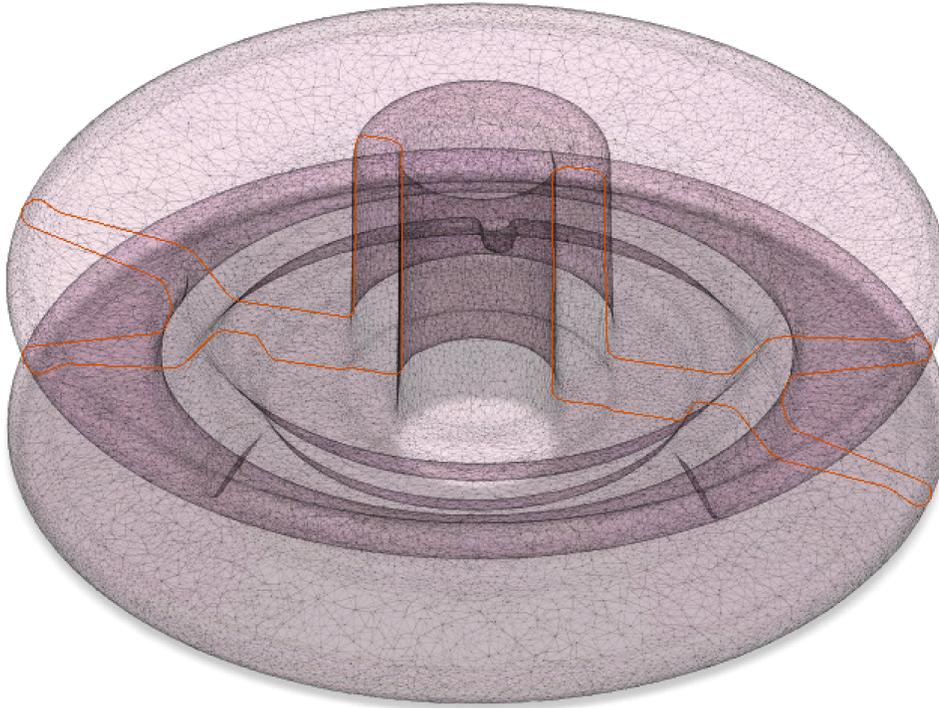
Let's use our pulley model as an example. The process is generally the same whether you are using a cleaned, scanned mesh, or a precise mesh.

The plan for this mesh is to use a Revolve feature as the main feature to generate the geometry. So, we will generate a mesh section that reflects this plan. The first thing to notice is that there is a protrusion here that is not axial-symmetric:



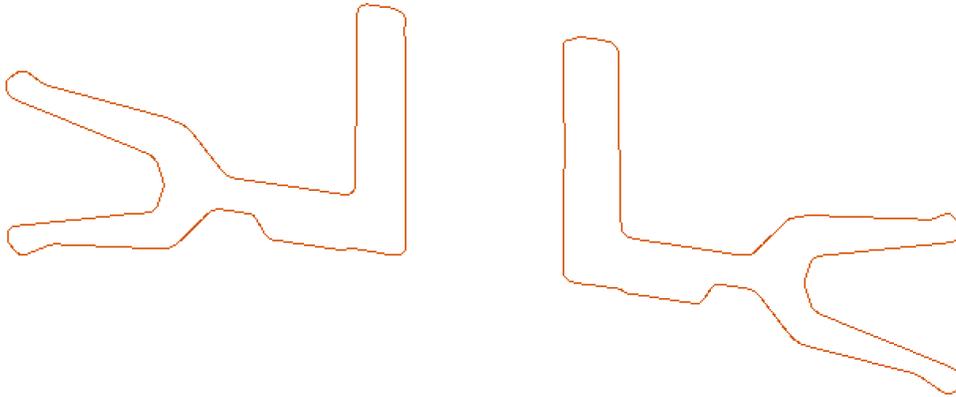
*FIGURE 43 - MESH SECTION PLAN: REVOLVE*

So, we want to pick our section in such a way as to avoid this. Here is one possible section location (mesh has been made 30% opaque to better show the section geometry):



*FIGURE 44 - MESH SECTION CREATED*

Here is what it looks like without the body:



*FIGURE 45 - MESH SECTION WITHOUT BODY*

You'll notice here that, even though the section geometry is closed, it does not define a profile. Why is that? Basically, it's because these sections are crap geometry. If you zoom in closely, you can see that the data here is pretty noisy. Lots of little line segments. If you think about it, you can see why. The mesh section command is literally just cutting through the mesh with a plane, so each intersection will generate just a tiny line for each facet. So, we need to do a bit more work before we can do real solid modeling with this geometry

### **2.5.2 The Fit Curves to Mesh Section command**

This command was added to Fusion for precisely this workflow. Now that we have defined some mesh sections, we need to fit precise, native Fusion curves to these sections. That is exactly what the Fit Curves to Mesh Section command is designed to do. This is a quite powerful command. Let's take a quick look at the dialog:

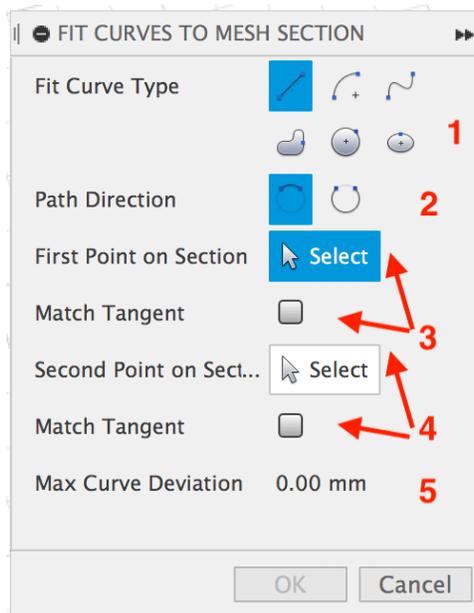


FIGURE 46 - MESH SECTION DIALOG

Each of these fields is important, and worth diving into just a little

1. **Fit Curve Type.** This choice lets you decide, for the next curve to be created, what type of curve to use. The choices are: line, arc, spline, closed spline, circle, and ellipse. This is where your judgement comes in: If you think the section curve should be a line, then choose line. If you think it is an arc, then choose an arc. Spline is easy, because it will always produce a good fit, but if you think the curve should be a circular arc, then choose arc.
2. **Path Direction.** This setting is important because the mesh section usually forms a loop. This setting choose whether the next curve created should take the short path or the long path. It is not clockwise/counter-clockwise.
3. **First Point on Section/Match Tangent.** This pair of settings controls the start of the curve. If you are using this command to fit more than one curve at a time, this input will be auto-populated by the “second point selection” of the previous curve. The “Match Tangent” setting applies when the first point is adjacent to another fitted curve, and, as its name implies, controls whether the fitted curve is constrained to be tangent to the adjacent curve.
4. **Second Point on Section/Match Tangent.** Same as #3, but for the second point. The “Match Tangent” check box only applies if the second point is adjacent to another fitted curve.
5. **Max Curve Deviation.** Use this to control how closely a fitted curve is to the mesh section. Be aware, though, that with curve types such as arc, circle, and ellipse, too tight of a tolerance can cause a fit failure

For this example, we only need to fit curves to one of the sections, because our intention is a Revolve feature. So, pick one side. Here is the result:

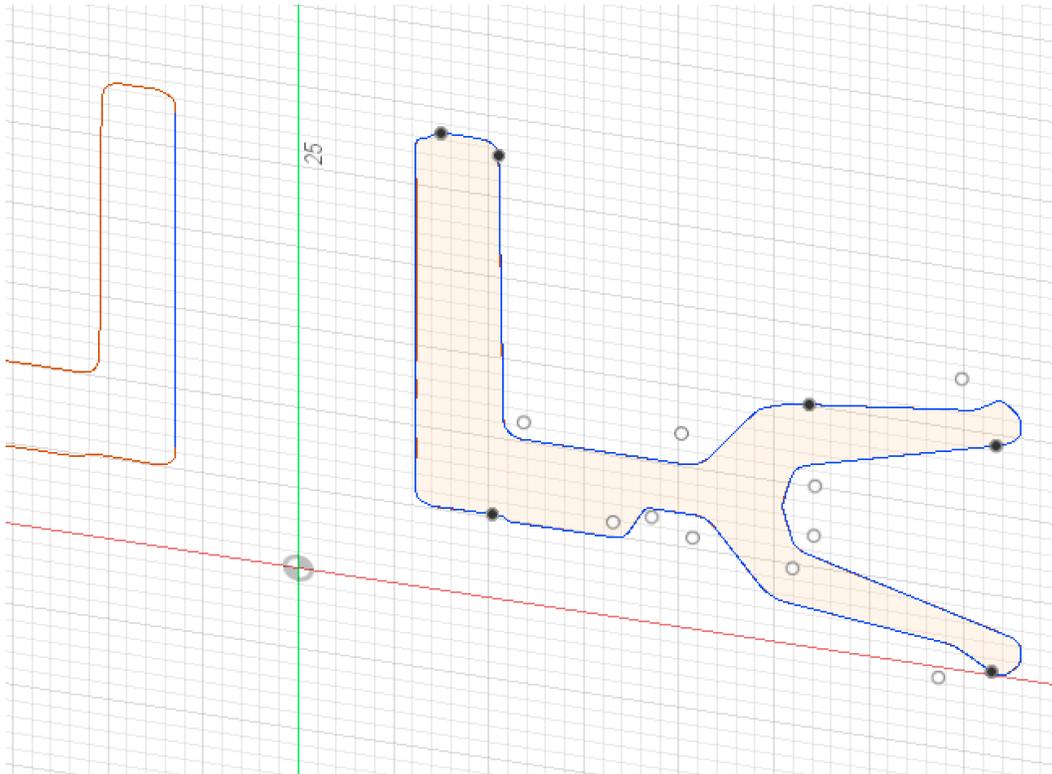


FIGURE 47 - MESH SECTION FIT CURVES RESULT

A couple of things to note. While we only need one profile, you'll notice that I also did a line fit from the vertical line on the other section. This is so that I can generate a midpoint line to get the revolve as close as possible to the original.

I used a mix here of lines, arcs and splines.

**A word about de-features the profile.** In the above section, I just fitted curves for the whole thing. You may want to avoid areas which should be done as features after the main solid modeling operation, such as fillets and chamfers. You can do this using the Extend command in sketch:



FIGURE 48 - MESH SECTION - REMOVING FILLETS

Here is a screencast of this process: [using fit curves to mesh section](#) – note: This video does not contain audio

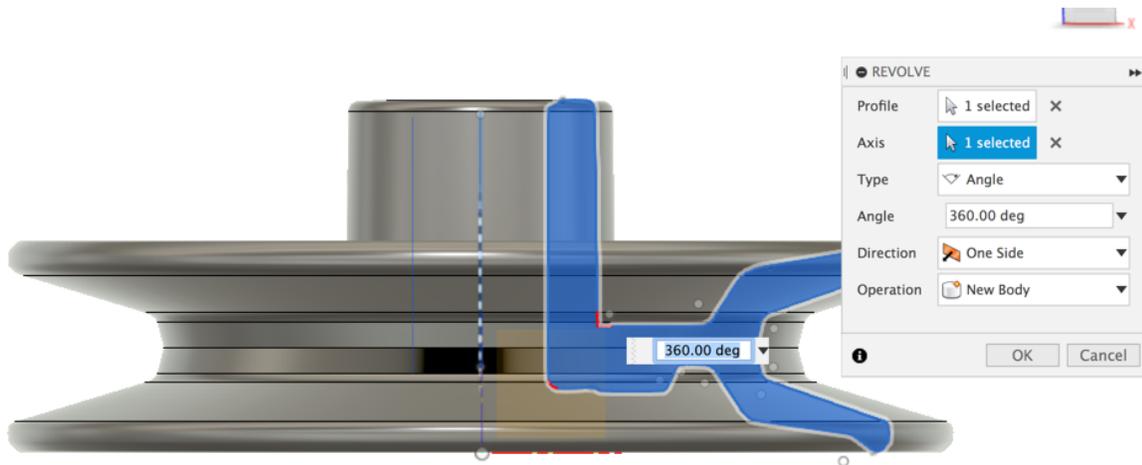
**Tip:** This command works really well when you fit the whole section in one go. You can stop and re-start the command, but it can get confused. I find it's better to just pick a direction (usually clockwise around the profile), and do it all in one command invocation.

### 2.5.2.1 A word about associativity in a parametric design

In a Parametric design, the sketch and the mesh section curves **are** associative to changes in the mesh. However, the fitted curves discussed in the next topic **are not** associative to the mesh section curves, so be aware of that when using this workflow. Make any mesh change before starting to fit curves.

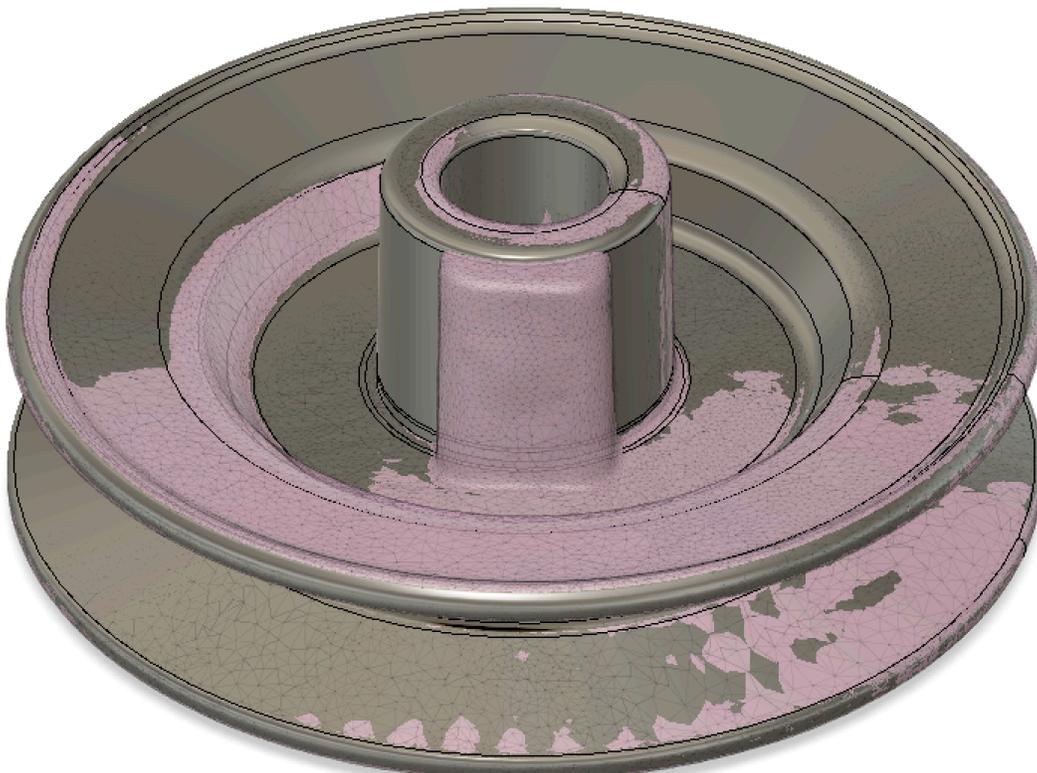
### 2.5.3 Create the parametric features

The final step in reverse engineering a part from a Precise mesh, once you have the proper profiles, paths and rail curves from the mesh, is to create the features for the part. In this case, we'll use a Revolve feature.



*FIGURE 49 - MESH SECTION REVOLVE RESULTS*

Then, let's check how close the result is:



*FIGURE 50 - MESH SECTION COMPARISON WITH MESH*

Not bad! This “Z-fighting” is a good indication that the approximation is pretty close. It looks like the mesh may have been rotated a bit, and some tweaking can probably get better results, but this is the basic idea.

Next, let's add the protrusion on the side of the pulley. First, create a new mesh section sketch:

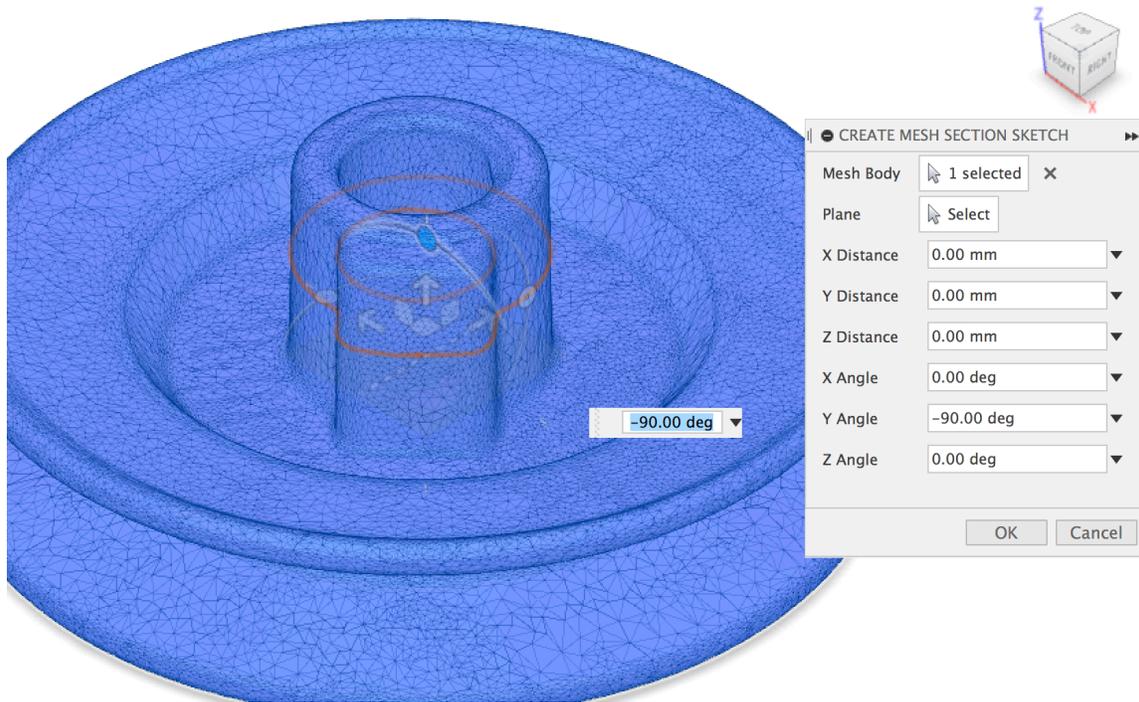


FIGURE 51 - MESH SECTION ADDING SECOND SECTION

Next, because we will be working against the solid from the Revolve, let's project the face from the solid:

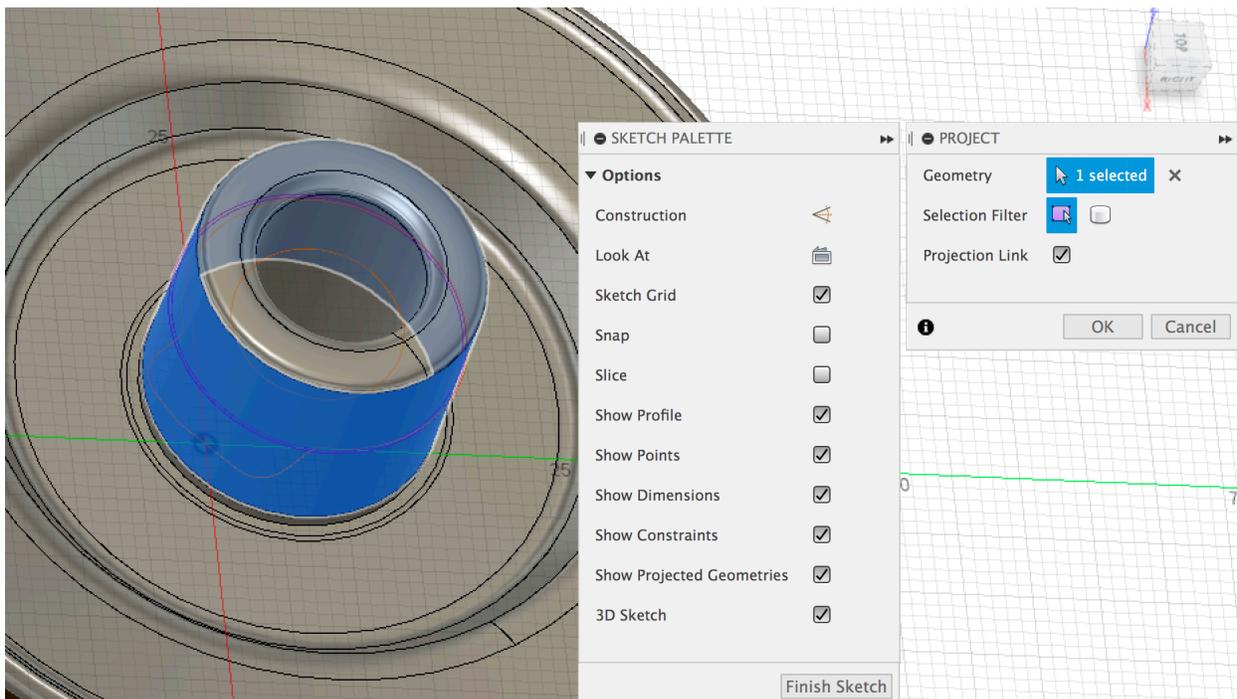
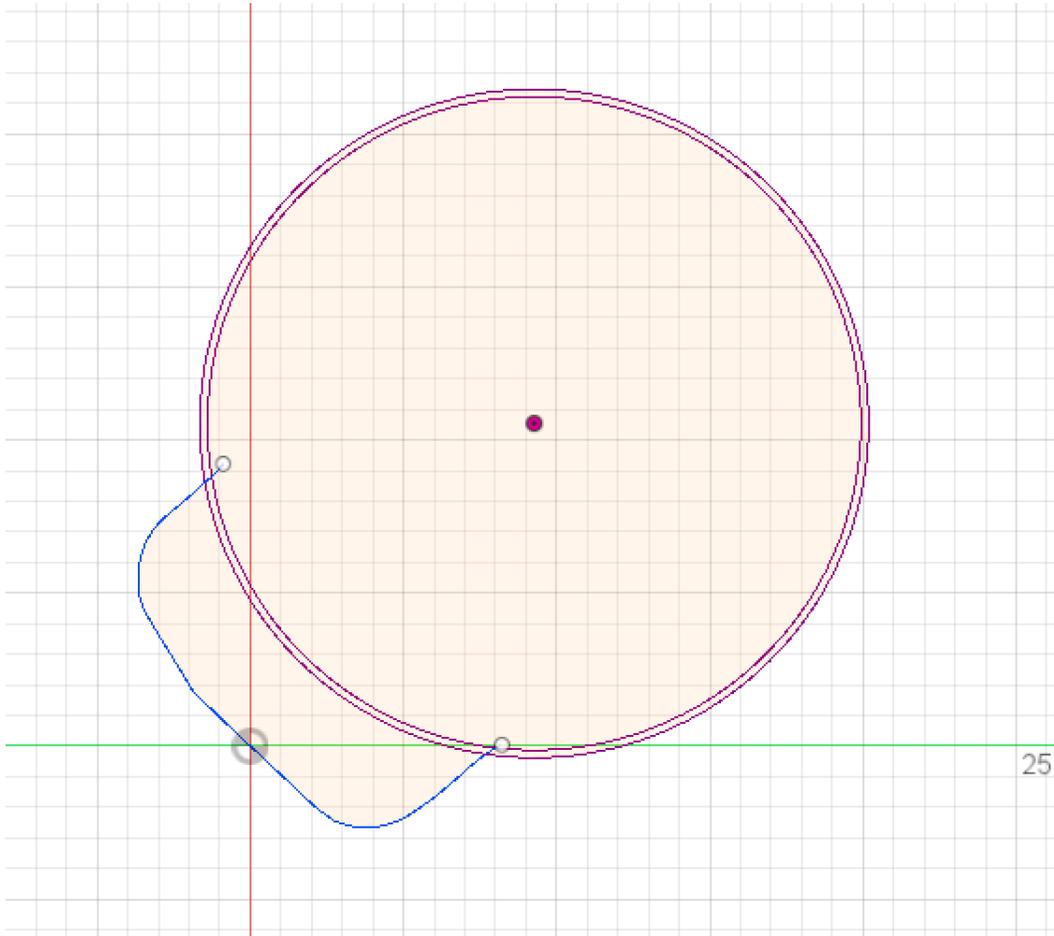


FIGURE 52 - MESH SECTION PROJECTING BODY INTERSECTION

This also means we only need to fit curves for a portion of the mesh section:



*FIGURE 53 - MESH SECTION PROTRUSION PROFILE*

Then, we just Extrude the portion of the profile between the curve fitting and the projected cylinder:

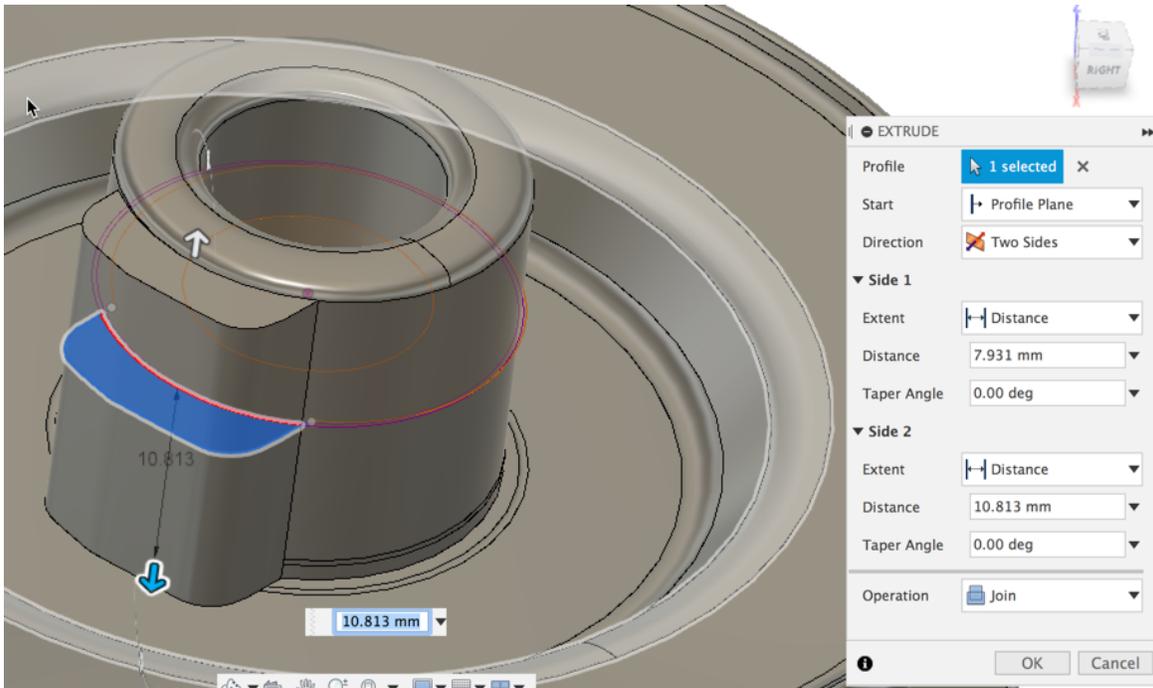


FIGURE 54 - MESH SECTION PROTRUSION EXTRUDE

Add a couple of fillets, and we're all set:

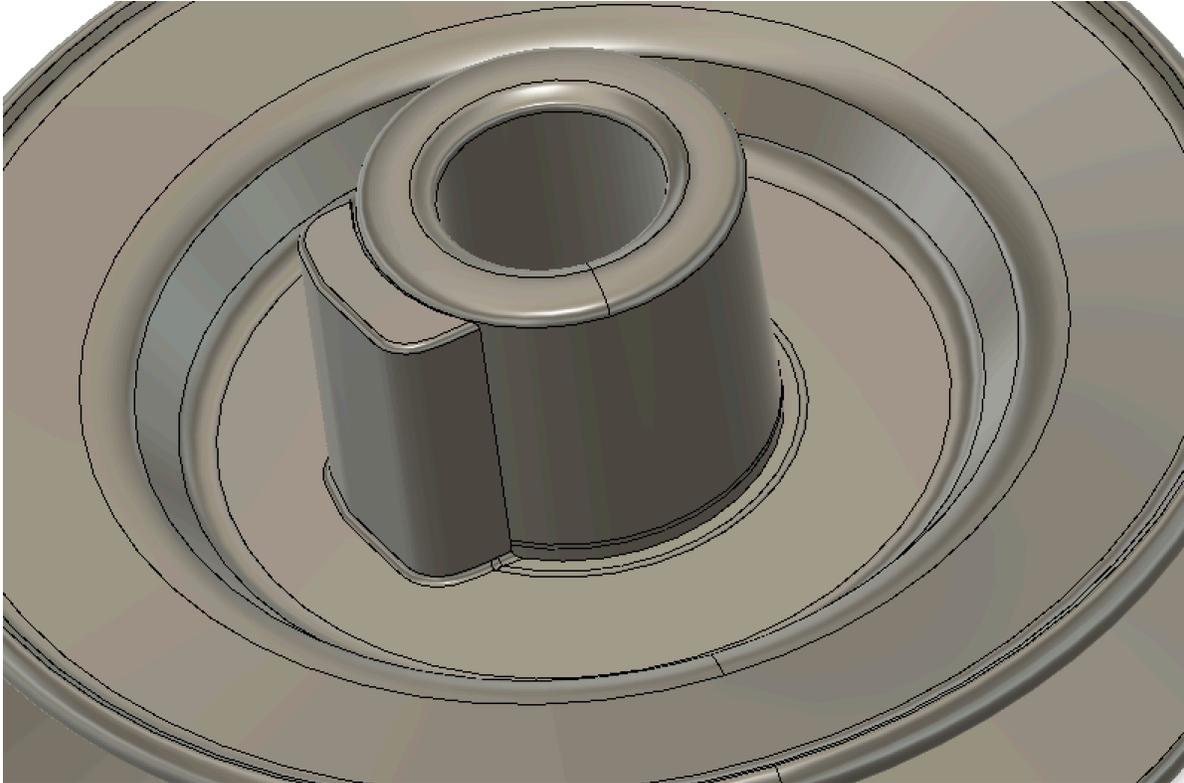


FIGURE 55 - MESH SECTION PROTRUSION RESULTS

At the end of this process, we have a fully parametric version of this design that can be edited to make adjustments, etc. This is far more useful than a dumb body version of the part.

## 2.6 Looking toward the future

AU is as much about looking at what's next as it is about working with the current products, right? Well, we are going to do our own "Main Stage" presentation here, without the DJ and flashing lights. We've been hard at work on Reverse Engineering workflows for mesh data in Fusion. It's not quite ready just yet, but we can give you a bit of a preview, I think.

We've seen, above, an example of some Autodesk technology for converting "organic" meshes to BRep by first converting to quad meshes, then to T-Splines, then to BRep.

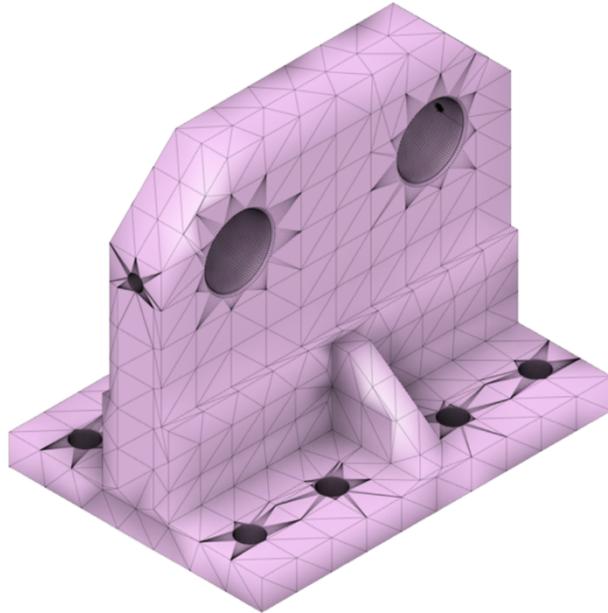
Next, addressing the "precise" clean mesh, Fusion will also soon have the ability to do a more automatic conversion to BRep for this workflow. We say "more automatic", because there are some caveats here. You need to tell Fusion where the face boundaries are. The good news is that the existing Mesh command "Generate Face Groups" will do this, if the mesh is really well-formed (as in: it came from BRep data originally via a Save as STL-type operation)



FIGURE 56 - MESH FACE GROUPS

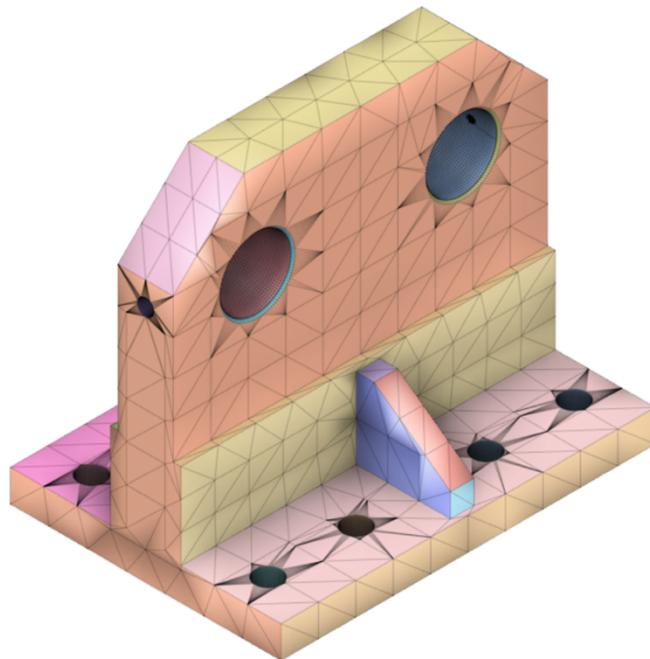
Below is an example simple design and the current results of this prototype:

This is the original mesh body. Clearly a "precise", "clean" mesh:



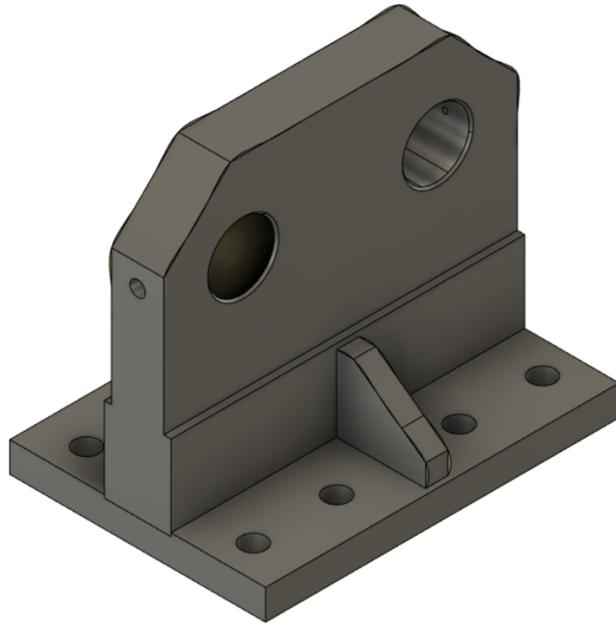
*FIGURE 57 - PRECISE MESH EXAMPLE*

This is the result of the automatic face group command:



*FIGURE 58 - MESH FACE GROUPS FROM PRECISE MESH*

Then, this is the result of the automatic Mesh to BRep command:



*FIGURE 59 - AUTOMATICALLY GENERATED BREP FROM FACE GROUPS*

A caveat here is that this process will not generate an editable parametric design, leaving you with an uneditable BRep model, so there is still a place for the Create Mesh Section Sketch/Fit Curves to Mesh Section workflow.

### **3 Reverse engineering from image data**

Sometimes you may not even have a mesh model of the component you need. You just have a couple of old photos of the part, and some critical dimensions to indicate size. Let's take a quick look at this workflow. The main tools here are Insert Attached Canvas and the Calibrate tool, followed by some modeling techniques that make use of the canvases as references.

Let's look at an example from the furniture industry. Imagine that you want to design this chair, and all you have are these images:



FIGURE 60 - CHAIR IMAGES

The first step is to separate out a couple of these images, and attach them to the origin planes:



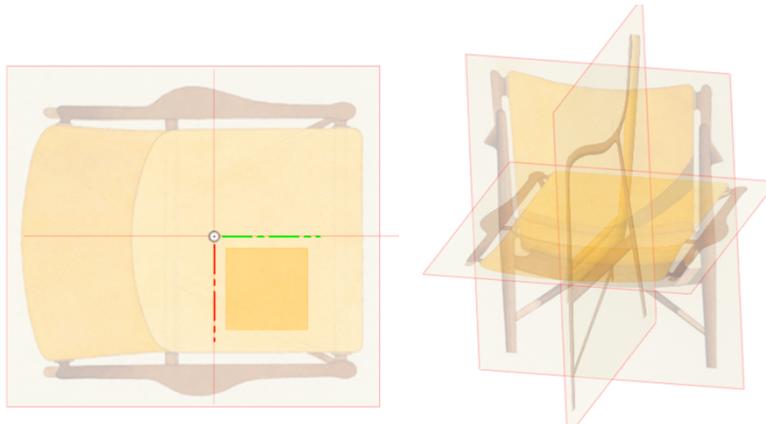


FIGURE 61 - CHAIR CANVAS IMAGES

### 3.1 Insert Attached Canvas

This command allows you to insert an image from any common image format into your design. By setting up several images on some pre-defined workplanes, it is easy to use these images to help reverse engineer from just photos. In this particular case, because the design is inherently symmetric, it is important to make sure that the images are centered on the origin planes. This will make creating symmetric geometry much easier.

**Tip:** PNG images allow a transparent Alpha channel, making tracing profiles much easier. Other bitmap formats are supported, but we find that PNG gives the best experience in this workflow.

**Tip:** Also, setting the opacity of the images to around 30% makes it so you can still see the images, but you can also see the model geometry you create.

### 3.2 Calibrate

Next, you want to make sure that the images are the correct size for your design. The Calibrate command is not in any toolbar dropdown, it is only available as a context menu item on an inserted Canvas. This command allows you to semi-precisely scale the Canvas so that it represents the true dimensions of the part. It requires some care in placing the extents of the dimension, as there is nothing to “snap” to in an image.

Select the Canvas, and right click to see the command:

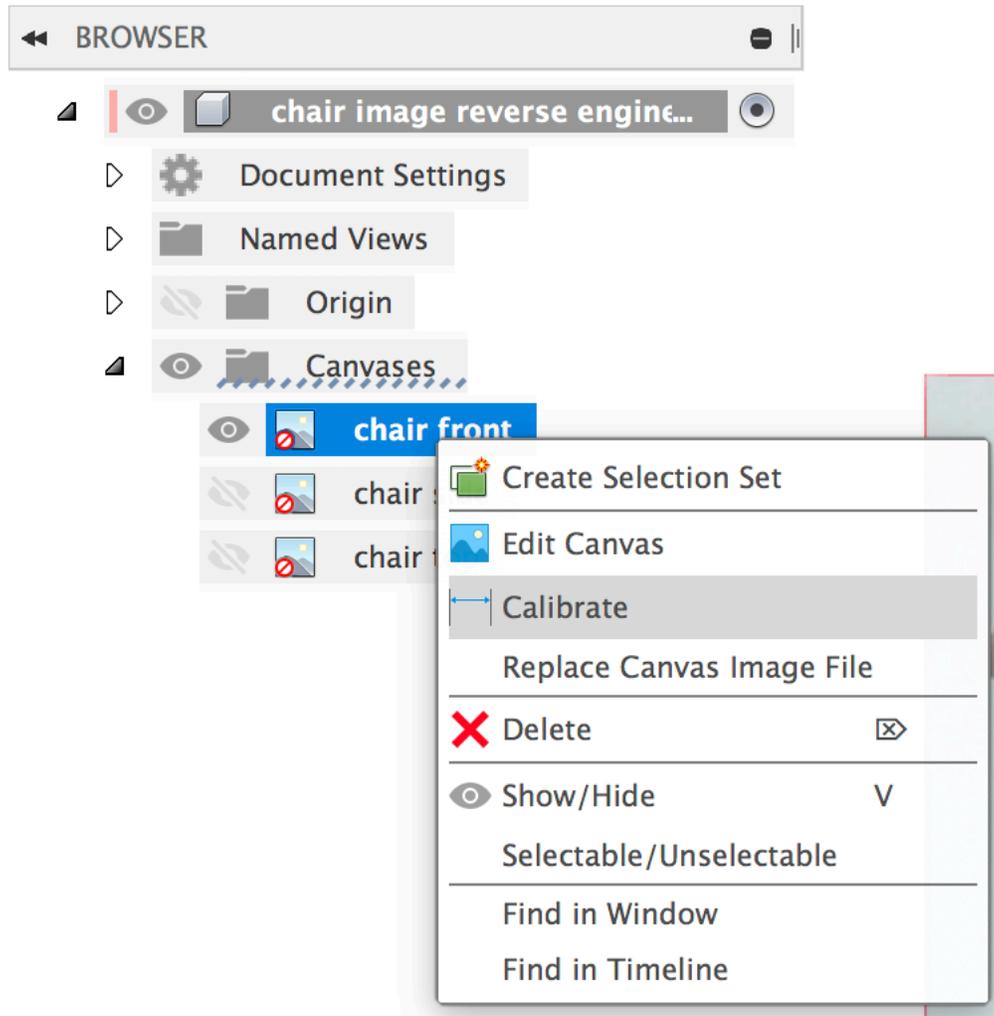


FIGURE 62 - CALIBRATE MENU

Then, select two important points on the image, for which the value is known:



FIGURE 63 - CALIBRATE COMMAND

Repeat this for the other images, and use Edit to adjust the positioning as closely as possible. This can be a bit tricky, and was tricky in this case.

### 3.3 Creating Sketches and Features to reverse engineer from Canvas data

The next step is to use regular modeling commands, but visually referencing the Canvas data. This particular example makes use of Sketch and Sculpt/T-Splines, but similar techniques can be applied to solid or surface modeling as well.

#### 3.3.1 Creating 2D Sketches

In this example, we ultimately want 3D sketches for the frame. To start toward that goal, we can first create 2D sketches on each view. This is usually a simple tracing operation. In the chair example, we only need centerlines of the frame. This results in these set of 2D lines:

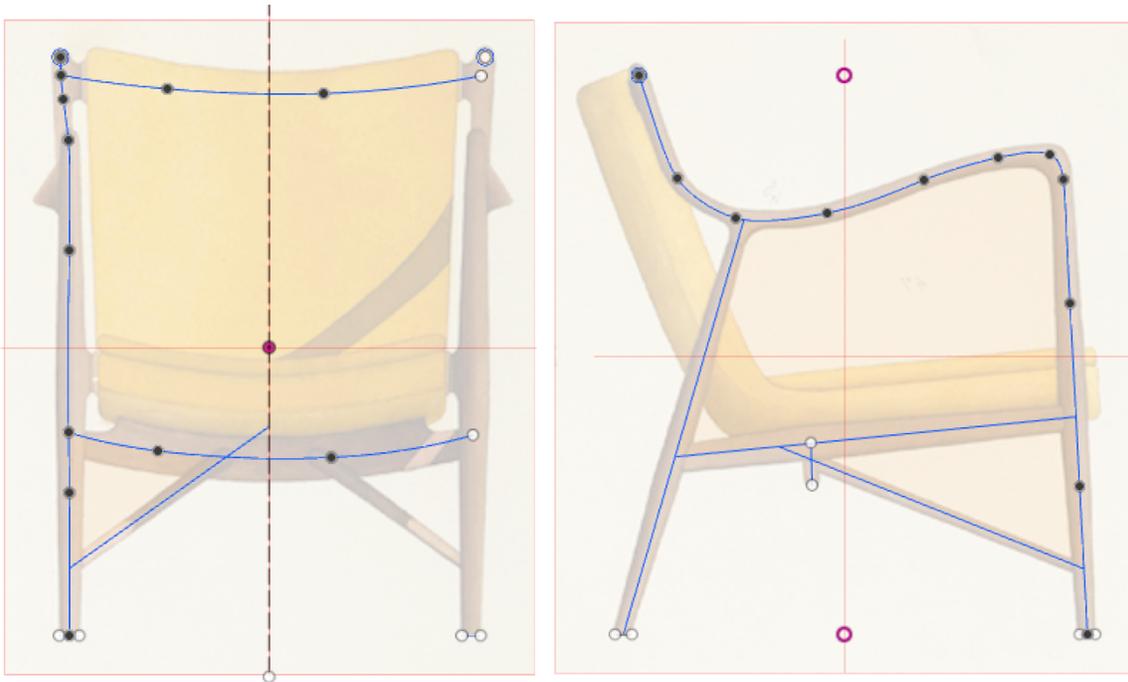


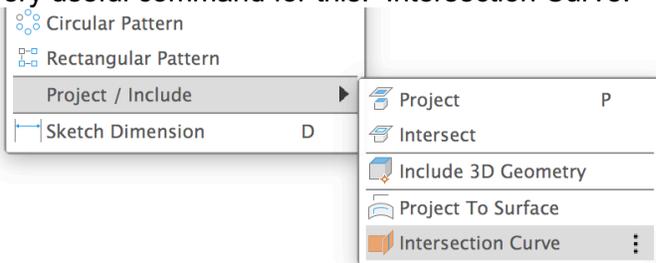
FIGURE 64 - 2D SKETCHES FOR THE CHAIR FRAME

Note: The plan here is to model only one side of the frame (other than the cross pieces), and mirror the results. This is why only one set of 2D curves for the side parts of the frame have been created.

Here is a screencast showing the creation of these sketches: [2D chair frame sketch creation](#). This screencast does not contain audio.

### 3.3.2 Creating 3D sketches

In the case of this chair, we want to create 3D sketches from this 2D sketch data. Fusion has a very useful command for this: Intersection Curve:



**Creates a 3D sketch curve from intersecting geometry.**

**The first selection is the sketch curve. The second selection is the sketch curve or faces to intersect with. You can select multiple curves or faces to intersect with.**

FIGURE 65 - INTERSECTION CURVE COMMAND

This command is for precisely this kind of workflow: It can create a 3D curve from a pair of intersecting 2D curves (it can do this for surfaces, as well, but in this usage, we will focus on the 2D curves created in the previous step). For instance, in this example, to create the 3D curve for the top armrest/back/front leg curve, we can select the two curves in red below:

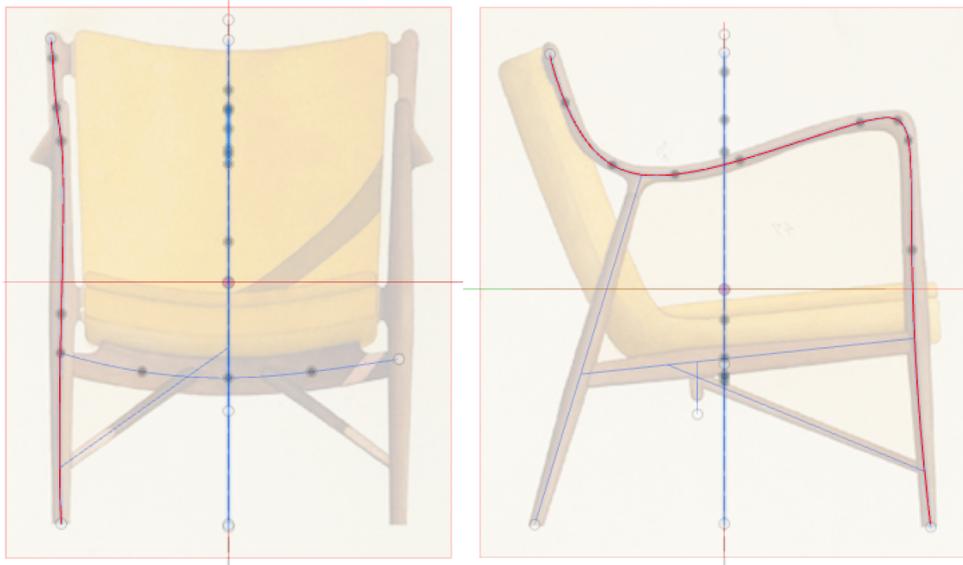
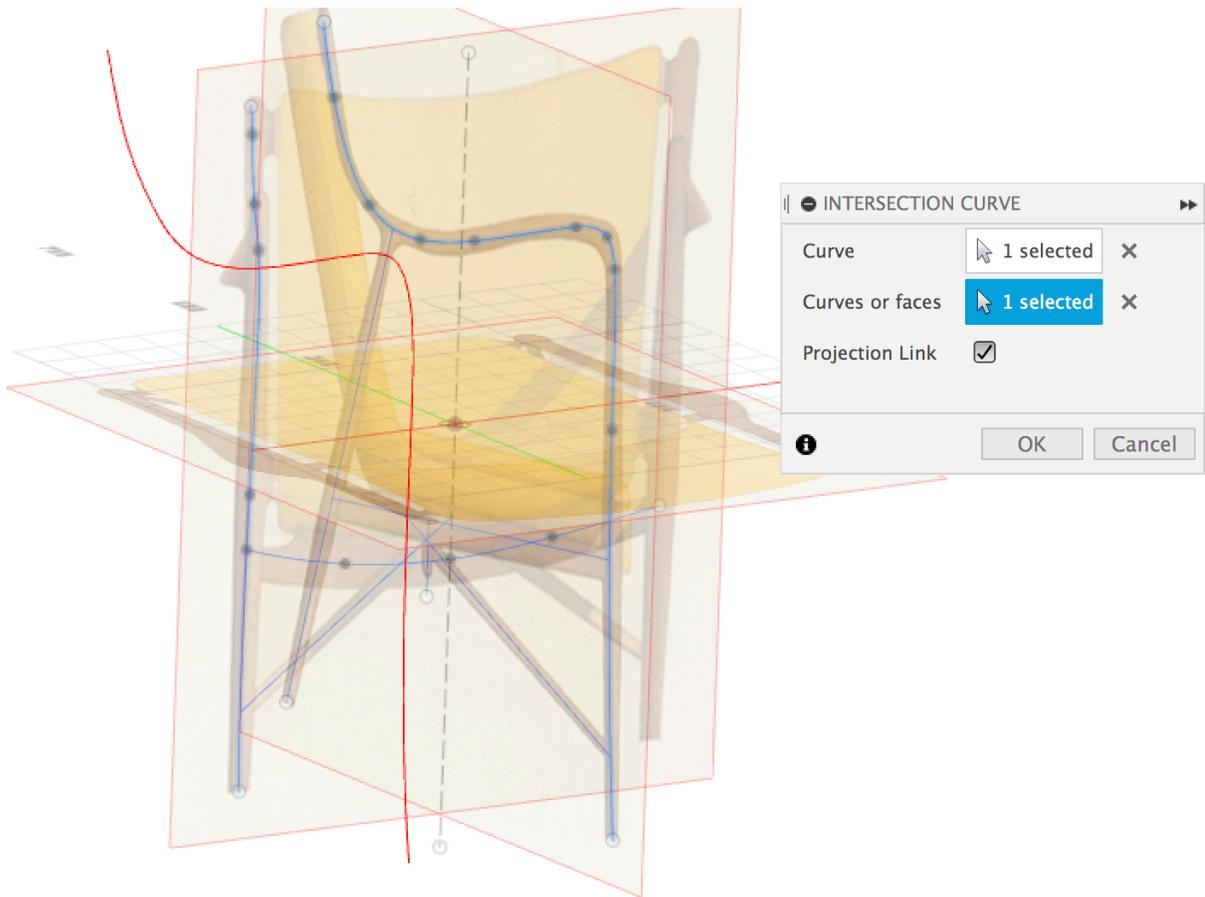


FIGURE 66 - INTERSECTION CURVE EXAMPLE INPUT

And the resulting 3D curve is:



*FIGURE 67 - INTERSECTION CURVE RESULT*

Completing this for the rest of the curves, this is the result:

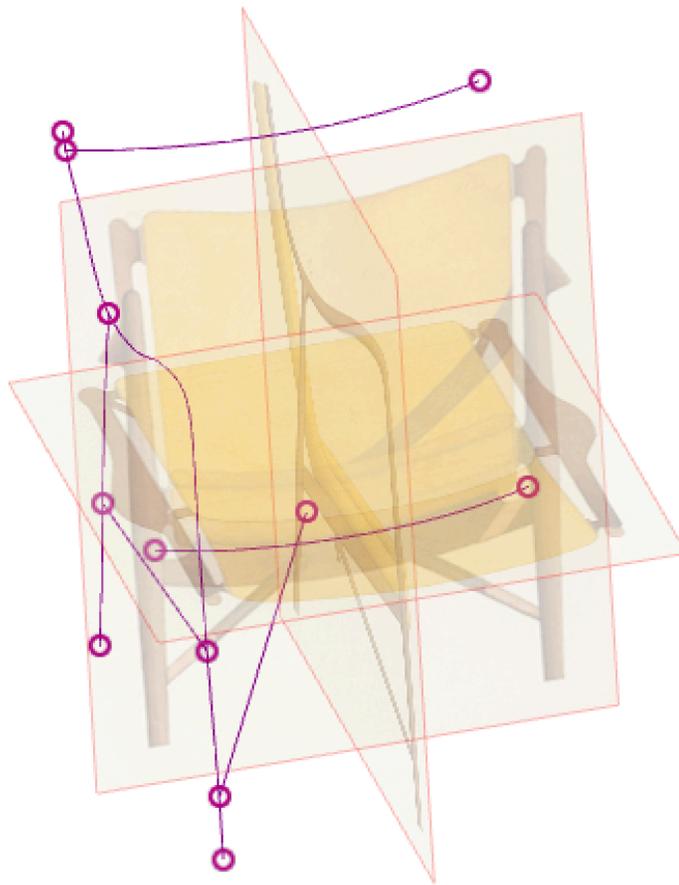


FIGURE 68 - CHAIR FULL 3D CURVE RESULT

Here is a screencast showing building 3D curves using Intersection Curve for this chair: [3D curve creation using Intersection Curve](#). This screencast does not contain audio.

### 3.3.3 Creating solid geometry

The next step in this method is to use the sketch data to create 3D solid geometry (or surface geometry, if that is your preferred modeling workflow). In the chair example, we used Sculpt/Form/T-Splines to model the frame. In Sculpt mode, create a Pipe for each frame segment. Choose a diameter that is pretty close to the frame member size, smooth display, and square ends:

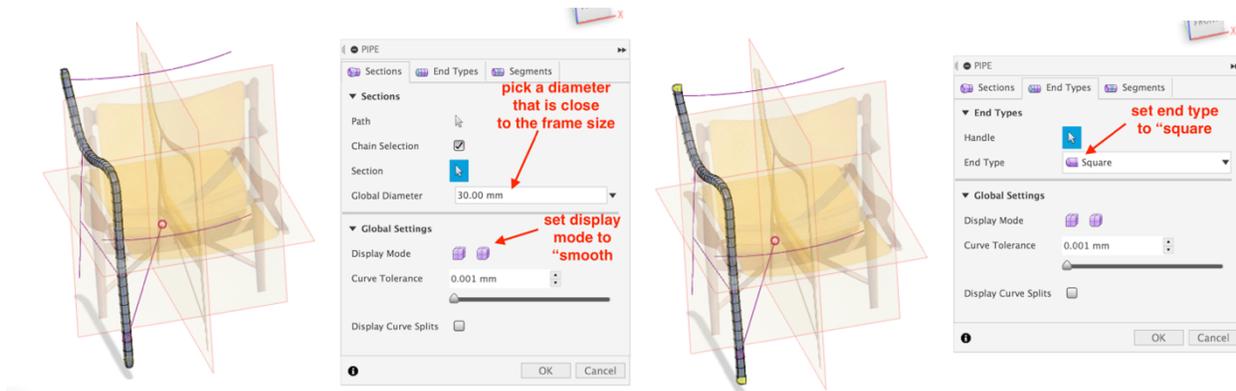


FIGURE 69 - T-SPLINE PIPE CREATION

Next, using Edit Form, massage the shape into one that matches the form, using the Canvases as a visual guide. This is not a T-Splines class, but tools like Soft Modification, scale, different orientation and transform modes can all be used to help edit the form into a shape that matches the frame elements:

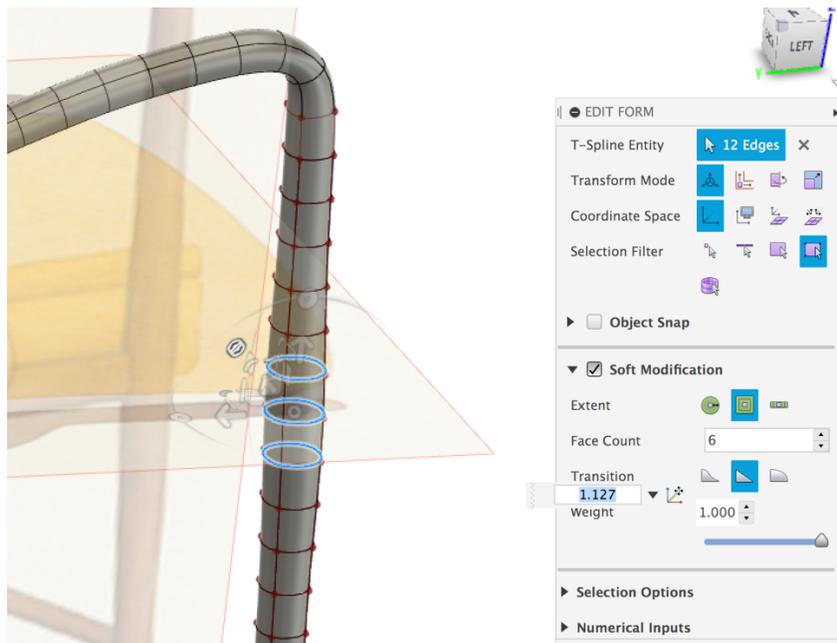
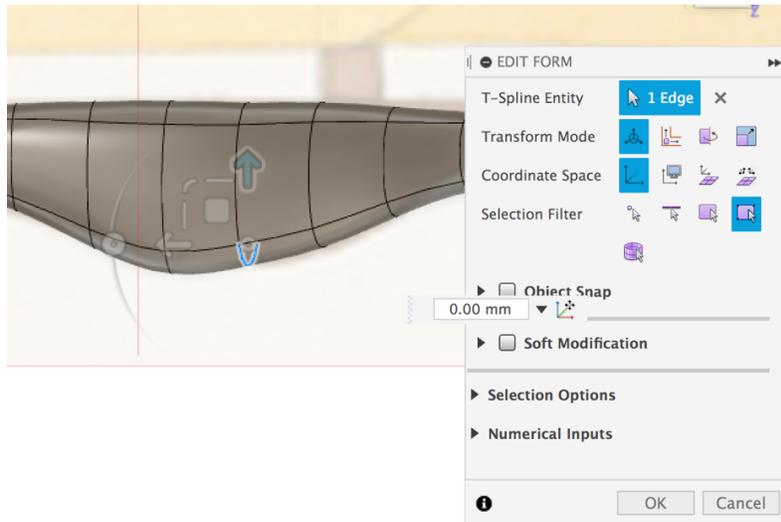
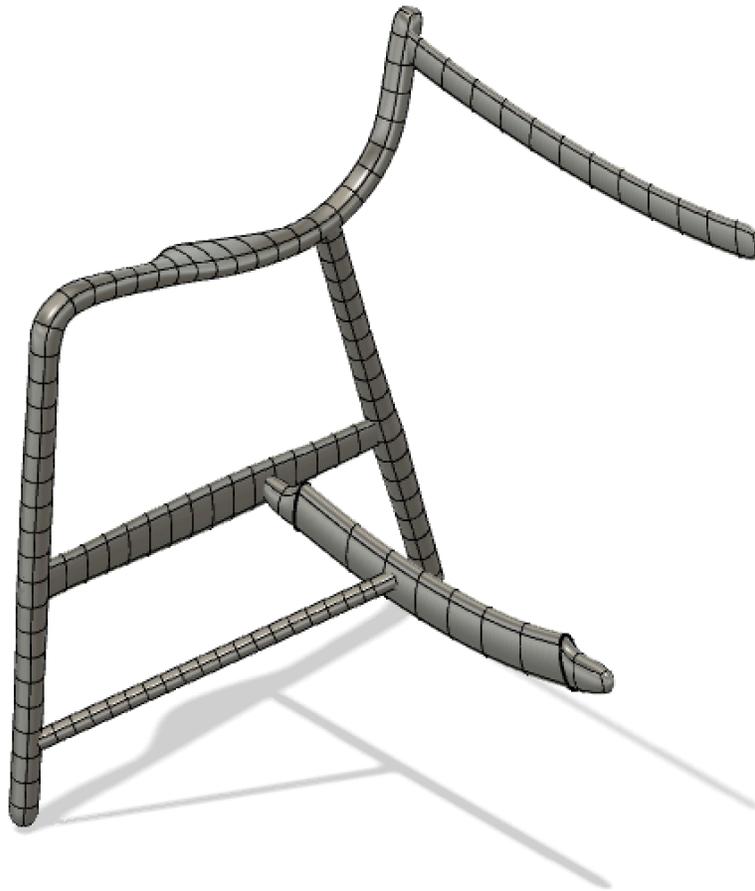


FIGURE 70 - T-SPLINE PIPE CREATION

Here is the result of this step – shapes of half of the frame have been defined:



*FIGURE 71 - CHAIR FRAME T-SPLINES RESULTS*

Here is a screencast showing the method used to create one of the frame members: [using T-Splines to create a frame element](#) This screencast does not contain audio.

Next, exit Sculpt mode to convert these forms to solids. Then, mirror the pieces of the frame to get the full frame, and Combine them all into a single solid (or leave them as separate components if you are going to manufacture this...):



*FIGURE 72 - CHAIR FRAME B-REP RESULT*

For the chair cushions in this example, we did just a basic T-Spline workflow of a Box primitive, then used Edit Form to get it into the correct shape. Final result:



FIGURE 73 - CHAIR FINAL RESULT

### 3.4 Summary of this workflow

In this section, we looked at a way that you can use simple image information, combined with a few sketch and T-Spline operations to reverse engineer a model.

## 4 Reverse engineering sheet metal parts

This section covers a “different” kind of reverse engineering workflow. However, this one is so useful and so common that we thought it was important to at least introduce in this class.

Sheet metal parts are very common in the design world. However, each CAD product has its own, slightly different sheet metal toolset, which are, of course, not compatible with each other. However, you still may have a requirement to take an existing design and reverse engineer it.

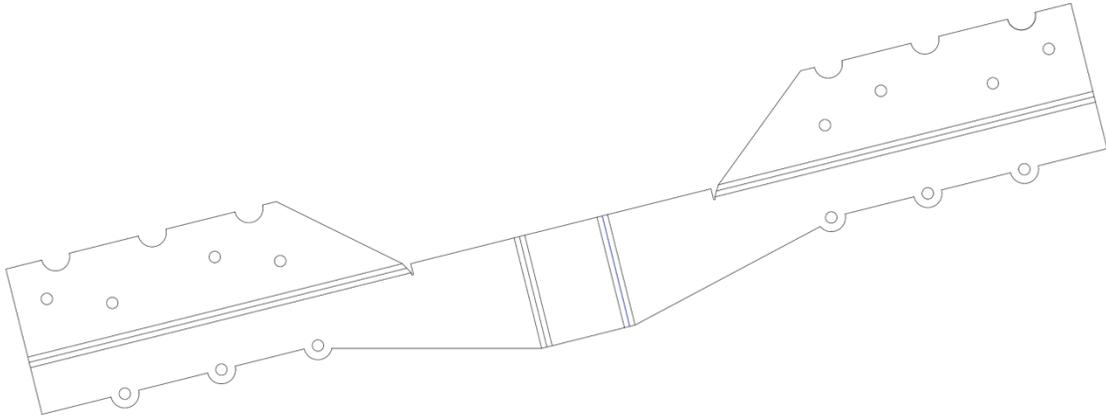
There are two main workflows that we will cover, each with very different starting points:

- Reverse engineering sheet metal parts from a 2D flat pattern (usually DXF)
- Reverse engineering sheet metal parts from a 3D “dumb solid” (STEP, Solidworks, CATIA, etc)

The example design used here is exceedingly simple, but the concepts transfer easily to more complex designs.

#### 4.1 Sheet metal from flat pattern

Often, all that you have to start with is an existing flat pattern design. The most common format of this data is DXF. So, say, for example, that all you have is this:



*FIGURE 74 - FLAT PATTERN DXF*

You know this is a sheet metal part, and you want to re-create the folded version of it.

##### 4.1.1 Import DXF into Fusion

The first step is to import the DXF into a Fusion sketch. We recommend turning off “bend extent” layers, if your DXF has them. However, having the bend line itself is critical to this workflow:

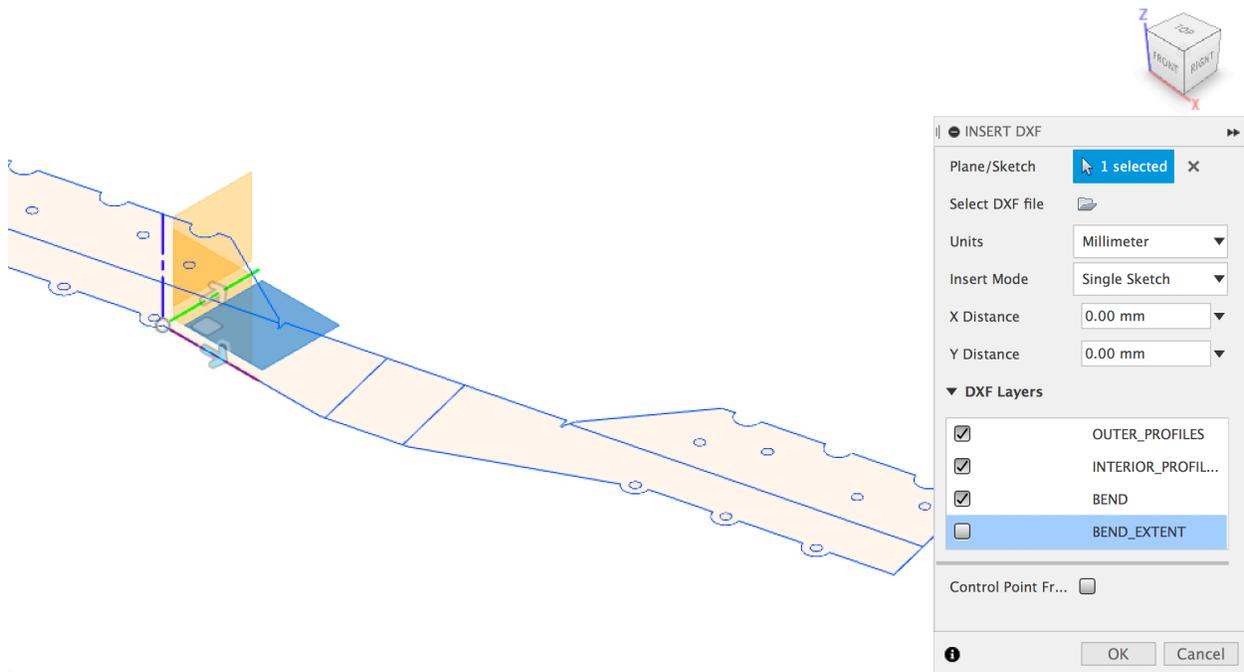


FIGURE 75 - INSERT FLAT PATTERN DXF INTO SKETCH

Next, we recommend converting the bend lines to construction. This is not strictly necessary, but will simplify the next step:

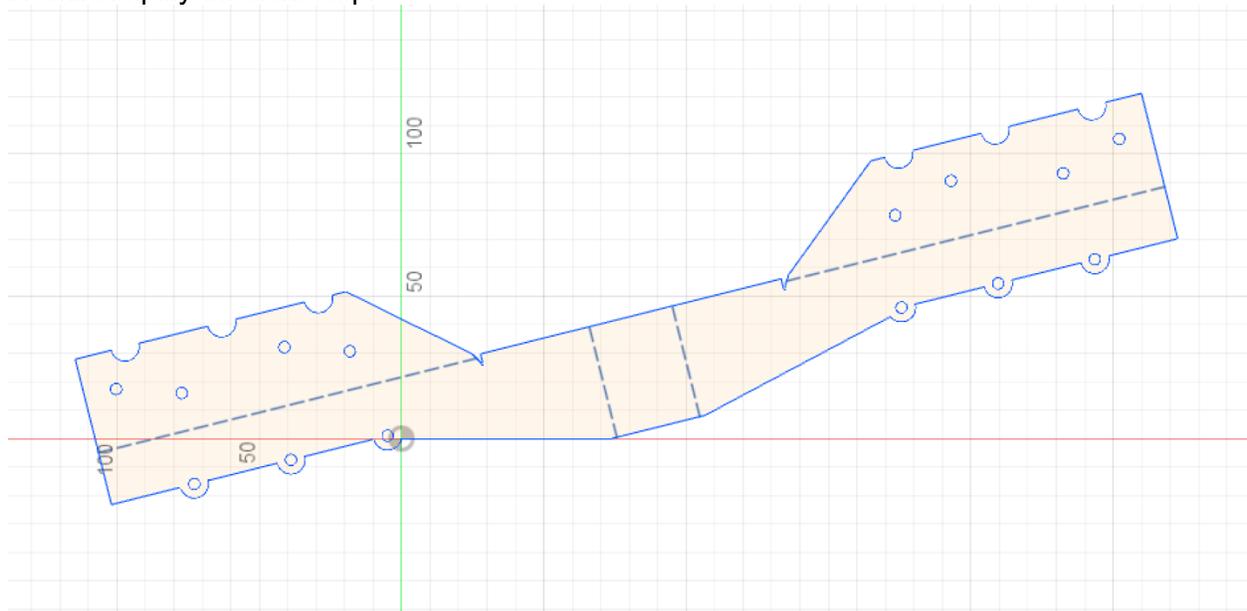


FIGURE 76 - SHEET METAL BEND LINES AS CONSTRUCTION

#### 4.1.2 Create a Flange

The next step is to create a Flange feature from the flat pattern:

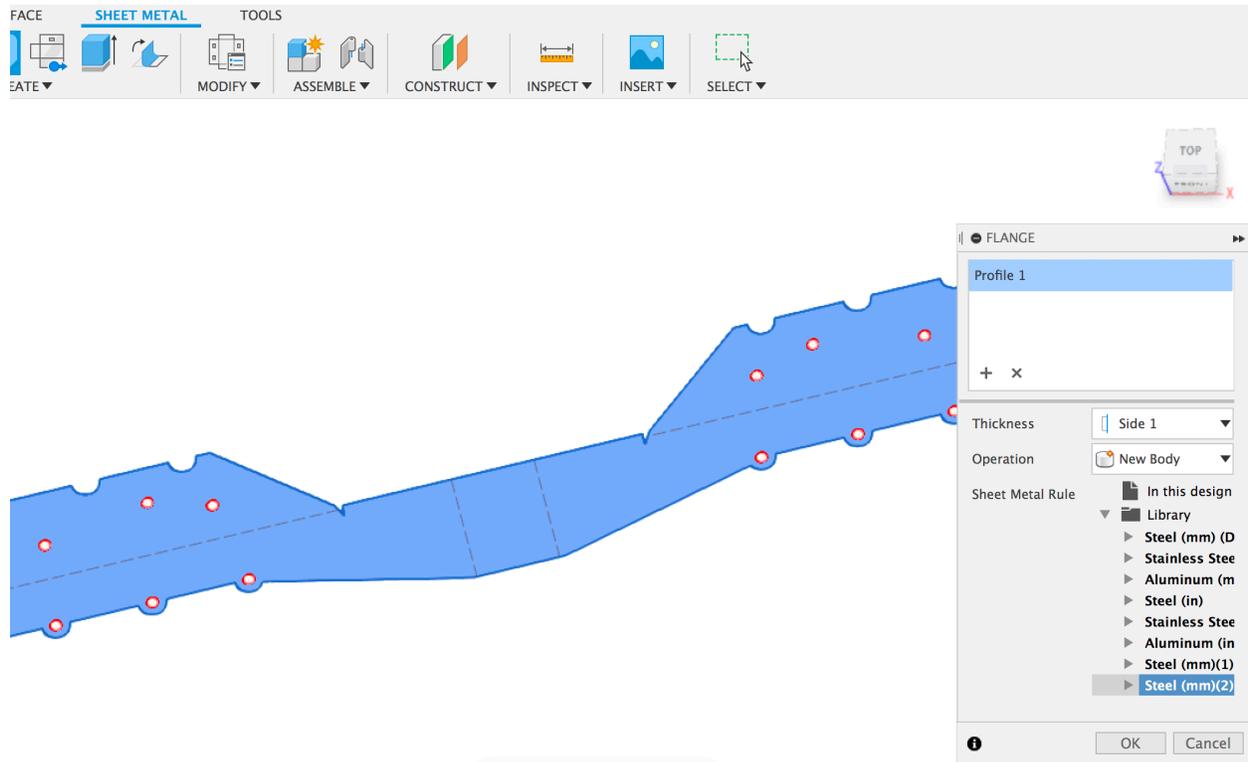


FIGURE 77 - SHEET METAL FLANGE FROM FLAT PATTERN

Make sure to select a sheet metal rule that matches your material.

**Tip:** Creating Bends (the next step) will be easier if you choose “Side 2” to extrude the material away from the sketch. This is not critical to do, however. The process will work even with the sketch on the back side of the part.

#### 4.1.3 Create Bends

The next step is to create Bend features, using the bend lines as input:

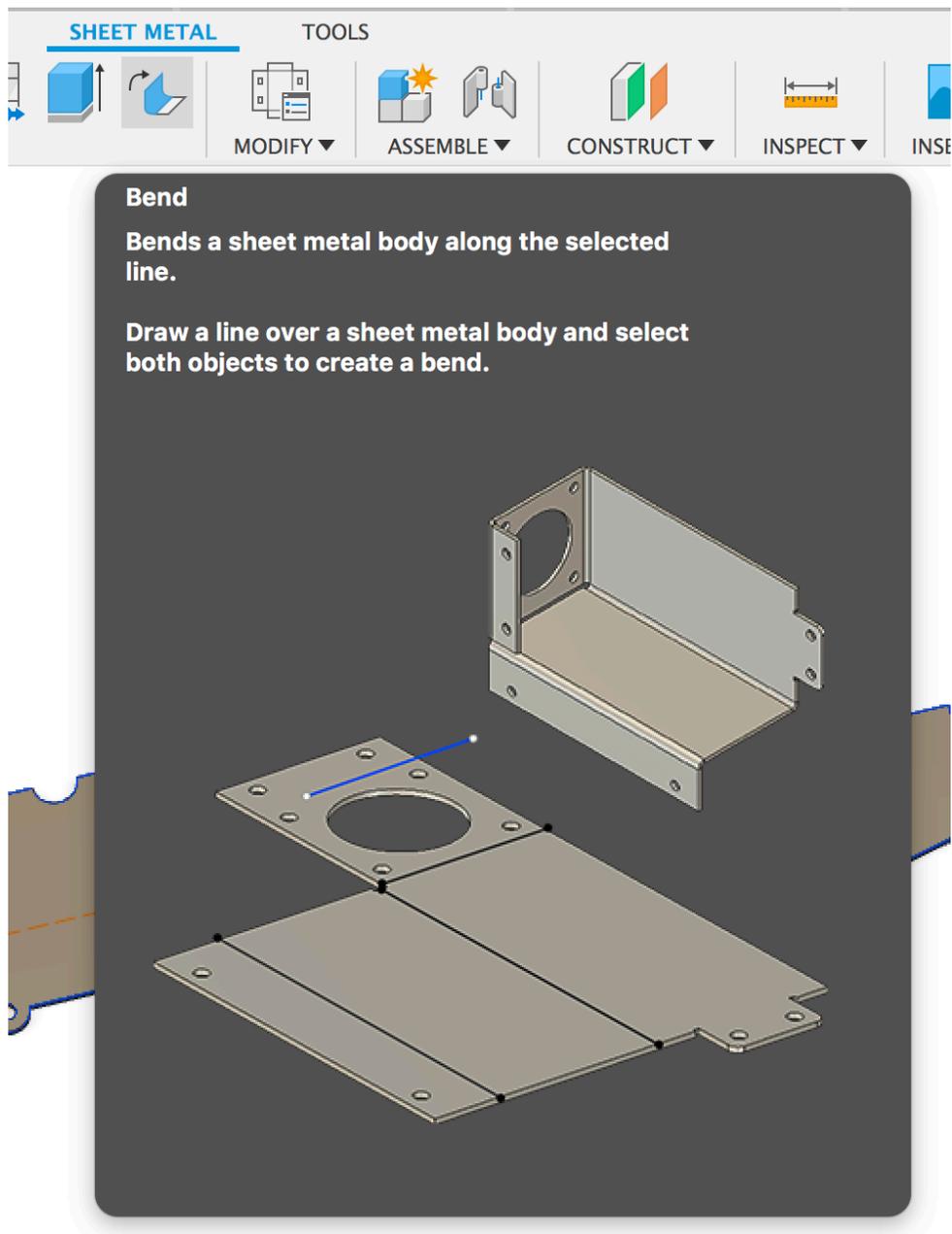


FIGURE 78 - SHEET METAL BEND COMMAND



*FIGURE 79 - SHEET METAL BEND COMMAND PREVIEW*

Then, just repeat this for all the bends in the part:



*FIGURE 80 - SHEET METAL PART FROM FLAT PATTERN*

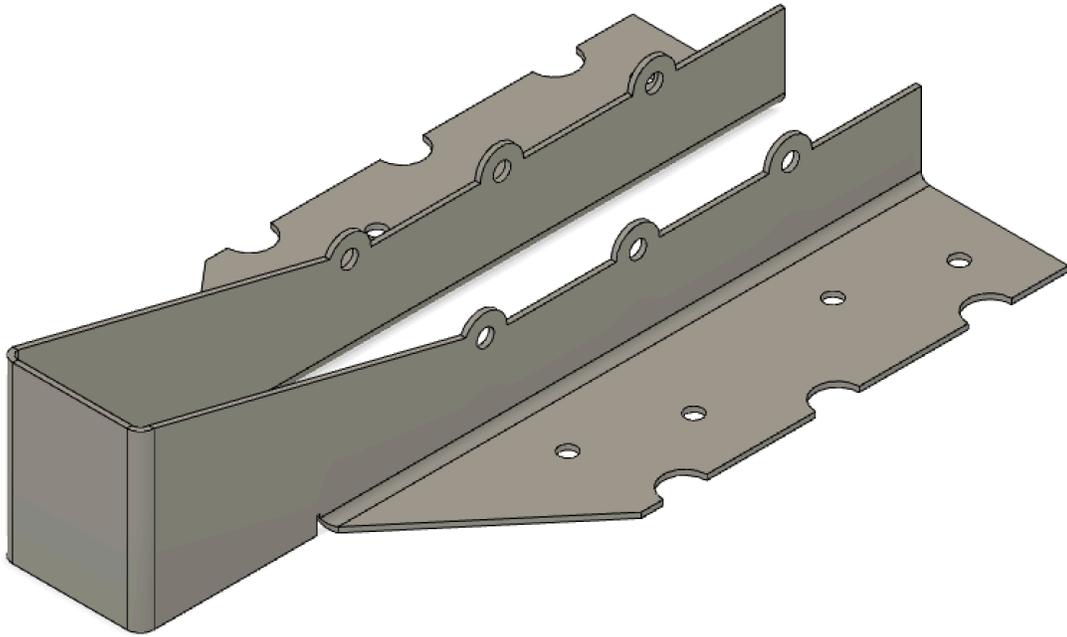
## **4.2 From a folded 3D part**

This workflow starts with a folded sheet metal part. Often this comes from another CAD system, and can be found in STEP format, or in the native format of the original CAD system. You want to convert this into a native Fusion sheet metal part. Note that this method requires the source to be a solid, BRep design. This will not work with surface models or with mesh models.

We'll use the same simple example here, but the concept extends to more complex parts.

### **4.2.1 Import the model**

The first step is to import the model. No matter the source format, it will come in as a “dumb solid” model:



*FIGURE 81 - IMPORT STEP FOLDED SHEET METAL PART*

#### **4.2.2 Convert to parametric design**

The next step is to convert the design to a history-based design. This is required because the sheet metal tools are only available in timeline/history-based designs.

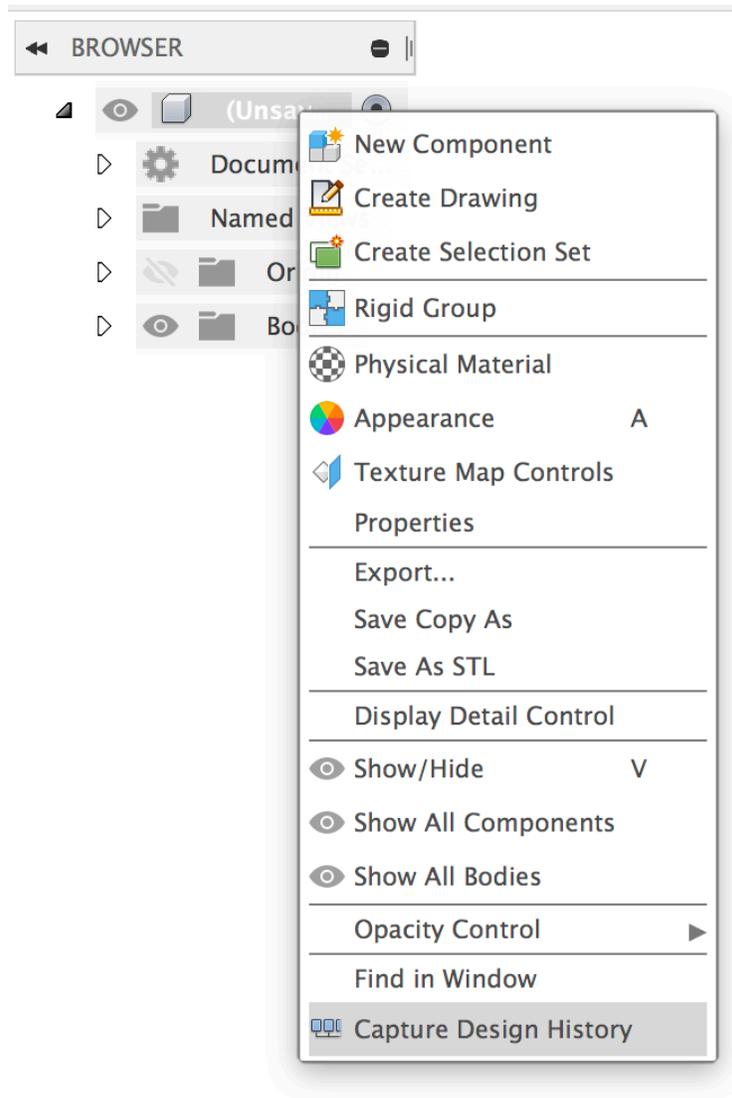


FIGURE 82 - CONVERT TO PARAMETRIC DESIGN

#### 4.2.3 Convert to a sheet metal component

The next and final step is to simply select the Convert to Sheet Metal command:

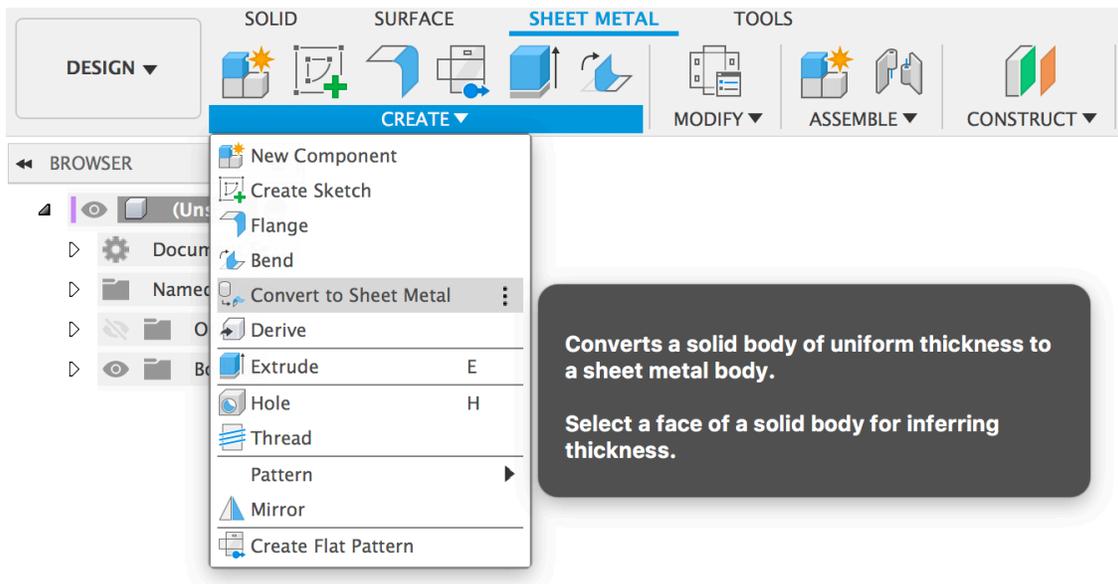


FIGURE 83 - CONVERT TO SHEET METAL COMMAND

This command does pretty much what it says it does – it detects a uniform-thickness part, checks that it follows the sheet metal design rules, and converts it to Sheet Metal:

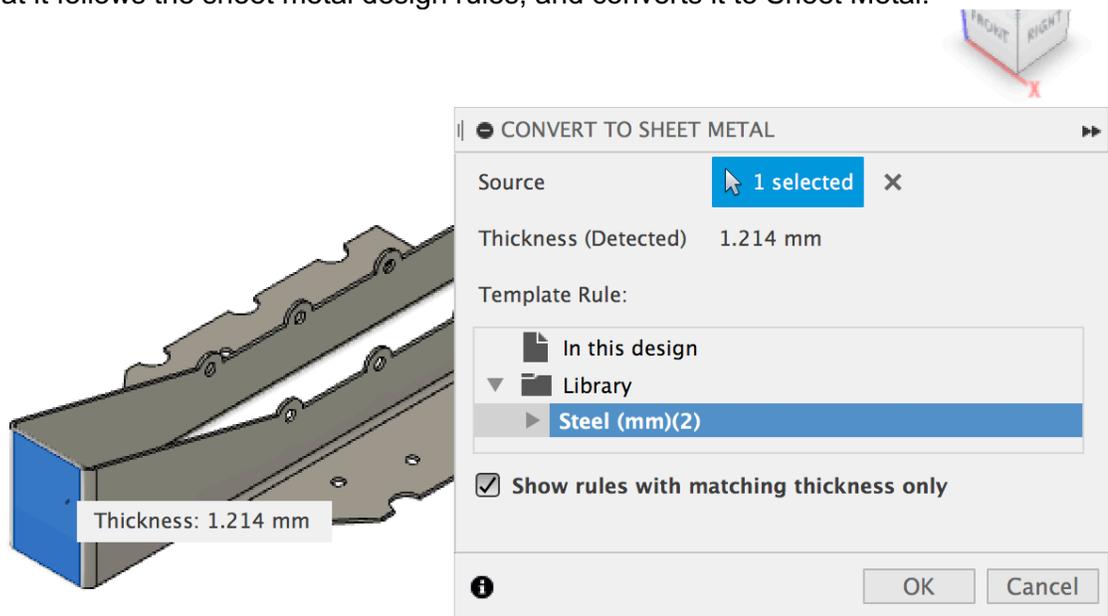
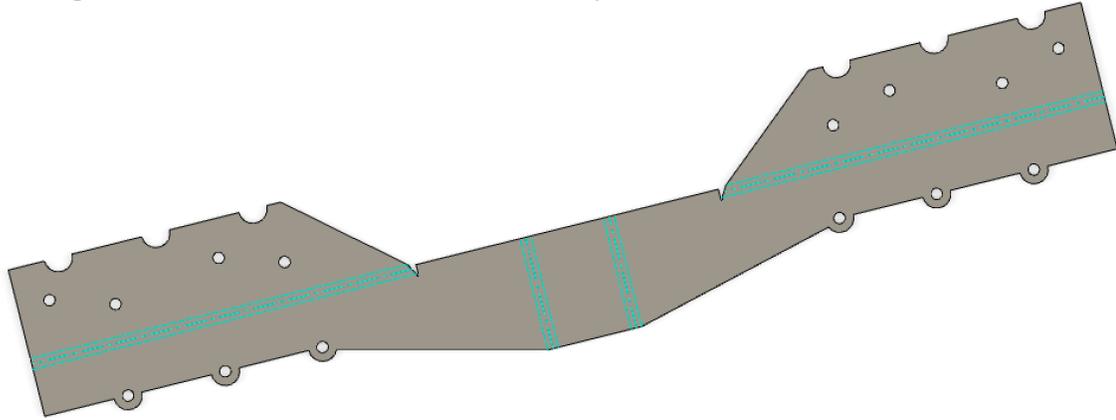


FIGURE 84 - CONVERT TO SHEET METAL

This command will select a default sheet metal rule, but you can choose a different one, if you desire.

Once this command is complete, this is a full-fledged sheet metal component. For instance, you can add flange features, unfold/refold, or make a flat pattern:



*FIGURE 85 - FLAT PATTERN FROM CONVERTED SHEET METAL PART*

**Note:** This command does not detect features in the imported data. That is, it will not create flange features for existing areas of the design, etc.