AUTODESK UNIVERSITY

PD20709

Fusion 360 - Modeling Tips from the Experts

Peter Doering
Trippy Lighting LLC

Jeff Strater
Autodesk Inc.

Learning Objectives

- Learn how to get the most out of Fusion 360
- Learn how to make your Fusion 360 designs more efficient and better performing when edited
- Learn how to make your Fusion 360 designs more stable, so you see fewer errors when making edits
- Gain confidence in your ability to approach any design problem using Fusion 360

Description

This class is for Fusion 360 software users who want to go beyond the basics and learn how to get the most out of their design experiences. You will learn the favorite tricks and design techniques of 2 experts. This class is for Fusion 360 users who understand the basics of sketching, parametric solid and surface modeling, and sculpting with T-Splines. We will help you avoid some common pitfalls, and you'll learn how to build more efficient and stable designs. This session features Fusion 360.

For an up-to-date version of this document, see: Fusion Modeling Tips from the Experts. We plan to keep adding new tips over time, so this becomes more of a living document than a static one-time handout.

AUTODESK UNIVERSITY

Your AU Experts

Peter Doering

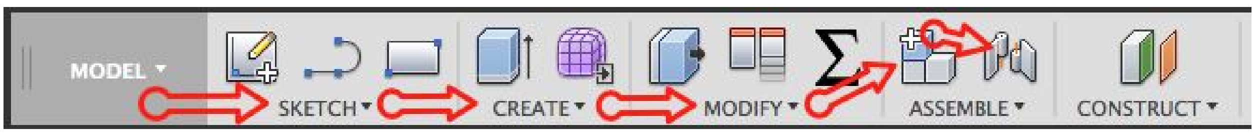
Is a mechatronics engineer, Autodesk Expert Elite and part of the Fusion 360 Customer Advisory Board. He has worked with computer graphics & CAD and software for 25+ years. His experience is in factory automation machinery, and technical product design. For his own company he designs and manufactures LED lighting systems.

Jeff Strater

Senior Software Architect in the Fusion 360 team. He has been with Fusion since the very beginning (anyone remember Project Freeway? Inventor Fusion?). His focus is on general modeling/sketching. Before that, Jeff was a developer and architect on Inventor, also before R1. In what little spare time seems to be available, Jeff plays guitar (badly), runs, bicycles, and uses Fusion modeling and CAM to make his own guitars (not quite so badly…).

Modeling Workflows - what to start with ?

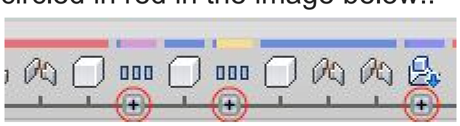
When starting to design in Fusion 360, the user interface similar to many other 3D CAD software suggests that one would start with one or more sketches, then use these sketches as the basis to create a 3D body maybe by using an Extrude or Revolve operation. One might continue with modifying that geometry, perhaps by adding a fillet or chamfer.
Finally that body might be converted into a Component so it can be assembled to one of the other components.



This is the workflow in a nutshell:
Sketch -> Body -> Modify/Add Features -> Component -> Assemble

However, as designs grow in complexity, with an increasing number of sketches, features, components and joints, the workflow above has a number of disadvantages due to the way data is structured in Fusion 360

- As the number of sketches/features increases it will become increasingly difficult to associate and find a sketch for a given geometry. Even worse, instead of editing an existing sketch, possibly another sketch is created to modify existing geometry.
- Components, when grouped are displayed in the timeline icons with a plus(+) sign circled in red in the image below..

  They can be expanded and collapsed similar to the functionality in the browser, which allows to also shrink the timeline considerably. Without sketches and features associated with a component there is nothing to collapse and the timeline will get unwieldy very quickly.
- The lack of structure makes such designs very hard to understand, even for the creator

A workflow that does not exhibit these disadvantages on the Fusion 360 forum has been dubbed :

Fusion 360's R.U.L.E #1 - or Component first.

When in doubt, before doing anything, create a component and make sure it's activated.

Component(Activated) -> Sketch -> Body -> Modify/Add Features -> Assemble

All objects created after activating the component such as sketches, bodies, construction geometry, joint origins, etc. are created in that component. Newly created components are automatically activated.

This has several advantages:

1. On activation the timeline is filtered to show only those items in the timeline that pertain to that component. That will make the quickly growing timeline much easier to work with.
2. Sketches, construction geometry, joint origins etc are created in the structure of the component.
3. If a component is exported to the data panel with "save copy as" this will also export the complete parametric design history.

Remember:

1. The joints in the "Assemble" menu only work with components.
2. Drawings can only be created from components
3. Only components show on the BOM.
4. Only components can be isolated.
5. Color cycling/color swatch only works on components.
6. The above points apply to assemblies as well

There are a number of other workflows that are perfectly valid but the first step would not be creating a component. For the most part these are top-down design workflows. All of the following workflows with the exception of the last one have one thing in common, they aim to create a component as soon as possible, so features can be collected in the component.

Skeletal design - One Sketch. Multiple Bodies.

Sketch -> 1+n Bodies -> Components (activated) -> Modify/Add Features -> Assemble

When you create a skeleton sketch that carries features of several parts of the design and is used to extrude or otherwise create several bodies that then are turned into components. That conversion into a component, however, should happen as soon as possible because features added to a body contained in a component are all added to that components design history.

T-Spline & Digital Carving

When the design starts with a T-Spline for example the exterior shell of a product that is then split into one or more bodies. Here also conversion into a component should happen as soon as the bodies are created.
T-Spline -> Component (activated) -> Modify/Add Features -. Assemble
or
T-Spline-> split body -> 1+n Bodies -> Components(activated) -> Modify/Add Features -> Assemble

Standalone & Linked Component (XREF)

Sketch -> Body -> Modify/Add Features

A design for a single component to be reused in other designs. No component creation is necessary in this case as that design when inserted into another design is inserted as as a component. This is also well explained in this 6 minute video tutorial. This can also be used for single 3D printed objects.

Editing Components - do they always have to be activated ?

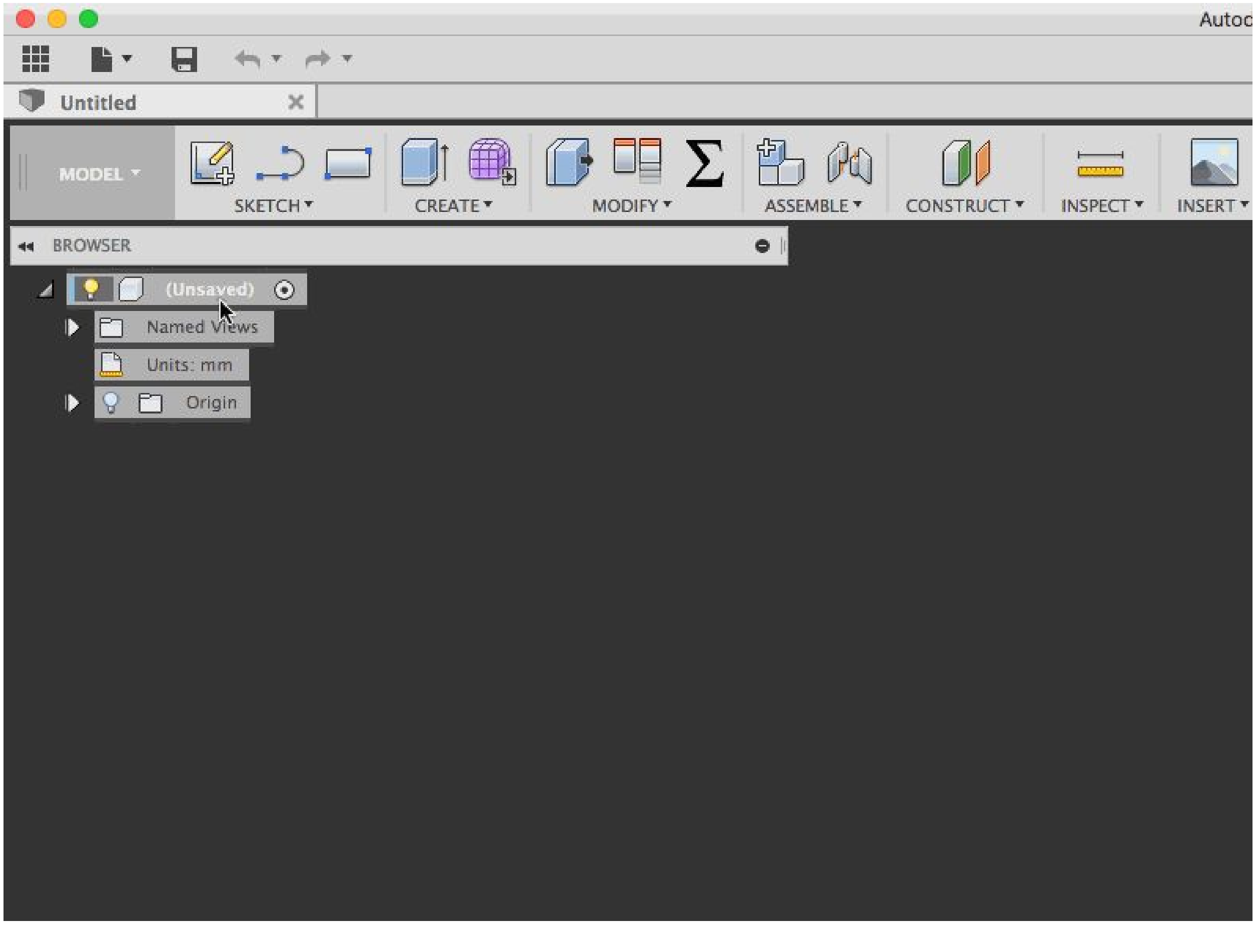
Generally speaking activating components before editing is advised and is a good habit to get used to. It is, however, only necessary when adding “objects” to a component. In essence if you are using any of the options in the menus highlighted in red in in the image below you should activate the component before doing so:

When you are editing existing sketches or features, component activation is not necessary (although it does not have any negative effects either). The “Modify” menu is not highlighted in red, because it entails modifying existing geometry. Fusion 360 is smart enough to determine which components the new features e.g. a fillet, or chamfer need to be added to without activation.

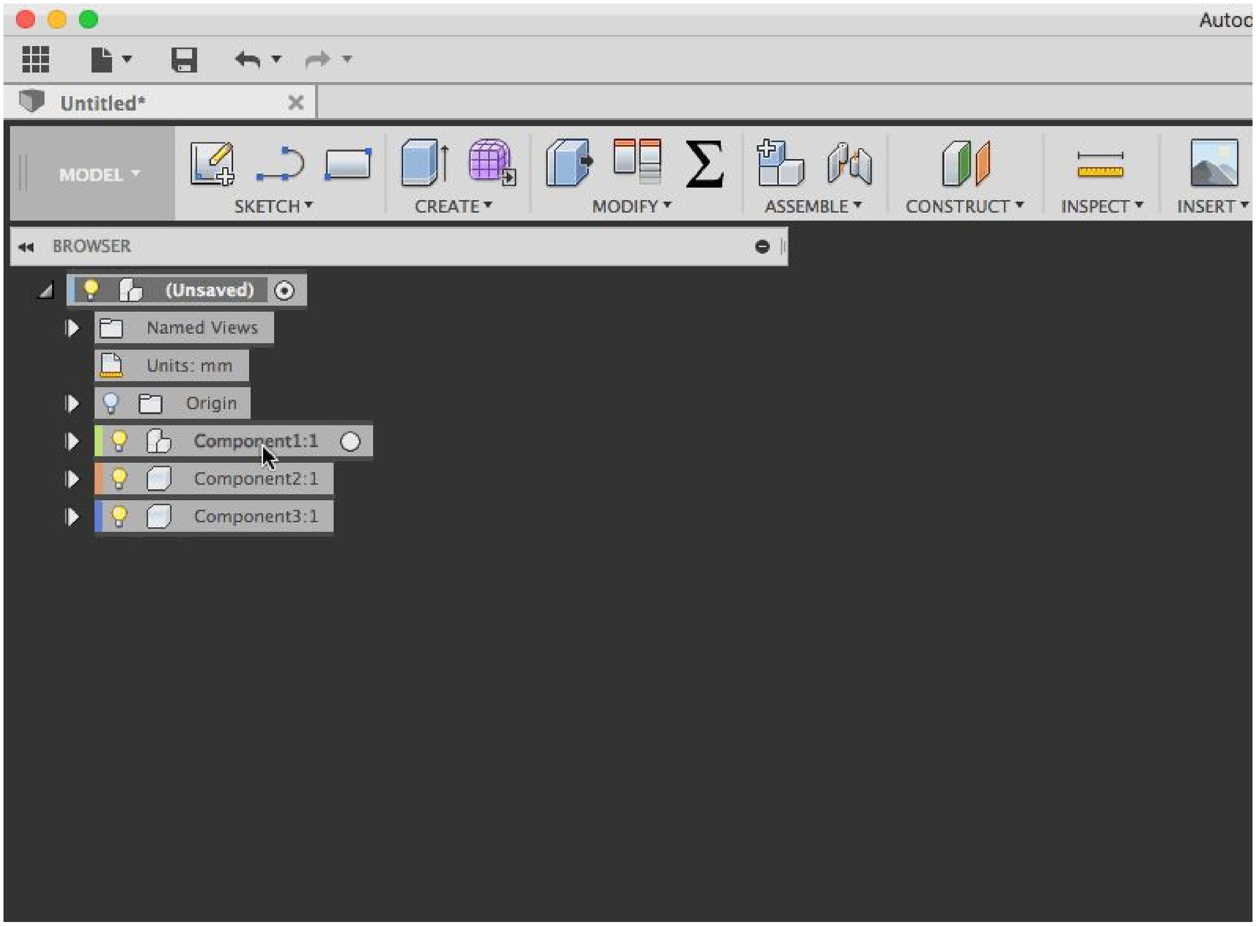
Related Topic: Bodies vs. Components in Fusion 360:

[Link to Fusion 360 learning resources](#)
[Youtube Quick TIP](#)
[Another web source](#)

The two methods of how to create components:



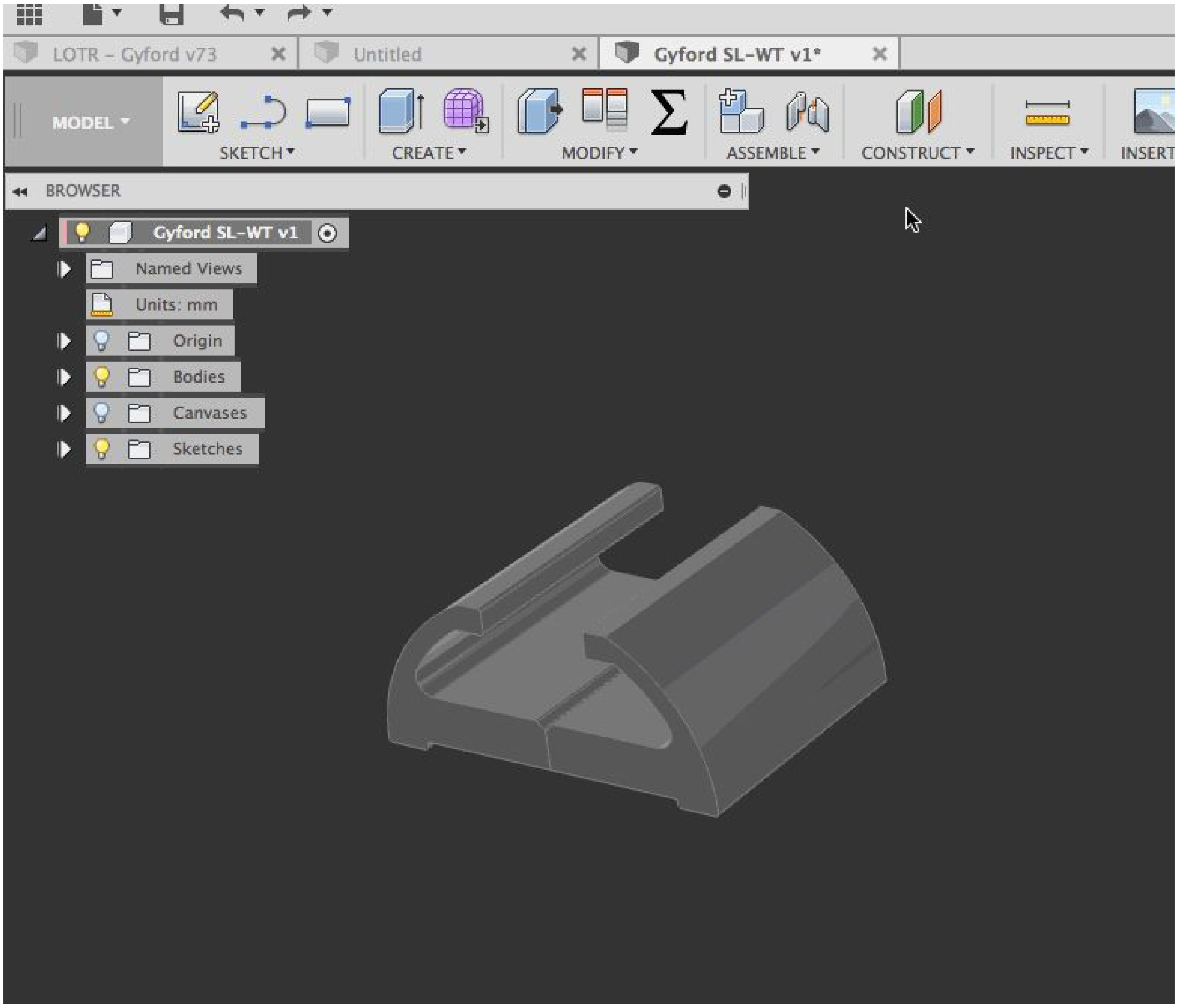
The two methods of how to activate a component:



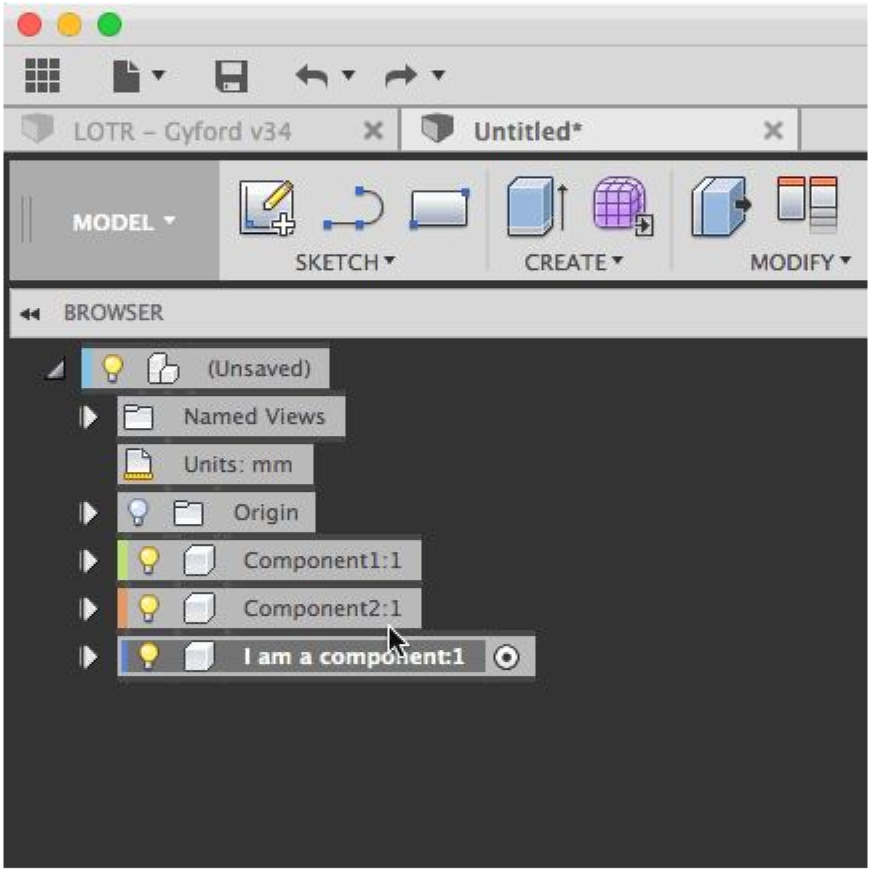
Components can also activated directly in the viewport.
Select the component in the viewport by double-left-clicking on it, then right-click and select "Activate" from the pop-up- menu.

Remember, single-left-clicking on an object in the viewport, a face, edge or vertex will select the body that object belongs to, not the component that body might be part of. The highlighting in the browser will also look different for the two scenarios.

If you forgot to create a component as the first step and already have a sketch and created one or more bodies from it, it might not be too late to still move these into a new component.



If you already have components and want to create an assembly from those, in the browser you can simply drag the components in the browser into another component.



Assembly structure

Now that we've created components and know how to navigate an assembly we need to think about how an assembly should be structured and what structure some of the workflows create that we discussed in the first section.

<table>
<tr>
<td>Assemblies should <u>not</u> be structured like this:

—+— Assembly
   |— Sketch
   |— Body
   |—+—Component
      |— Sketch
      |— Body
   |—+—Component
      |— Sketch
      |— Body

Bodies should not be in an assembly at the same level as components.</td>
<td>But like this: (Fusion 360's R.U.LE #1)

—+— Assembly
   |—+—Component
      |— Sketch
      |— Body
   |—+—Component
      |— Sketch
      |— Body
   |—+—Component
      |— Sketch
      |— Body

or like this: (Skeleton sketch at same level with several components)

—+— Assembly
   |—+—sub-assembly
      |— Sketch
      |—+—Component
         |— Sketch
         |— Body
      |—+—Component
         |— Sketch
         |— Body
   |—+—sub-assembly
      |—+—Component
         |— Sketch
         |— Body
      |—+—Component
         |— Sketch
         |— Body</td>
</tr>
</table>

Caveat when working with components:
Never use "Delete all features after history marker" unless you are absolutely sure these items are the very last items in the overall timeline of your complete design. When a component is activated it shows only a filtered view of timeline. The items shown may not be contiguous or at

the end of the timeline. “Delete all features after history marker” applies to the entire timeline, not only the items of the activated component!

Linked Components - XREFs

Fusion 360 does not differentiate between component and assembly files. This allows us to build a complete assembly with all components in a single file. Any given Fusion 360 file can be a component or an assembly of components.
When inserting an existing component from the data panel into the current design that component is automatically inserted as a linked component . If maintaining the link is not required the link can be broken. In that case the link should be broken before the design is saved. So the question is now:

When are linked components needed ?

1. When the component/assembly is to be reused in another design AND
2. Modifications to it are expected AND

   Note: When no modifications are expected e.g the part is a standard, off the shelf part such as a fastener or other catalog part, the link can be broken This should be done immediately after insertion, before the design is saved.

3. When the design the XREF is inserted into must reflect those modifications

   Note: In one-off designs that need to reflect an as-built state, automatic updates might be undesirable.

Even when knowing upfront that these three conditions apply, resist the urge to start the design with the component linked into a design. Most components that are to be re-used are developed in context and undergo many changes. Having to frequently update a design for changes made to a linked component can be cumbersome and even a hassle and unnecessarily slow the design process down. Consider waiting until the design for the reusable component is mature/complete and then export it to the data panel with “Save copy as”

Caveat: Currently, when exporting a component or assembly from a design into the data panel (with “save copy as”) it is exported as a stand-alone design. There is no automatic link back to the originating design.
Linked components are also the basis of distributed designs in collaborative environments when several people concurrently work on a design. In such cases having thoroughly developed and defined interfaces between components and assemblies helps to keep interferences and headaches at bay.

Current Limitations

The ability to use linked components in Fusion 360 was only added relatively recently. Currently there are some limitations:

- **Deleting a linked part from a design and then from the data panel.**
  When a component is linked into a design and the design is then saved, the component is linked to that version of the design. If the link to the component is broken and the design is saved again, the current version saves the state with the removed link, however Fusion 360 still maintains the link to the component in the previously saved version. The effect is that even if the component is not needed anymore in the current version and deleted, it still cannot be deleted from the data panel.
  If you are working on a design and go through a number of iterations to find the right part for your design, the data panel can become littered with unneeded objects that cannot be deleted quickly.

- **Changing object visibility in a linked component**
  When inserting a single linked component the visibility of the entire component can be change, however, the only objects within n that inserted component that allow changing their visibility from within the current design are joint origins.To cheng ethe visibility of sketches, construction geometry etc. in a linked component, that component has to be opened separately, edited tochenge visibility and than saved. Then the current design has to be updated to reflect that change. This will increase the version count for the linked component once for each editing operation.
  In case an assembly is inserted the visibility of the individual components can be changed. If you want to change the visibility of an object in one of the components in that inserted assembly then you'll have to open that component separately….
  And so forth… Deep Update alleviates some hassles with updating such changes but it is easy to see that using linked components too early in the design process requires a lot of forethought and often is not advisable.

- **User parameters** in linked components are not accessible from the current design.

Configurations - light

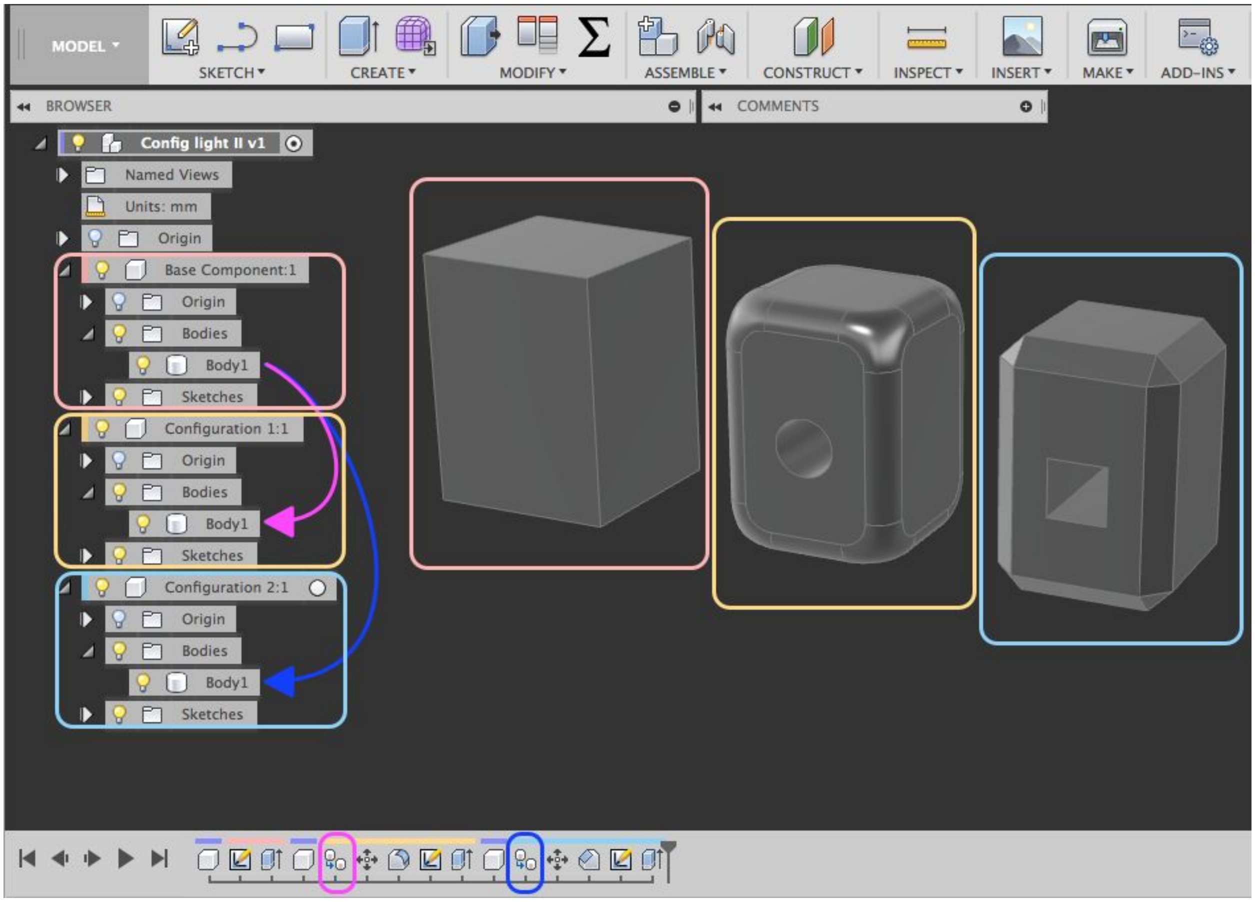
Wouldn't it be nice if you could design a component and a few variations of it, without having to redesign all the shared features for each different variant of that component but instead just add the additional features needed or that variant to the existing base design ?
In other CAD systems that feature is called Configurations. While a full featured version that also includes is not yet part of Fusion 360's feature set there is a workflow that that could be named Configurations - light.

1. Create an empty component #1 and make sure it's activated. -> Base Component
2. Create a sketch and then body #1 that holds all the common/shared features.
3. Create an empty component #2 and make sure it's activated.
4. Copy/paster the body #1 from component #1 into component #2.
5. Add the additional features to the body in component #2 -> Configuration #1
6. Create an empty component #3 and make sure it's activated.
7. Copy/paste the body #1 form component #1 into component #3.
8. Add the additional features to the body in component #3 -> Configuration #2
9. ….

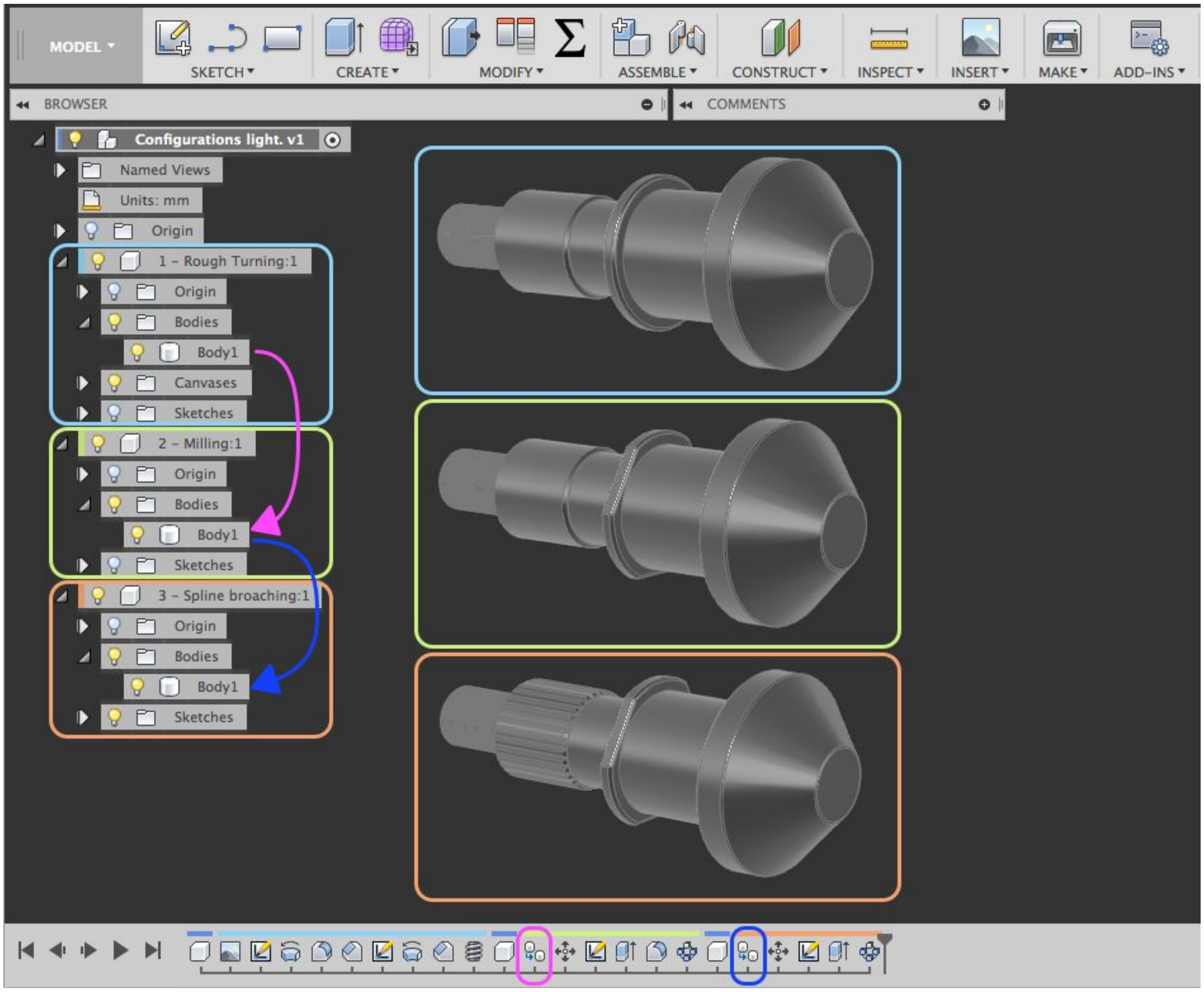
Of course the workflow above has a base component that you might not want to be include in the BOM. You can remove it (not delete) it from the design using the equivalently named function. This will remove it from the timeline from that point onwards and it will not be included in the BOM.

The workflow above is fully parametric:

- Dimensional changes to the sketches in the base component will be reflected in all other configurations.
- Features can be modified and added to the base component and be reflected in the configurations.



If a feature change must be reflected in all configurations, the timeline marker needs to be rolled back to before the copy/paste operation in step #4.
The image below shows another application of the configurations - light approach. This is useful for situations where for example a part goes through several manufacturing processes and each process needs a separate drawing, e.g. because some processes might be outsourced. Instead of copy/pasting the body from the first component into several others more or less in parallel, it copies the body from the first component into the second, then adds features and then copies the resulting body into the third component etc.



Copy/paste a component vs. copy/paste a body

It might help to visualize the underlying concepts between using copy/paste and copy/paste new when trying to duplicate a component and copy/past when trying to duplicate a body.

Sketch Tips:

How to get the most out of Fusion sketches with the least amount of frustration

This section discusses Fusion Sketch. While Fusion does not have the fastest and most powerful sketch environment in the industry, you can make it work well for you.

Some history of sketching in Fusion: When we started to develop Fusion, one thing we wanted to do was to handle sketches differently than other sketch tools. We wanted to develop a unified 2D/3D sketch environment. To do this, we had to develop our own sketch solver. This is something that has not been done in the CAD industry before. Most CAD tools use the same solver, which requires a separate 2D and 3D sketching environment. But, the Fusion team adapted their assembly solver to also be applicable for sketch solving. In the future, this will allow Fusion to have a truly unified 2D/3D solve.

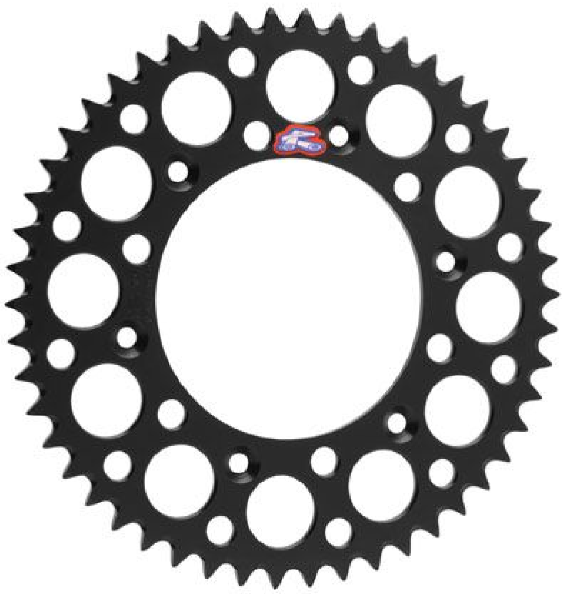
Tip 1: Simple is better than complex

In general, Fusion will perform better with a larger number of smaller sketches, and with feature patterns instead of sketch pattern. The more complex geometry and constraints you have, the more frustration you will feel. Performance with a simple sketch will be better, and the solve results will be more stable. These are 3 techniques to help you achieve simpler sketches:

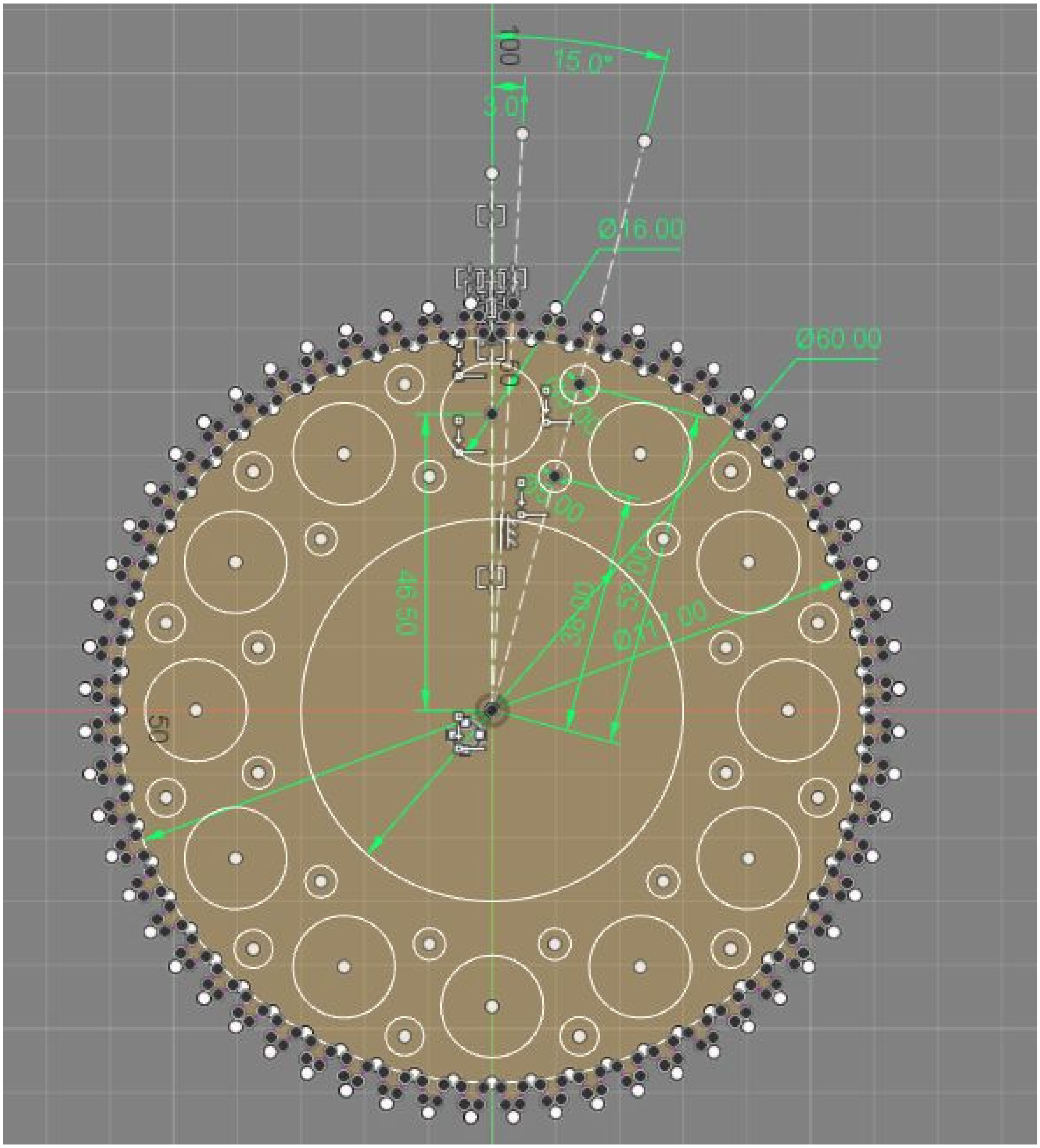
1. Use many small, simple sketches, instead of one large, complex sketch
2. Use feature patterns instead of sketch patterns
3. Use Fillet feature instead of Sketch Fillet

There are several ways you can achieve this: One is to separate sketches for each feature, rather than drive multiple features from a single sketch, and the other is to stay away from sketch patterns - use feature patterns again.

Example 1:

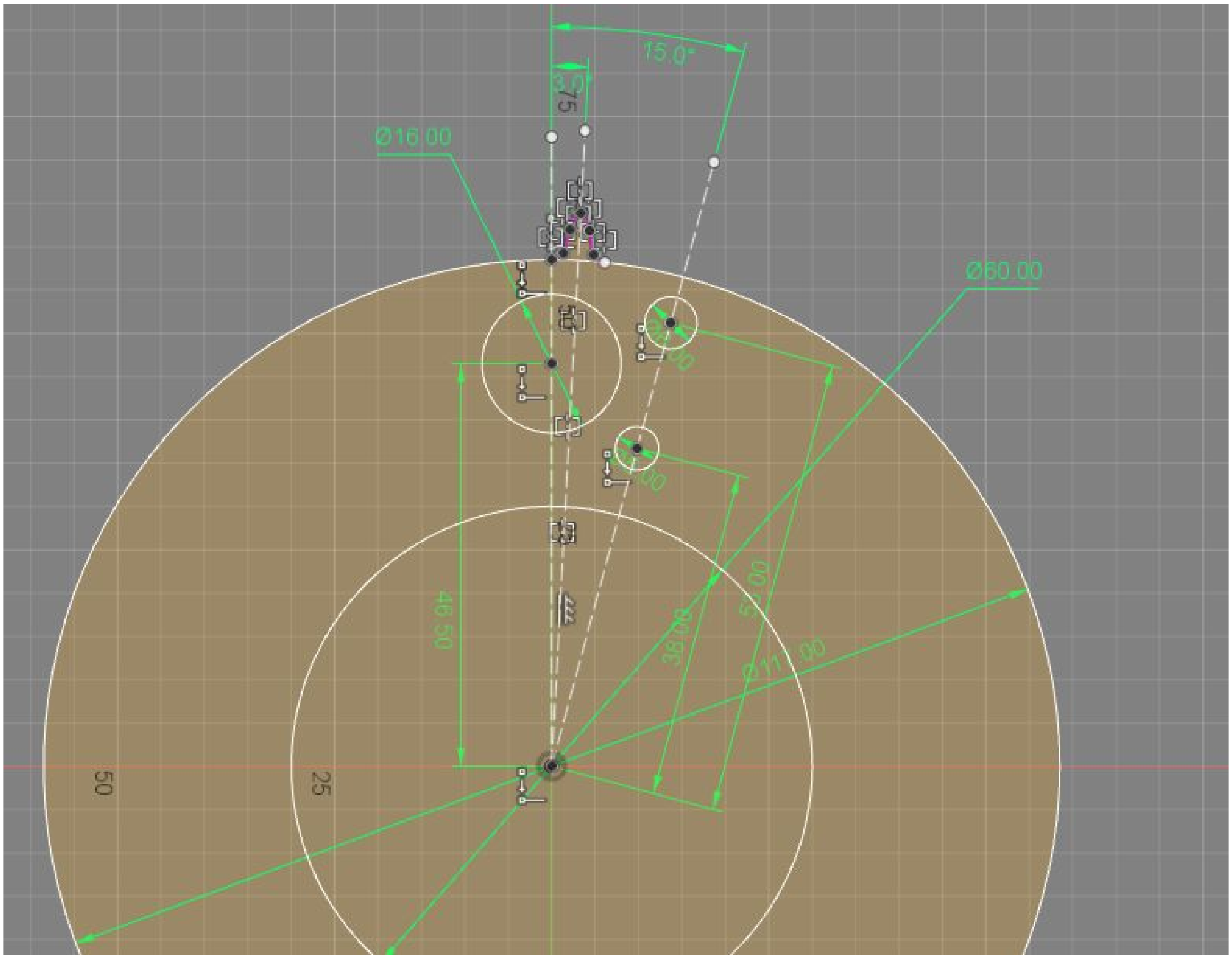


You could build this design with a single sketch:

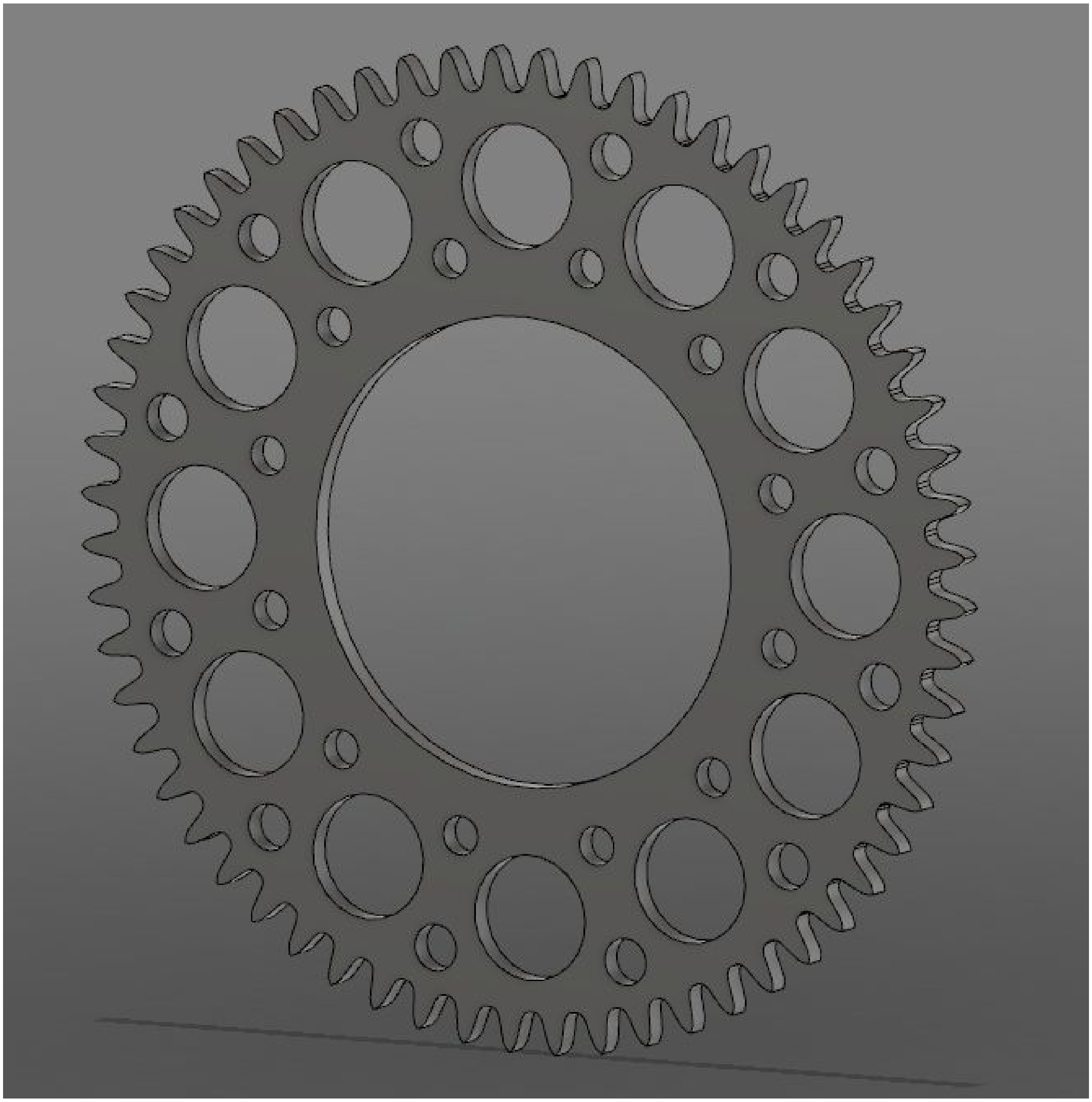


This design uses one sketch mirror, and two separate sketch patterns, one for the holes in the outer ring, and a separate for the gear teeth. This results in pretty bad sketch performance.

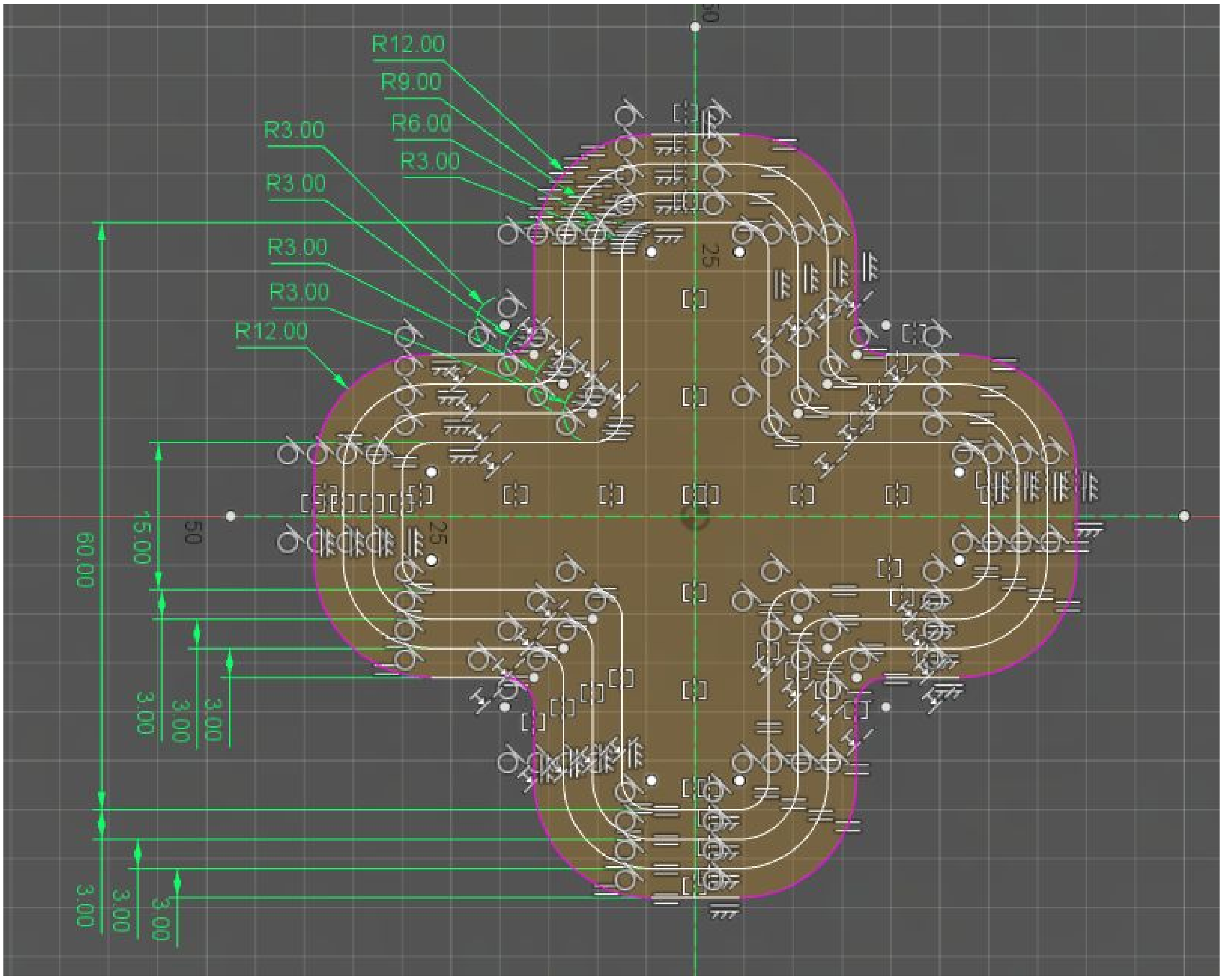
There is another approach to this design that uses a much simpler sketch:

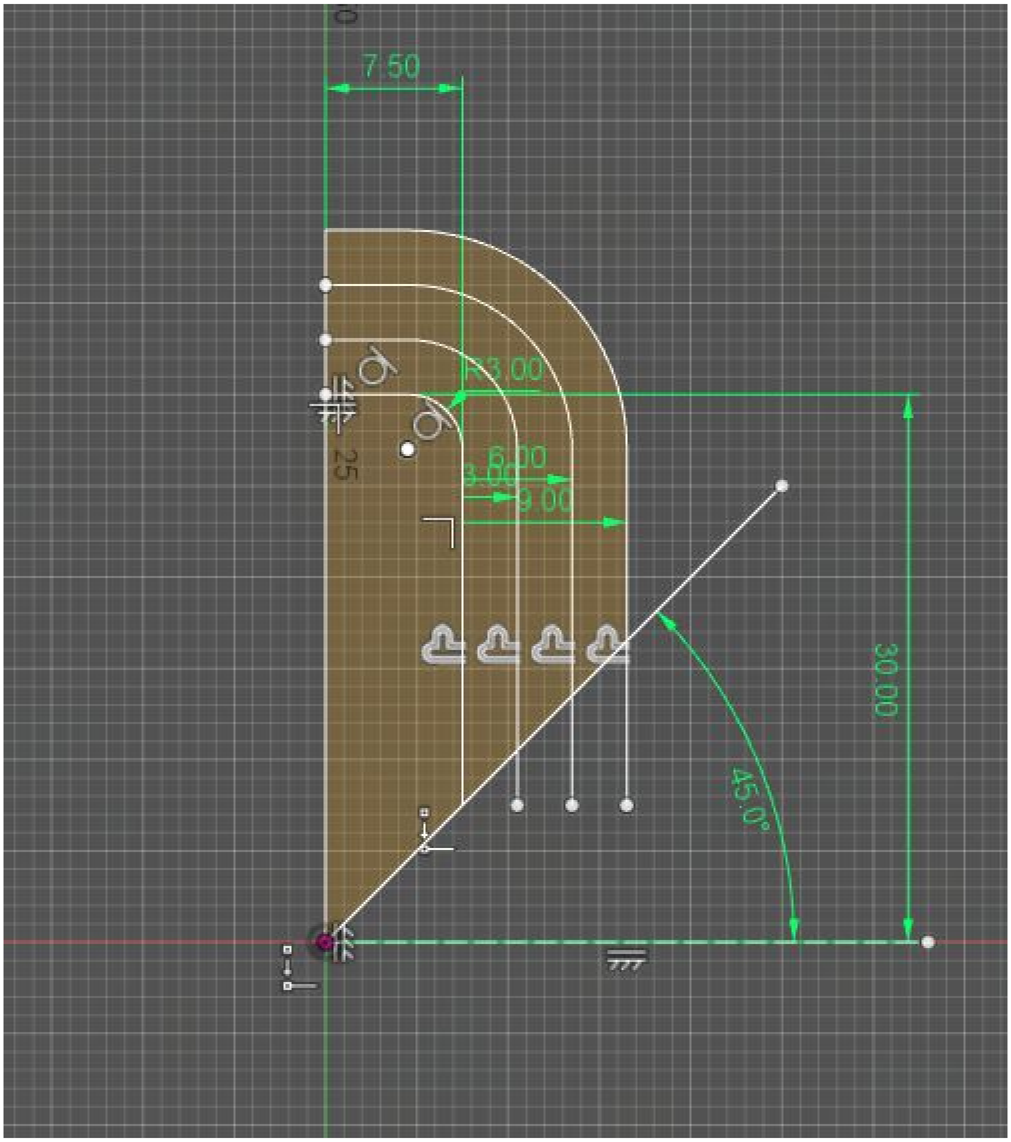


But which can still produce the same results, using feature pattern instead of sketch pattern:



Example 2:

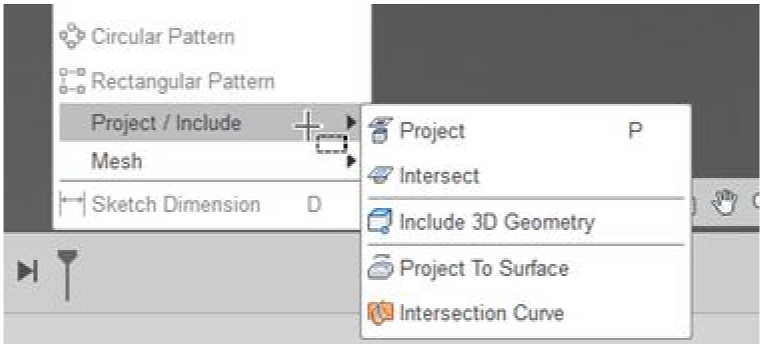

R12.00
R9.00
R6.00
R3.00
R3.00
R3.00
R3.00
R3.00
R12.00
60.00
15.00
50
25
3.00
3.00
3.00
3.00
3.00
3.00



With this simpler sketch, and some feature pattern and mirror features, you can achieve the same results as with the more complex sketch.

Tip 2: What are all those Project commands all about?

There are several related commands in the sketch Project menu:



What are all these commands, how do they differ, and when should you use each?

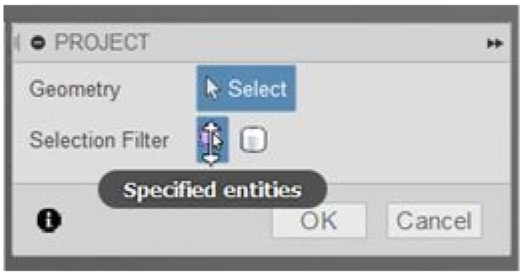
First, you need to understand that each of these do some flavor of creating sketch geometry that is dependent on some other object in your design. This means that you have created a relationship to that object. This can be very useful, but at the same time, it can limit you in the future. For example, in the screencast below, you can see that, while creating this very simple model, in sketch2, we project an edge of Extrude1. This can provide the ability to precisely position the circle in relation to this edge. However, this means that you can no longer drag sketch2 in front of Extrude1 in the timeline.

Screencast: [Sketch Project creates relationships]

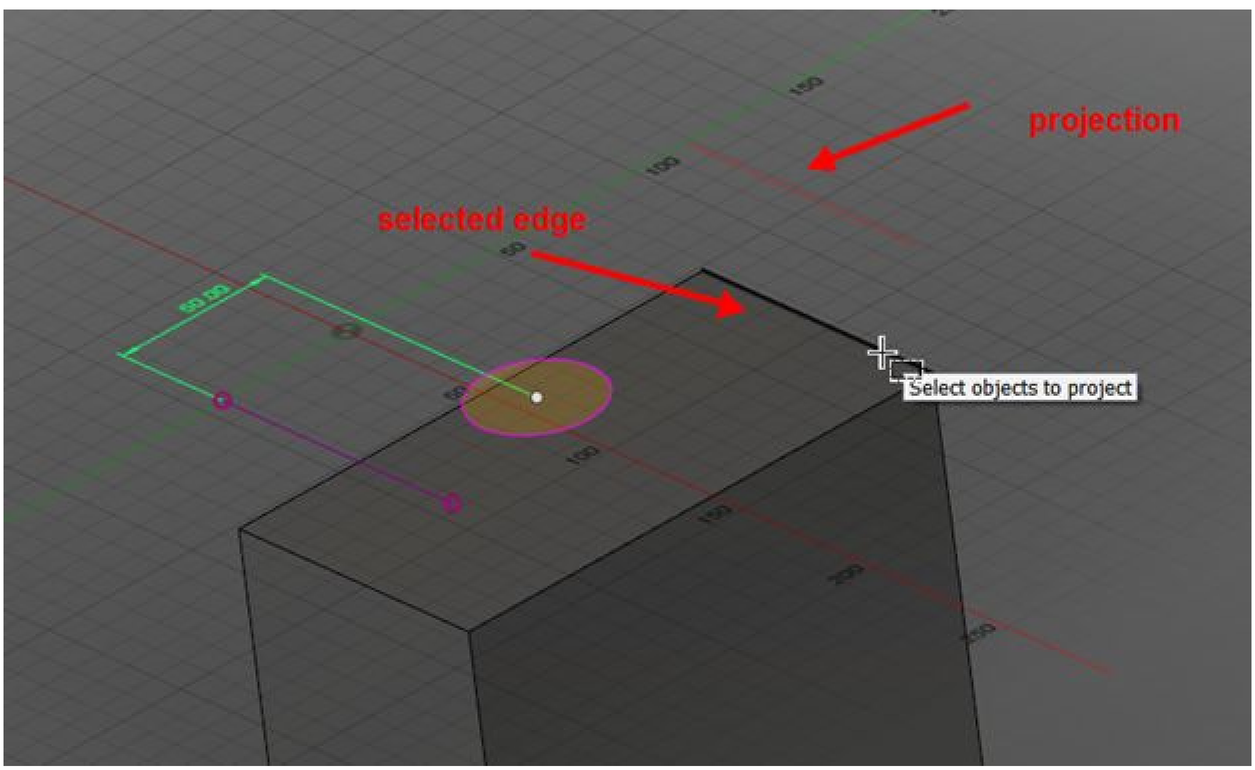
The Project command:

This is the workhorse of the Project commands. This command can create two types of projection:

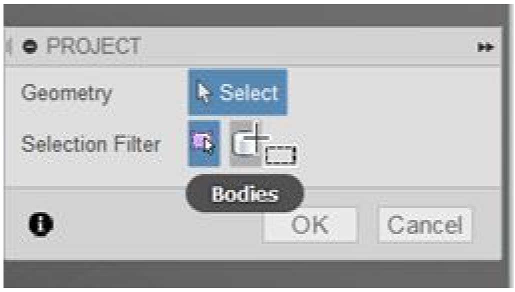
1. In "Specified Entities" mode:



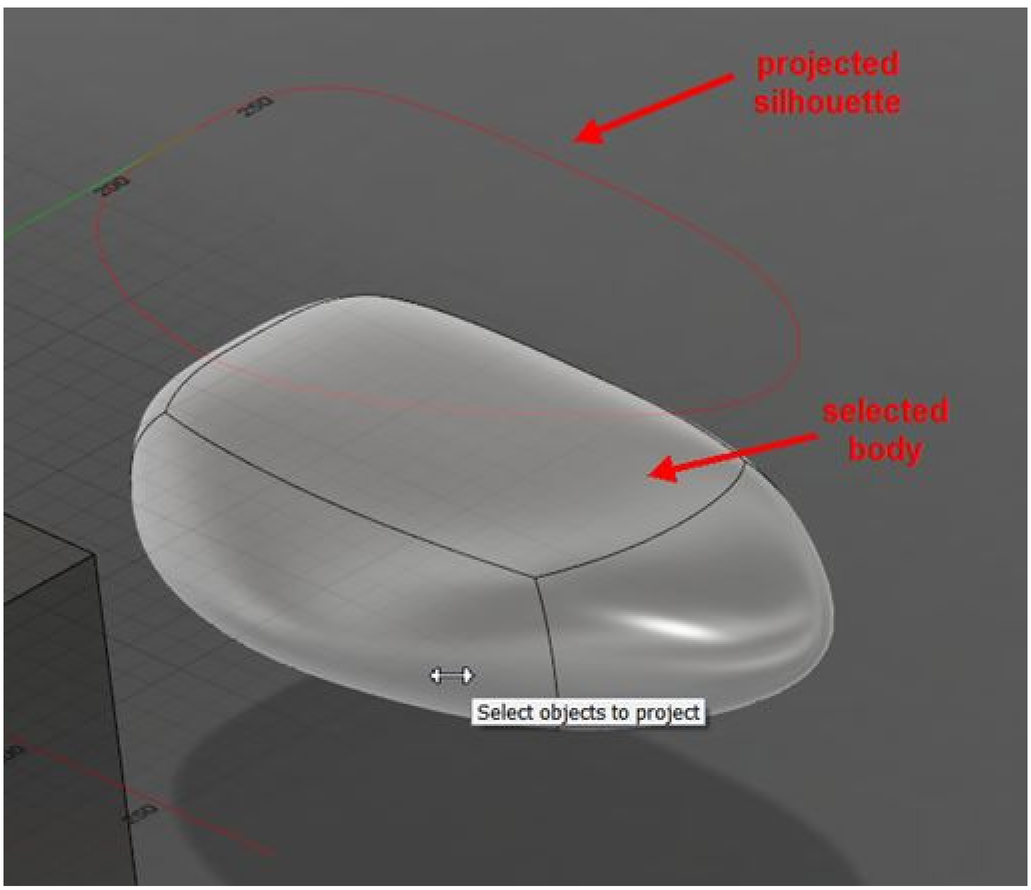
This command projects the selected object into the sketch. The object can be a face, edge, vertex, work geometry, other sketch geometry. The result is a 2D sketch curve or point, and is the geometric projection of that object into the sketch:



1. In “Bodies” mode:



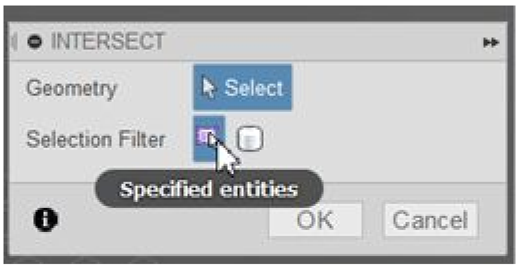
This command allows you to select a body. The result is the silhouette edge projection (meaning the outline that you would see if you looked straight down on the sketch from above) of that body into the sketch:



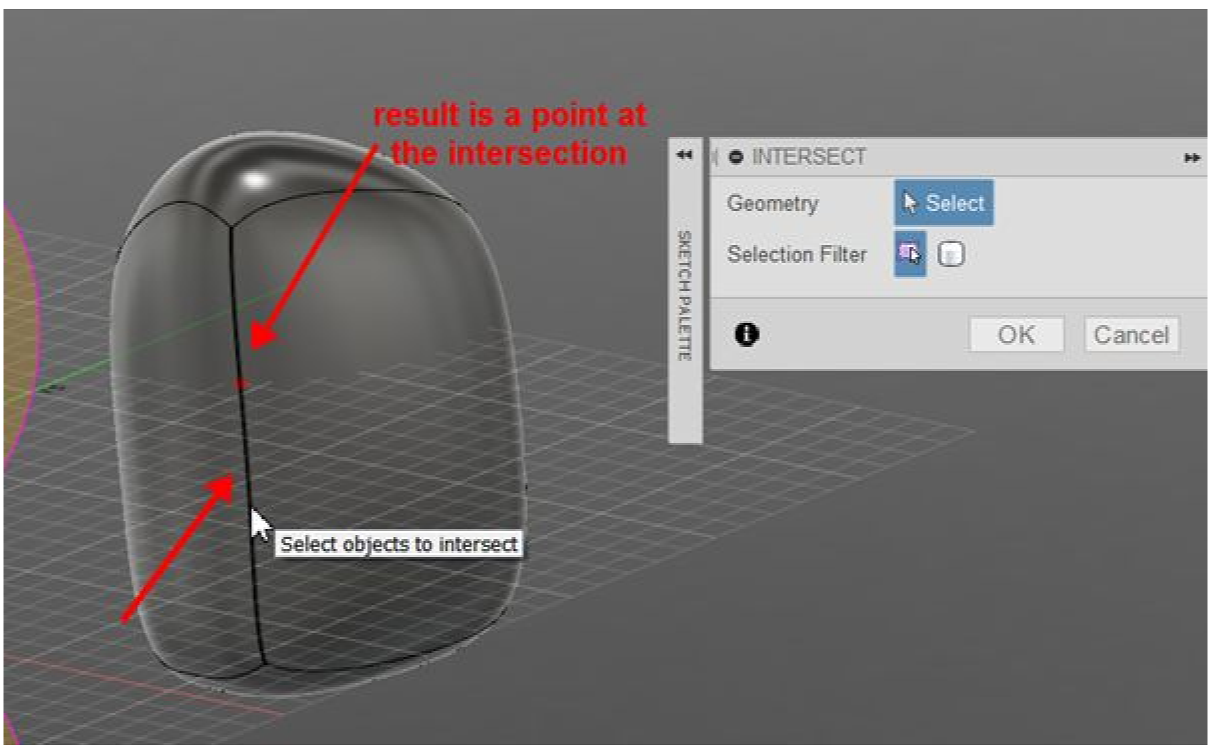
The Intersect command:

This command also has two basic functions:

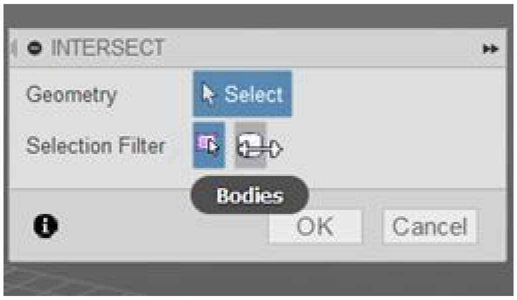
1. In “Specified Entities” mode:



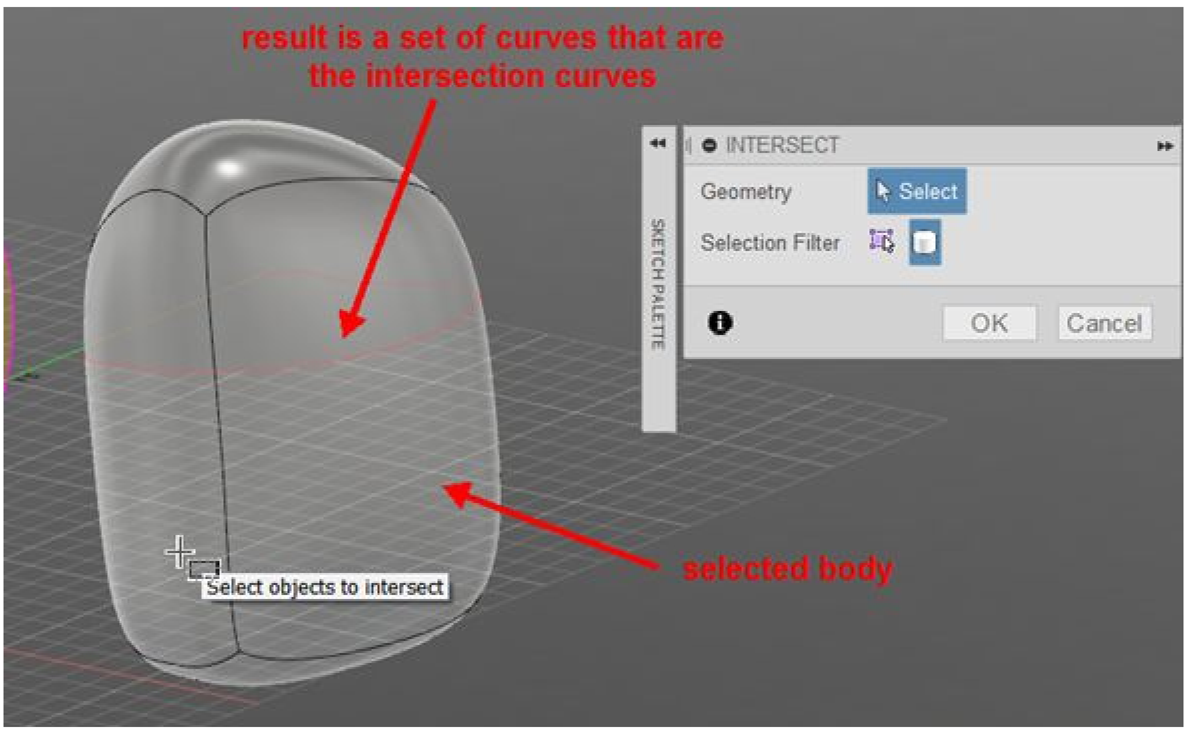
you can select objects such as edges, faces, etc, and the result will be the intersection of that object and the sketch plane. For instance:



2. In “Bodies” mode:



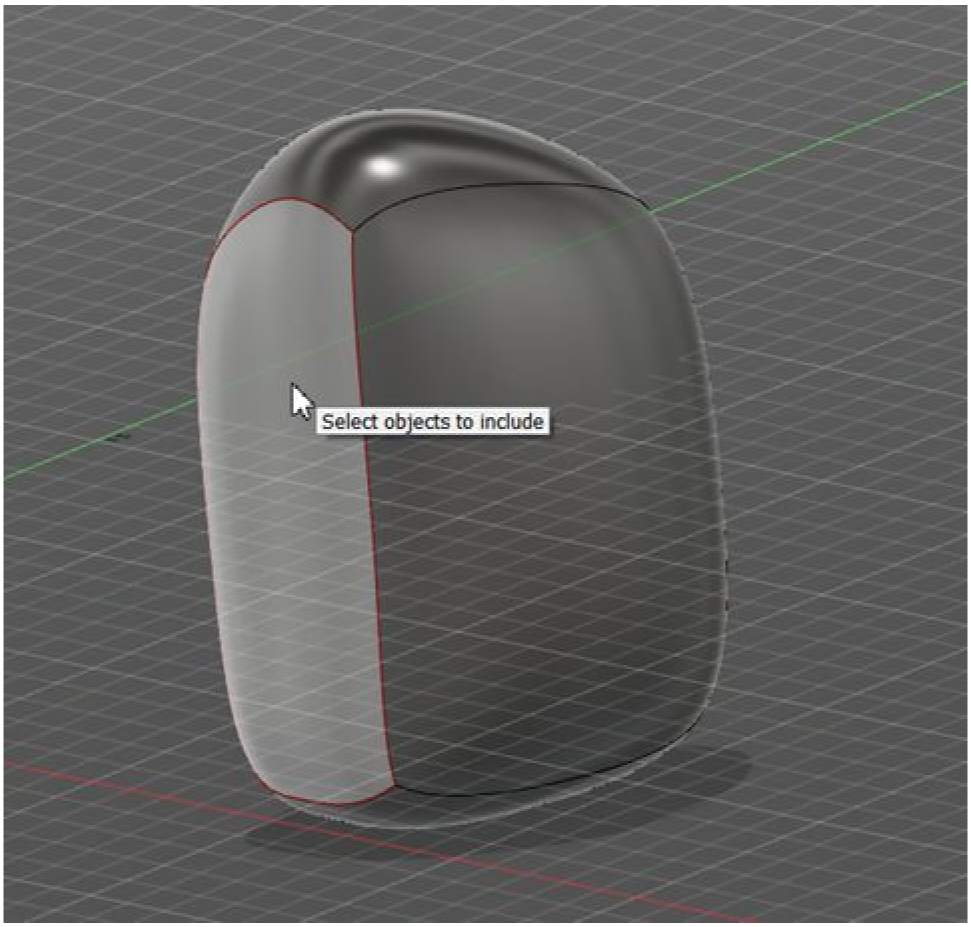
You can select a body. The result is the intersection of that entire body and the sketch plane:

3. 

The Include 3D Geometry command

This command pretty much functions as its name implies: It allows you to select objects such as faces/edges and produce a 3D curve or curves that is a member of the active sketch that is the same as the selected geometry. This is useful if you want to create a 3D line or spline that is tangent to an edge, for instance.

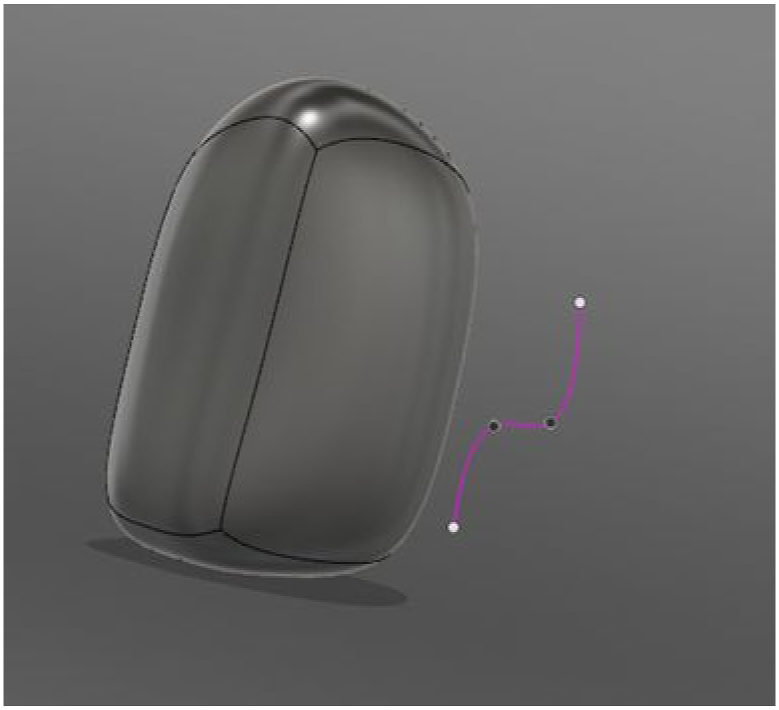
As an example, selecting this face will create 3D curves for each edge of that face:



The Project to Surface command

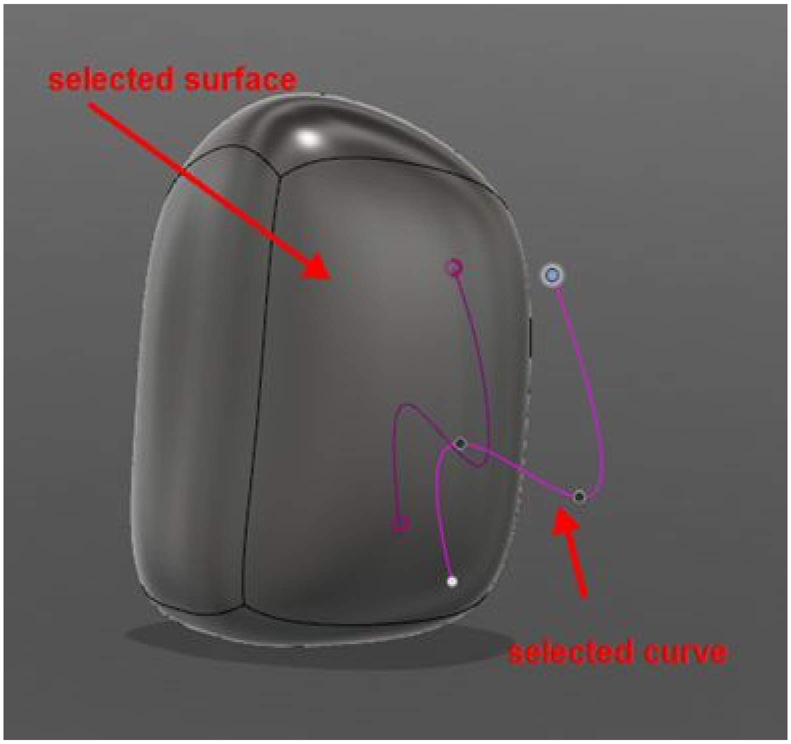
This command allows you to project a curve from one sketch onto a set of selected surfaces..

For example, consider this model:



Say, for example, that what you want to do is to project this curve onto this surface (perhaps for the purposes of sweeping a groove into the surface).
The screencast shows how you would do that: **http://autode.sk/2daZ8qc**

Here is the result:



More tips about the Project Curve to Surface command:

- This is not a “wrap” projection. It is a bit limited as to what you can do with it. You cannot, for instance, wrap a curve all the way around a cylinder
- You have to have two sketches for this command. This is a bit cumbersome, but there are reasons for it (those pesky relationships again)
- And, yeah, it’s weird that you have to pick a plane for a sketch that you intend to be just 3D, more on that later
- The different projection modes:
  - Along Vector is the most common. If no vector is selected, the sketch normal is used. Alternatively, you can select a work axis, line, etc as the direction. This gives you some flexibility in the results
  - Closest Point. This one, IMO, is less useful. It takes each point on the source curve, and finds the closest point on the selected surface.

My recommendation is just to experiment and see which gives you the best results

The Intersection Curve command

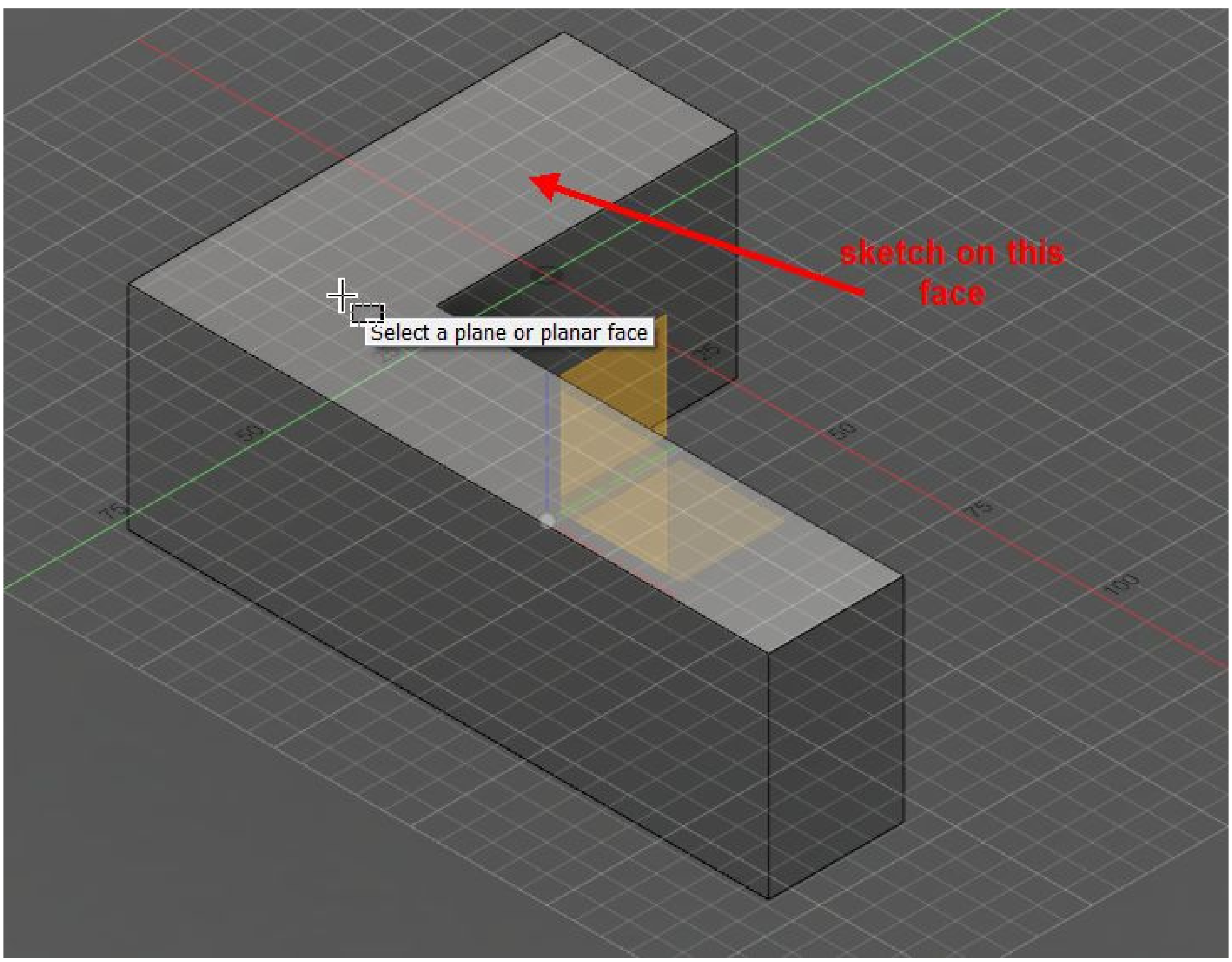
This command is not my favorite. It's complex to use – I can never remember how to use it. Basically it creates an intersection with a curve and one or more surfaces. Here's a screencast:

http://autode.sk/2dSrPpZ

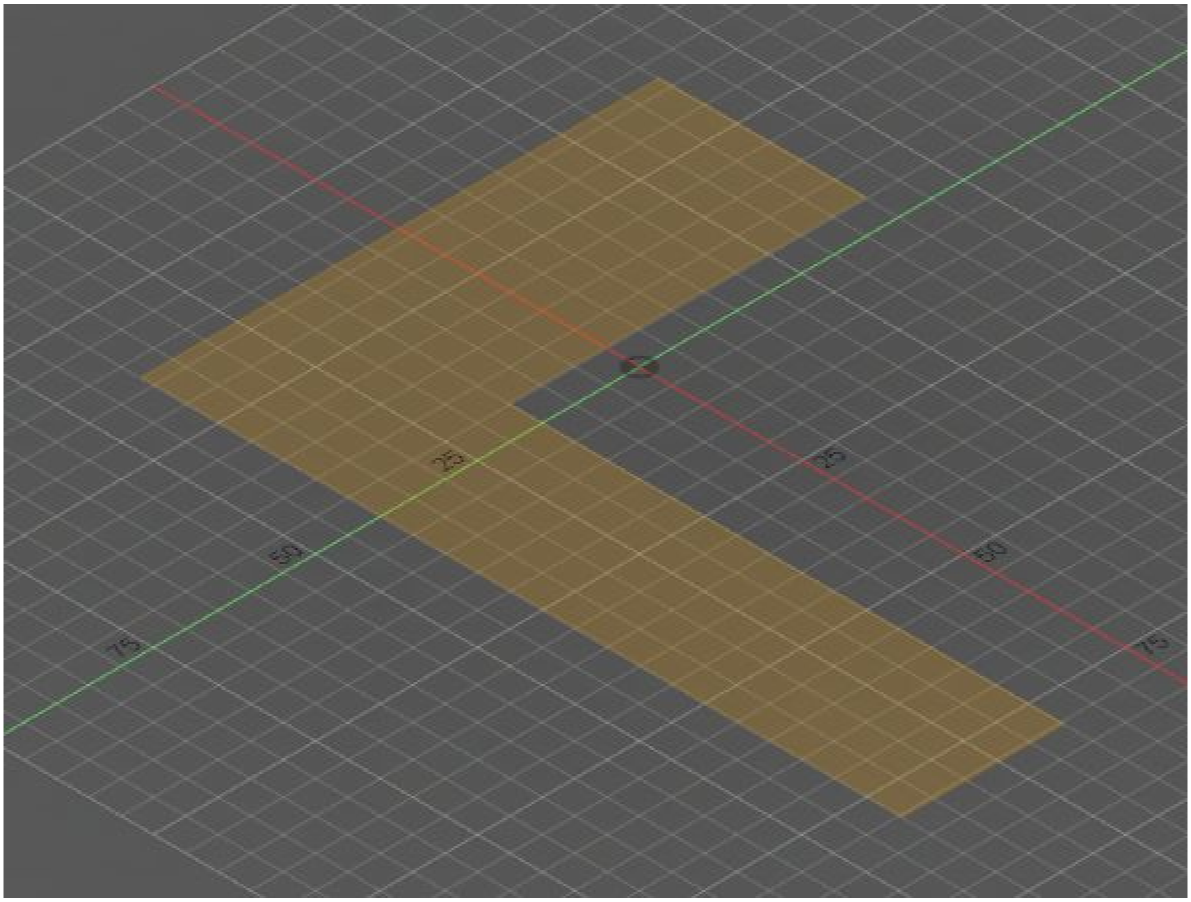
But, really, I don't find it all that useful, so I never use it.

Sketch Face projection

When you sketch on a BRep face, the edges of that face are automatically projected into the sketch. The curves themselves are not drawn, but you can tell projections have happened, because the profile is recognized:



And you can tell that the geometry has been projected, if you turn off the body:



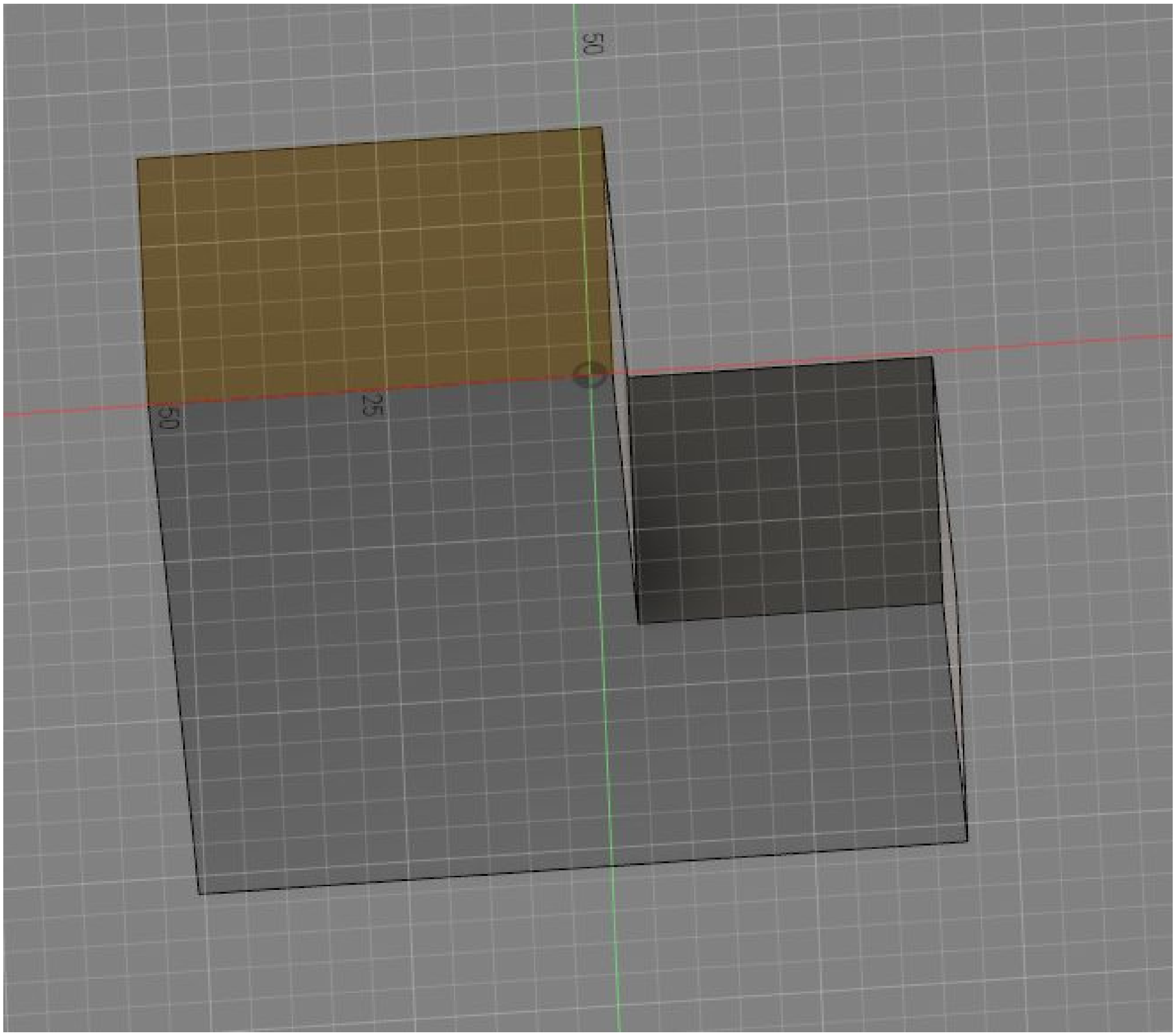
Even though the curves are not drawn, you can tell they exist because the profile exists.

Autoproject

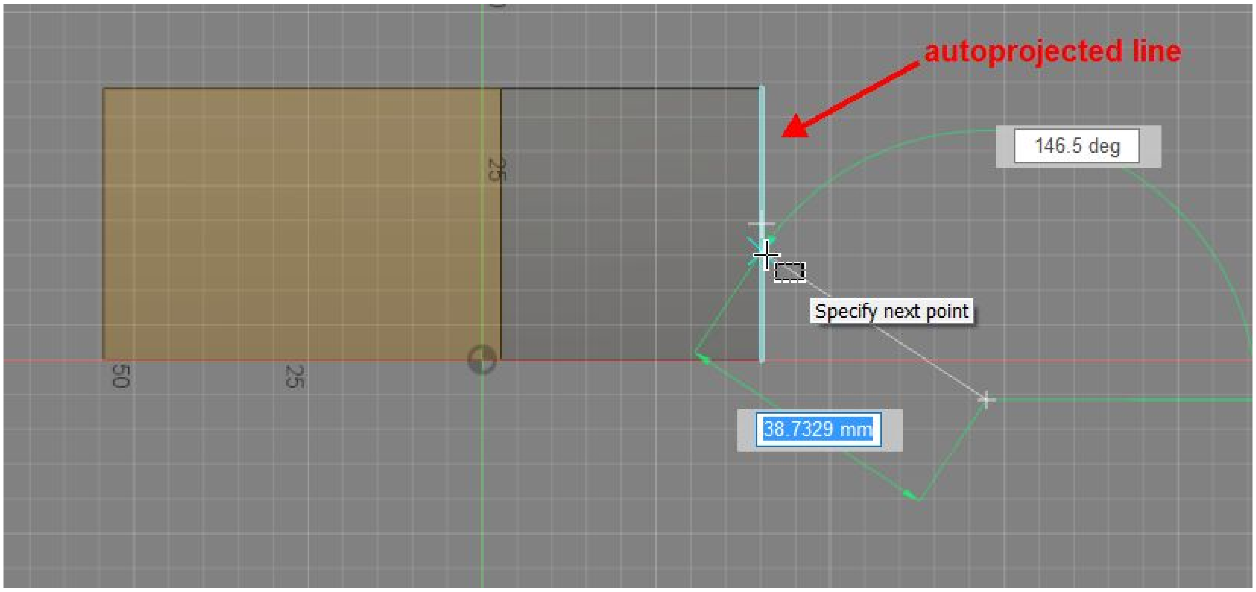
Fusion sketch also supports automatic projection, during sketch curve creation, if your view is looking at the sketch:

Screencast: **http://autode.sk/2er654d**

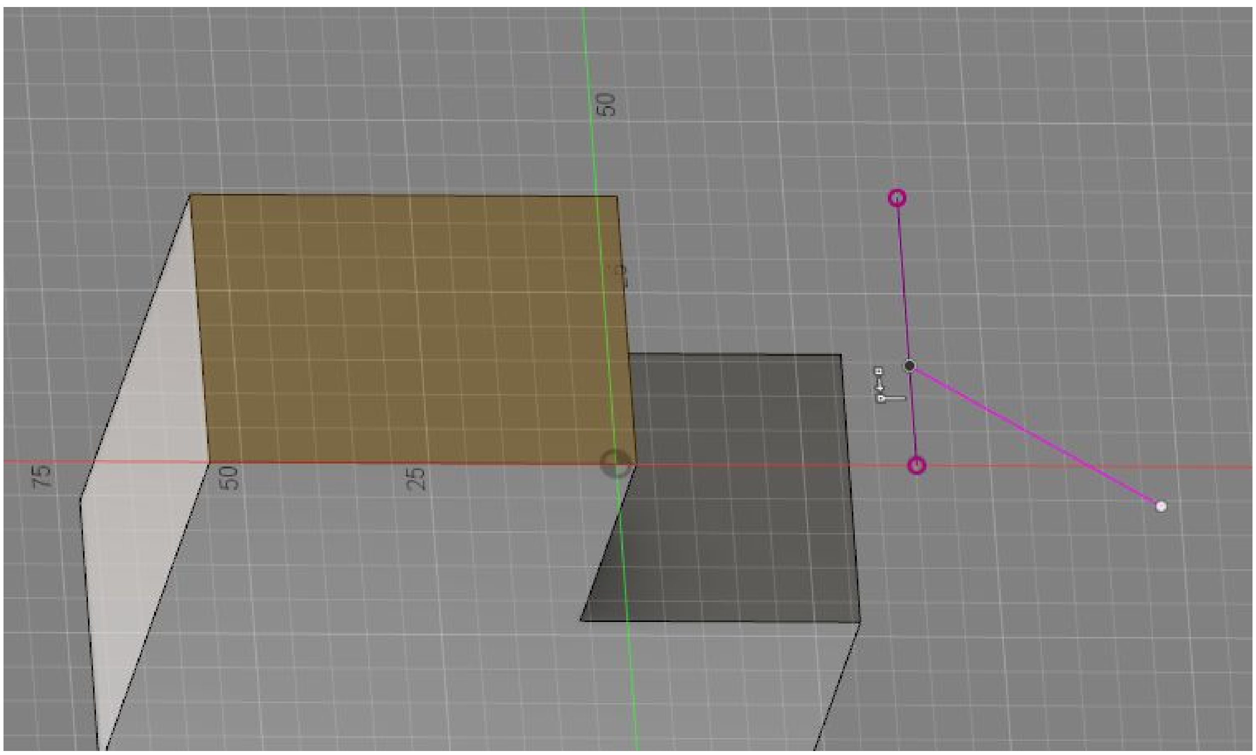
Sketch on a face:



Then, if you are looking at the sketch, and start the Line command, then you can infer a projection to an edge that is not already in the sketch:



Which you can see if you rotate the view:

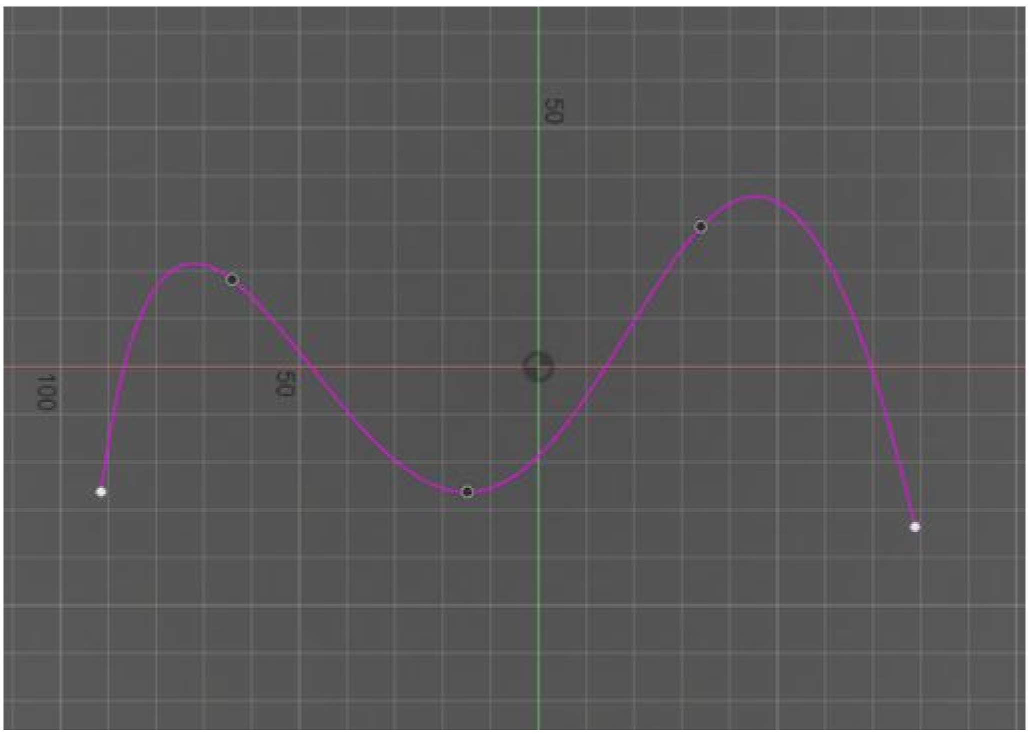


This, however, will create a reference to the edge, like other projections. However, because it happens automatically, it can be surprising to find later that the reference has failed, since it was created without an explicit Project command. So, personally, I never use this option. I'd rather project my own geometry explicitly.

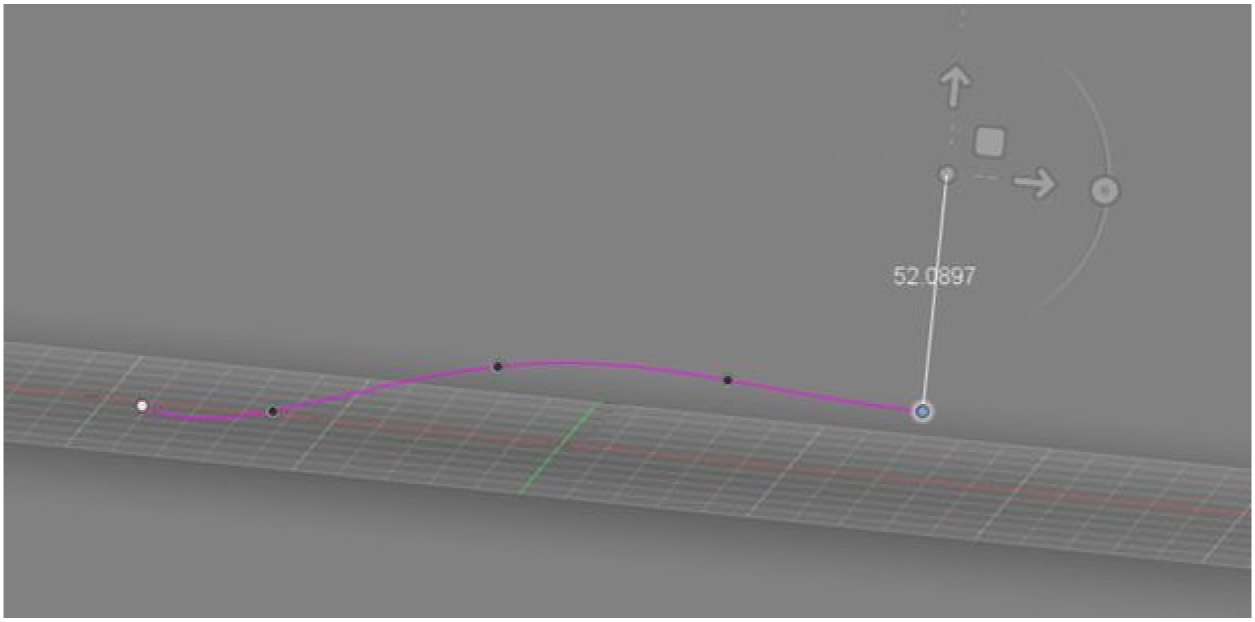
Tip 3: 2D and 3D sketches

This is one of the more powerful, and yet sometimes confusing aspects of Fusion sketch. Unlike Inventor, Solidworks, etc, Fusion has only one type of sketch. There is not a separate 2D and 3D sketch environment. Every sketch in Fusion can have 2D and 3D geometry. To be technically accurate, every sketch in Fusion is 3D. Some geometry is just constrained to a plane. That's why you can do this:

Start with a nice 2D spline curve:

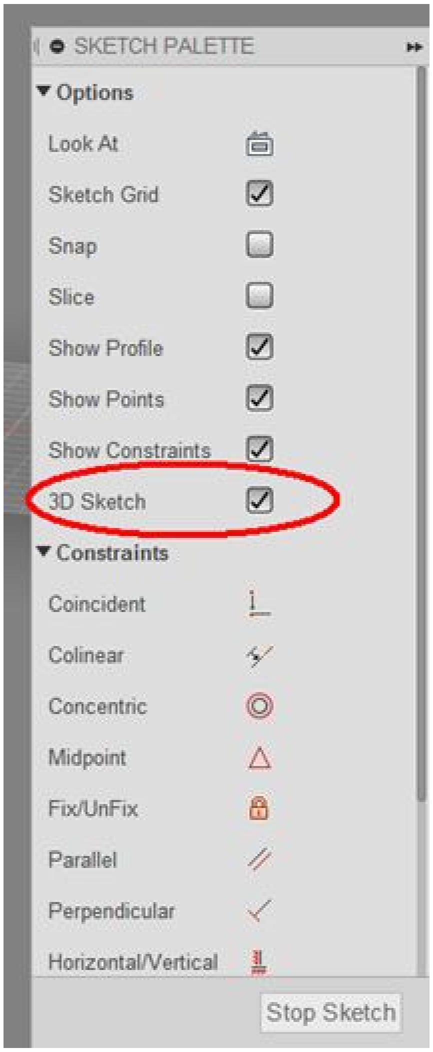


Rotate the view, fire up the Move command, select a fit point, and move it out of the sketch plane:

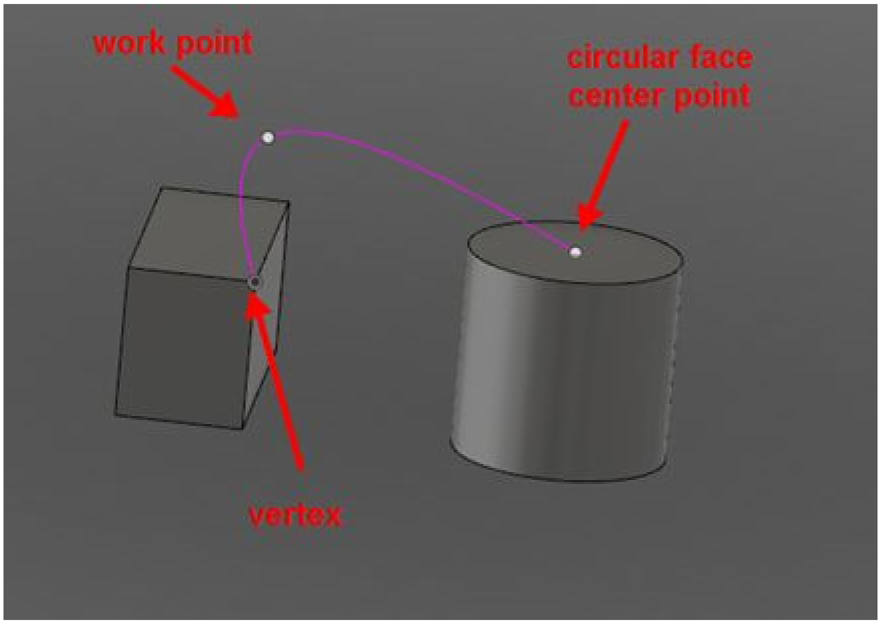


and voila, you have a 3D curve. No separate 3D sketch needed!

You can also (with some current limitations that we hope to remove soon), create 3D curves in a sketch, by just turning on “3D sketch” in the sketch palette:



With this on, you can snap to body vertices, work points, etc to create 3D lines and splines:



Because all Fusion sketches are inherently 2D/3D is why, when you start a sketch, you still have to select a sketch plane. If you do not have 3D snapping turned on, then when creating curve, all mouse movements snap to the sketch plane.

Some geometries and commands are inherently 2D only (offset, fillet, circle, dimensions, some constraints, etc)

For the time being, the types of constraints supported for 3D geometry is fairly limited (point to point coincident, tangent for splines), but we hope to increase this in the future.

Future vision: With this basic capability today, you can still do quite a bit. However, in the future, we hope to make this mixed 2D/3D environment much richer. On our list is:

- 3D dimensions
- More constraints (point on surface, parallel to plane, etc)
- More curve types (allow 3D arcs, etc)
- Multiple “islands of 2D” within a single sketch (to help with wire/pipe routing, etc)

Tip 4: Working with Fusion Splines

Splines in Fusion are also simultaneously loved and hated: You can do a lot with them, but they are currently a bit fiddly to get right. Some of that we are fixing in a release soon.

Don’t use too many fit points:

The whole idea of a spline curve is that it produces a nice smooth curve that goes through the points you specify. If you put too many points, a couple of things will happen:

1. The curve will not be very smooth
2. The curve will be very hard to edit
3. The resulting surface from that curve (from Loft, Sweep, even Extrude) will probably not be a very “nice” surface – it will have ripples, etc

It’s very tempting, especially when tracing a canvas, to put lots of points in it to be “more accurate”. For example:

But, that will not serve you well. Instead, use a lot fewer points, and use the tangent handles to get the shape you need. It's a bit of an art, but it's not that hard. You will need more points for areas of high curvature, but in general, use as few as you can get away with. Something like:

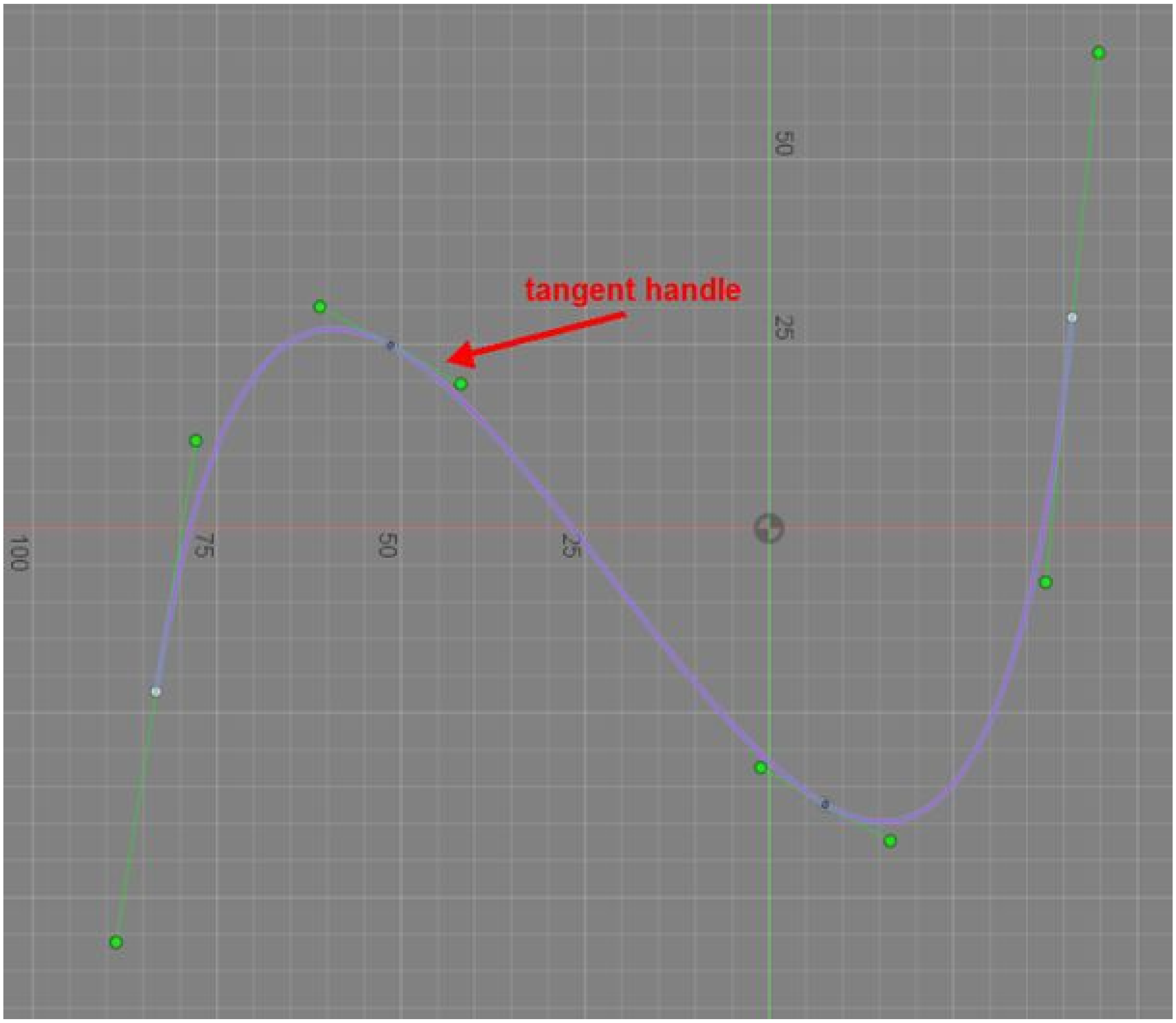
I did this in about 5 minutes, and I didn't even have to use tangent handles to do it.

About those tangent and curvature handles:

Speaking of tangent handles, this is one area of constant frustration for a lot of Fusion folks. Let me explain a bit about how they work, and once you understand that, it will help with using them to get the curve you want.

These handles allow you to control the shape of the curve by controlling the direction, influence, and curvature of the curve at a fit point.

If all you do is create a spline by plunking down points, you will get a nice, smooth curve through those points. But, for more advanced control, using the handles will get you more control. Here is what is displayed for a just-created Fusion spline that is selected:



By default, the tangent and curvature handles are “inactive” (meaning the spline will just follow its natural curvature at each fit point). The handles in the above image are inactive. In order to edit the handles, you have to “activate” them. There are a couple of ways to activate a tangent handle:

1. Explicitly. If you select the spline, all the inactive handles will be visible and green. When visible, they can be selected. If you select a tangent handle and

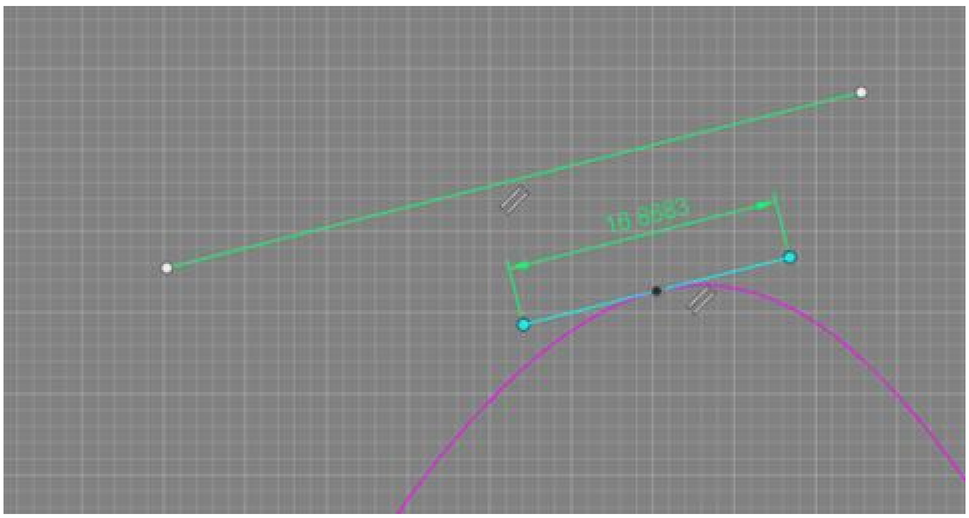
right click and choose “Activate Tangent Handle”, that will activate that handle in its current state.

2. Implicitly. When the Tangent Handles are visible, if you drag on the handle, it will be implicitly activated.

Here is a screencast showing how to explicitly and implicitly activate a tangent handle: http://autode.sk/2dVlI3u

When you drag a tangent handle, you are simultaneously controlling the direction (by changing the line direction) of the curve at that point, or the “influence” of the tangent direction at that point (a longer line will have more influence, a shorter one will have less)

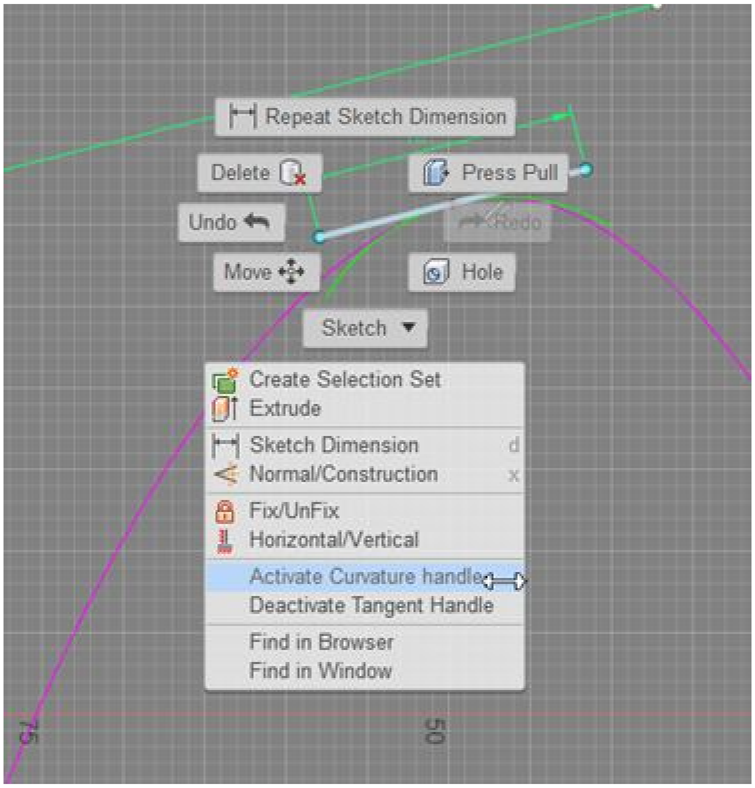
You can use constraints and dimensions to more precisely control the Tangent Handles:



Here I have made the handle parallel to a fixed line, and controlled its influence with a dimension.

The visibility of the handles, today, does not work well. Unactivated handles show up when the spline is selected, but, for some reason the activated handles do not. For those, you have to select the fit point. I think this is a bug, and we have a project planned to improve the usability of handles.

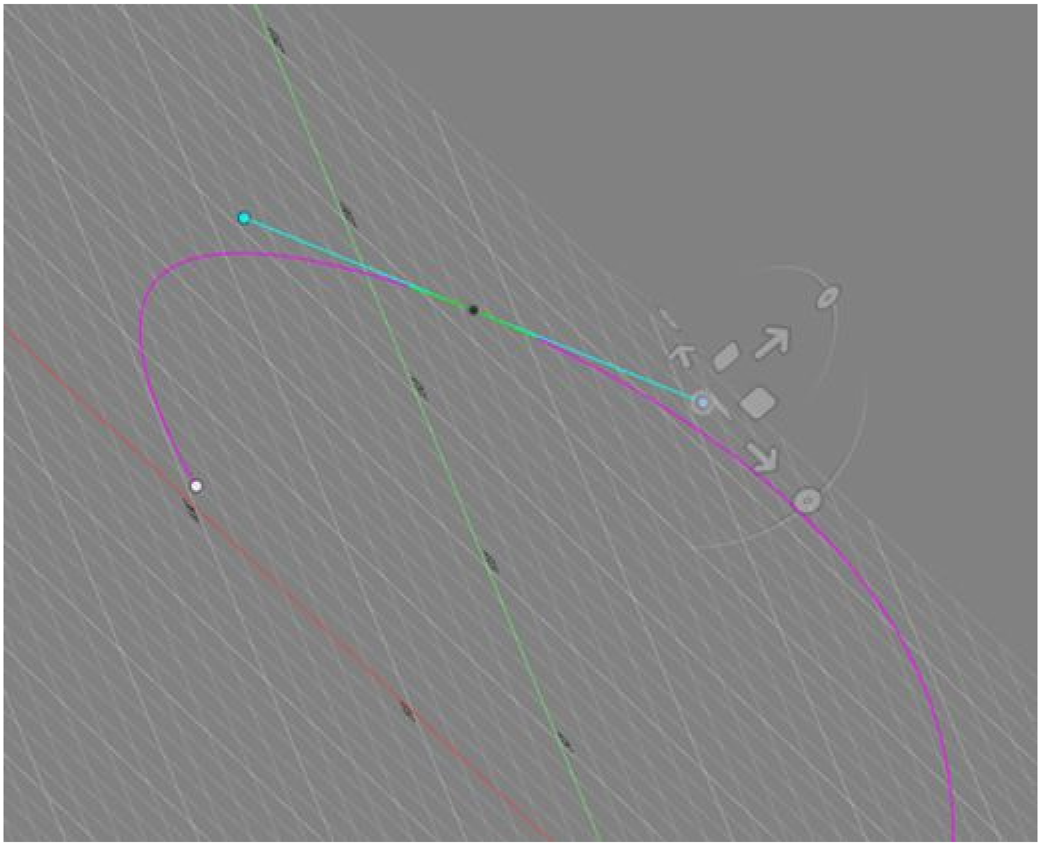
Curvature handles. These are slightly more advanced controls. They, also, need to be activated to be useful. However, for some reason lost to history, they cannot be implicitly activated. Also, for again an unknown reason, you activate Curvature Handles by selecting an active Tangent Handle. Go figure.



Here is another screencast showing this: http://autode.sk/2dJK6qz

Once activated, they can be dragged, dimensioned, etc:

Handles for 3D splines. Yes, 3D splines also have handles. They must be activated the same as 2D splines. The only difference is that, just like the splines themselves, you cannot just drag the handles. You have to use the Move command:



And here is a screen cast: http://autode.sk/2dBXDo8

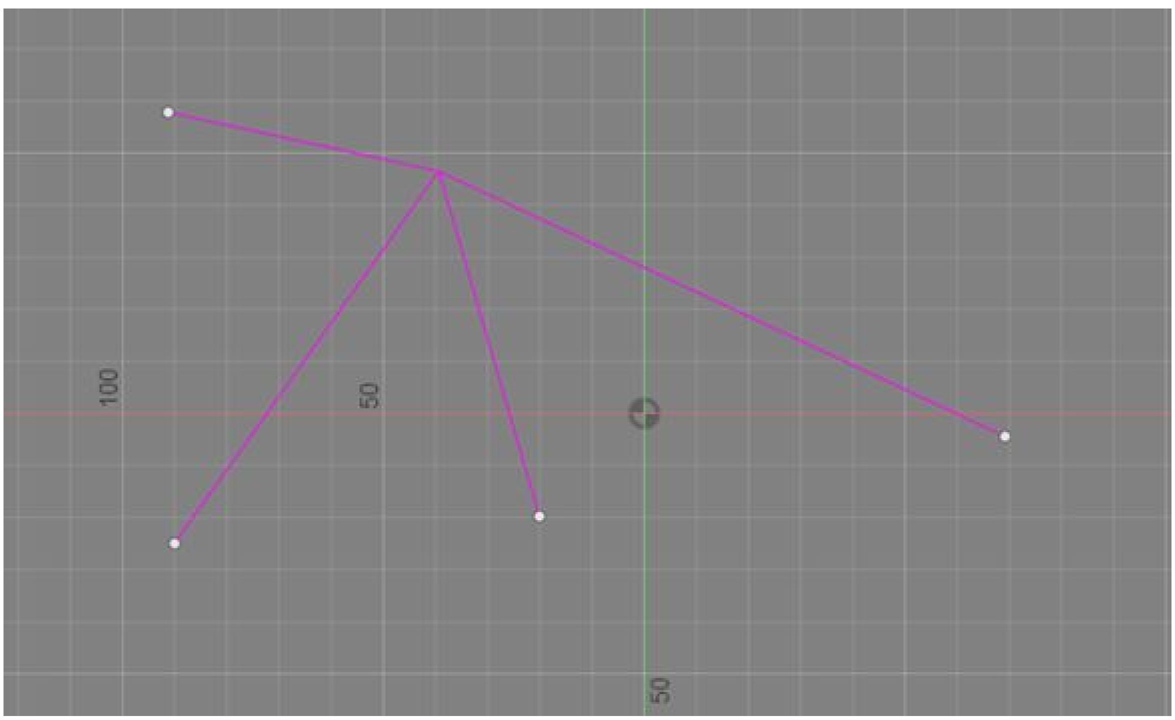
The main difference for 3D splines is that if the curvature handle is activated, it seems you cannot adjust even the tangent handles.

Tip 5: Understanding Point Merging and Coincident Constraints

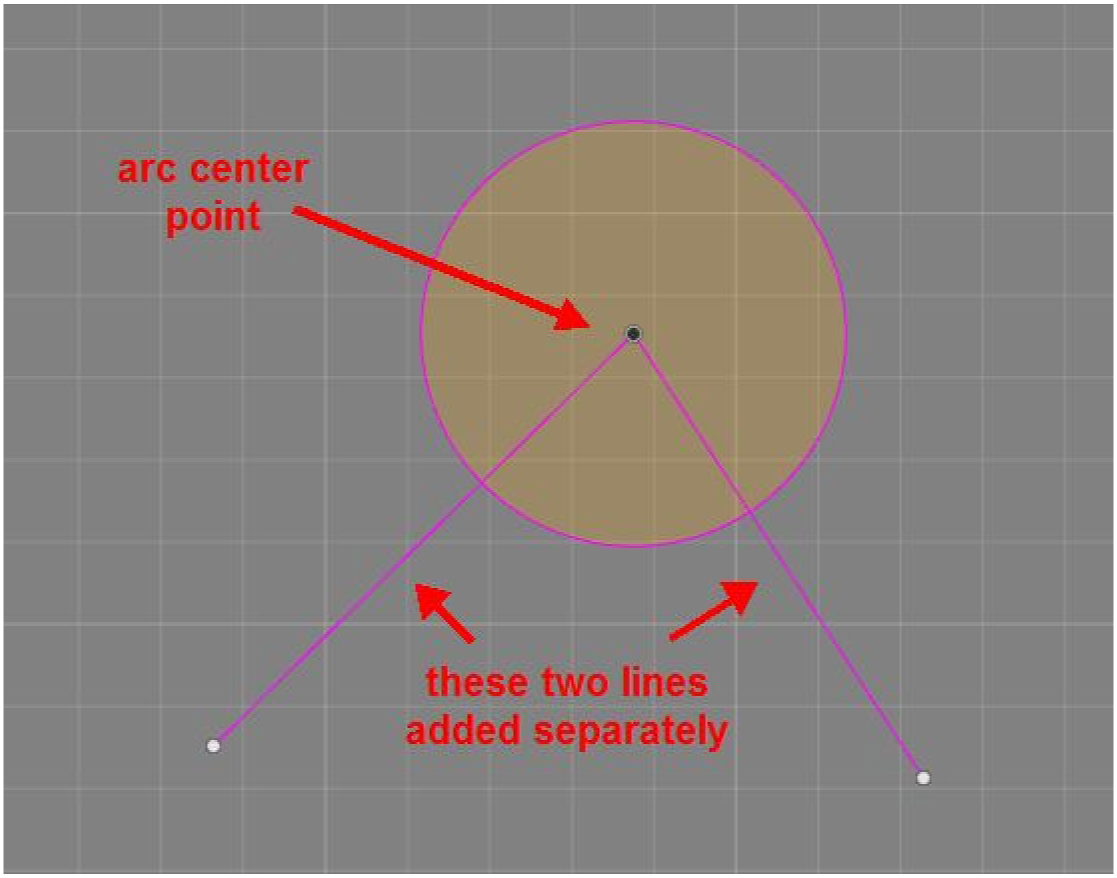
This section covers how Fusion manages sketch points. This can be an area of confusion, especially for new users.

In most cases, Fusion will “share” sketch points between curves that attach to them. For instance, when drawing lines, such as this:

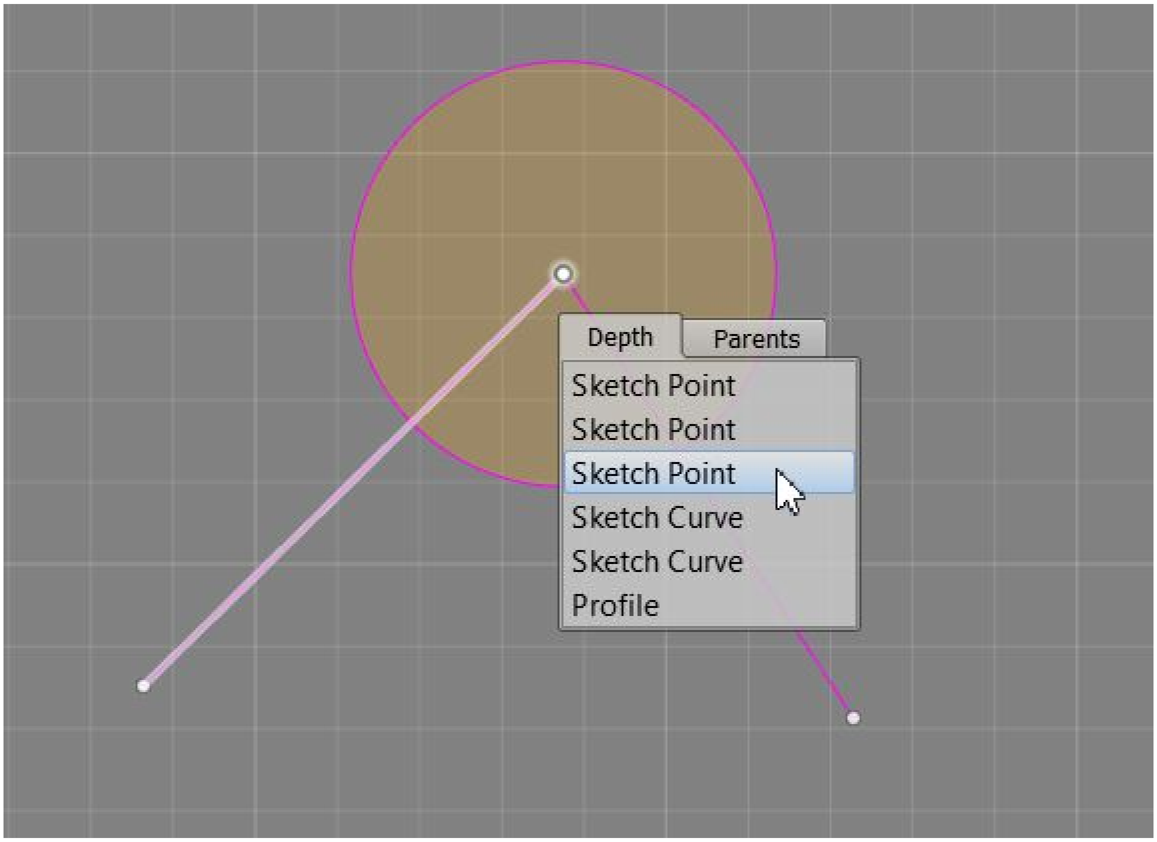
This is true, even if there are more than two curves:



The main exceptions to this are: Arc center points, and points placed using the Point command:

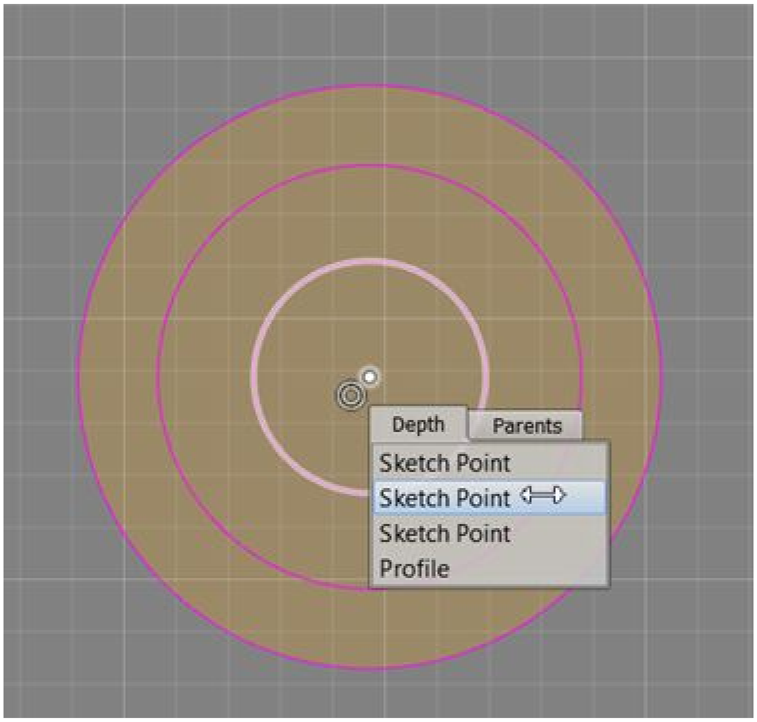


You can tell that this is not a merged point because it is displayed in black, You can also tell using Select Other:

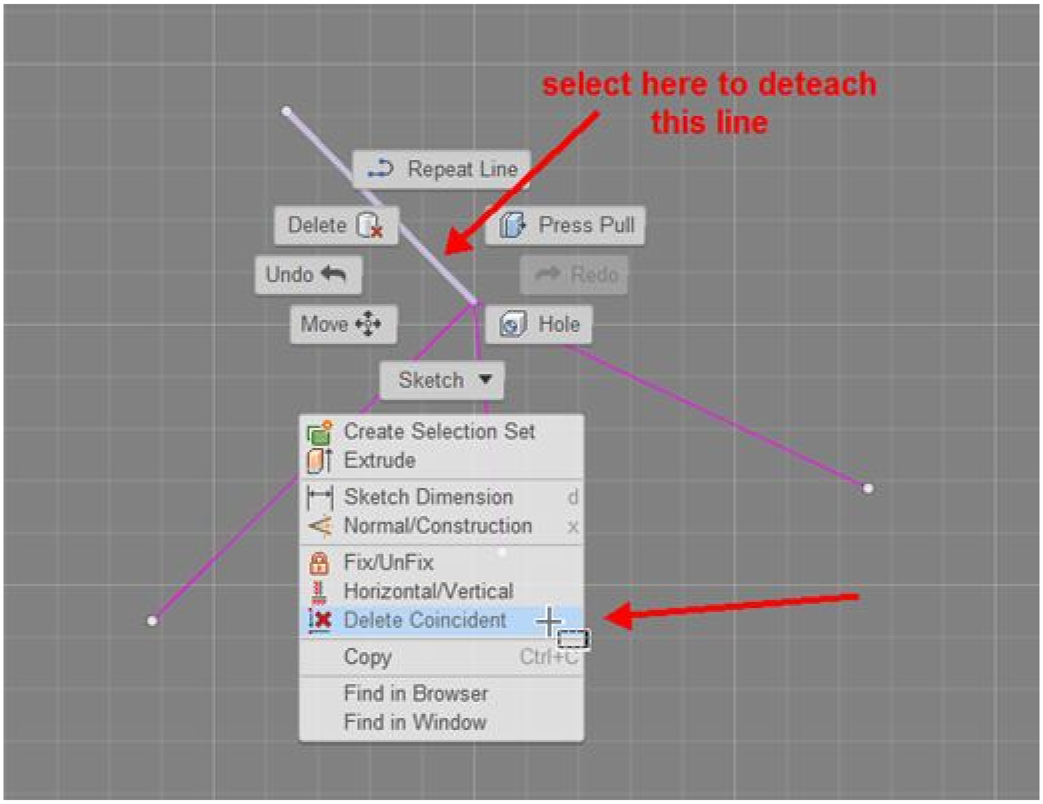


Recently added is cross-highlighting of the points and the curve to which they are connected, shown in the above image.

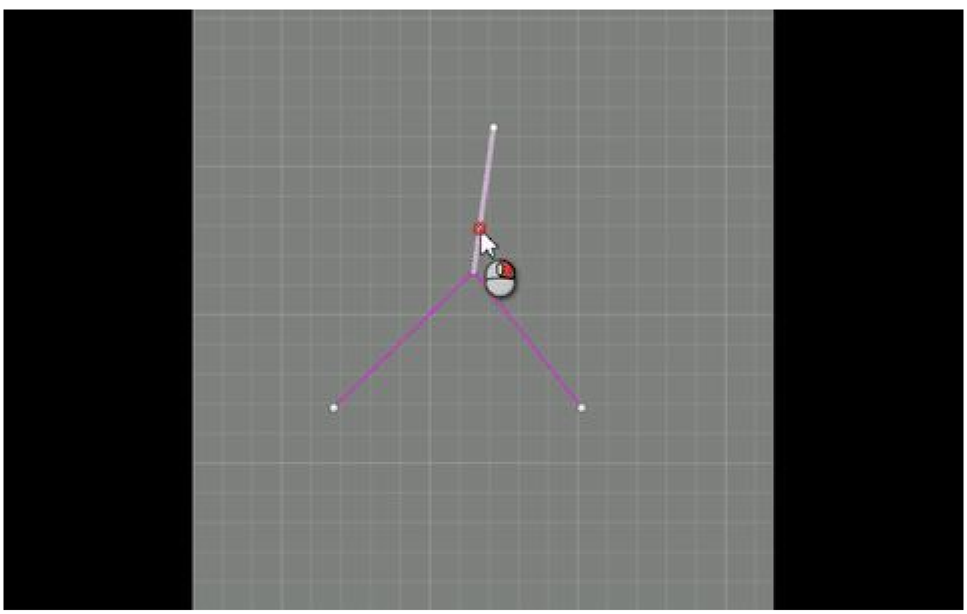
Point merging is also not done for arc/circle/ellipse centers:



How to break apart merged points. What if you have a single merged points, but you want to break it apart? Use the Delete Coincident command, selecting the curve you want to detach, near the shared point:



And here is a screencast: http://autode.sk/2dThl9Z

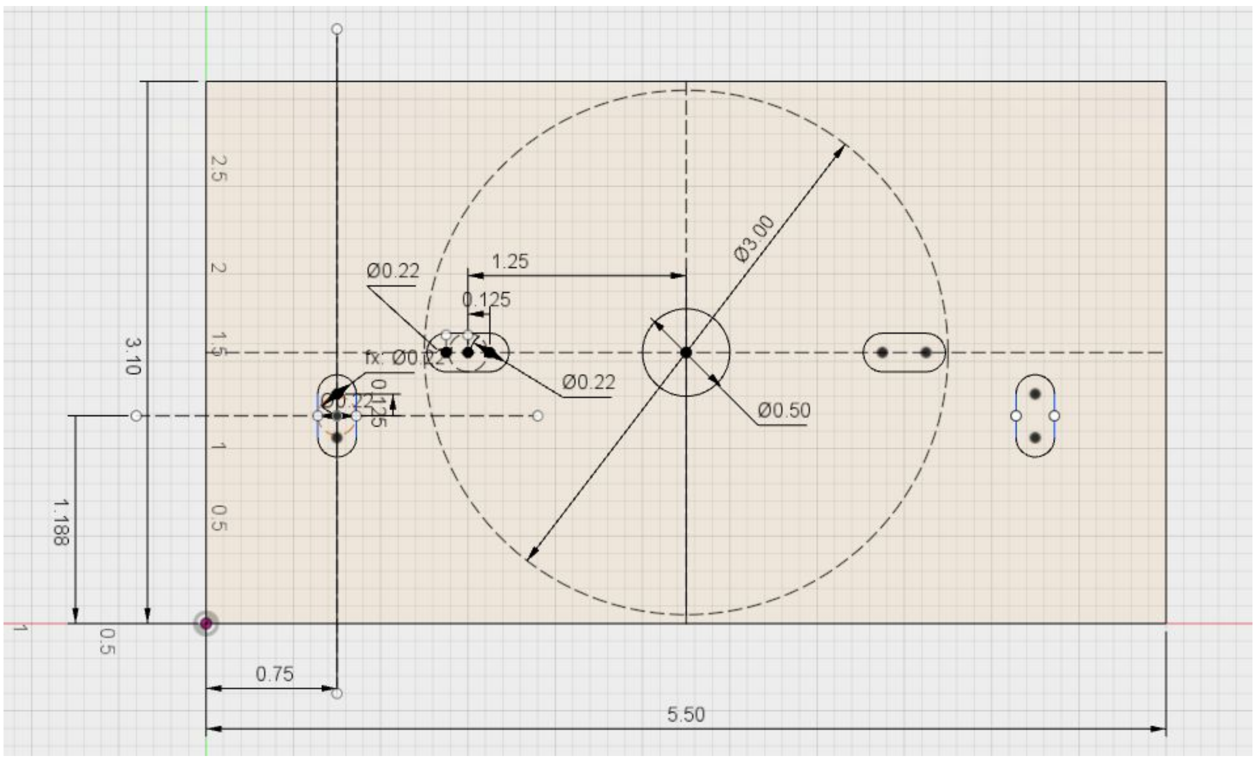


Tip 6: Sketch Profile Tips

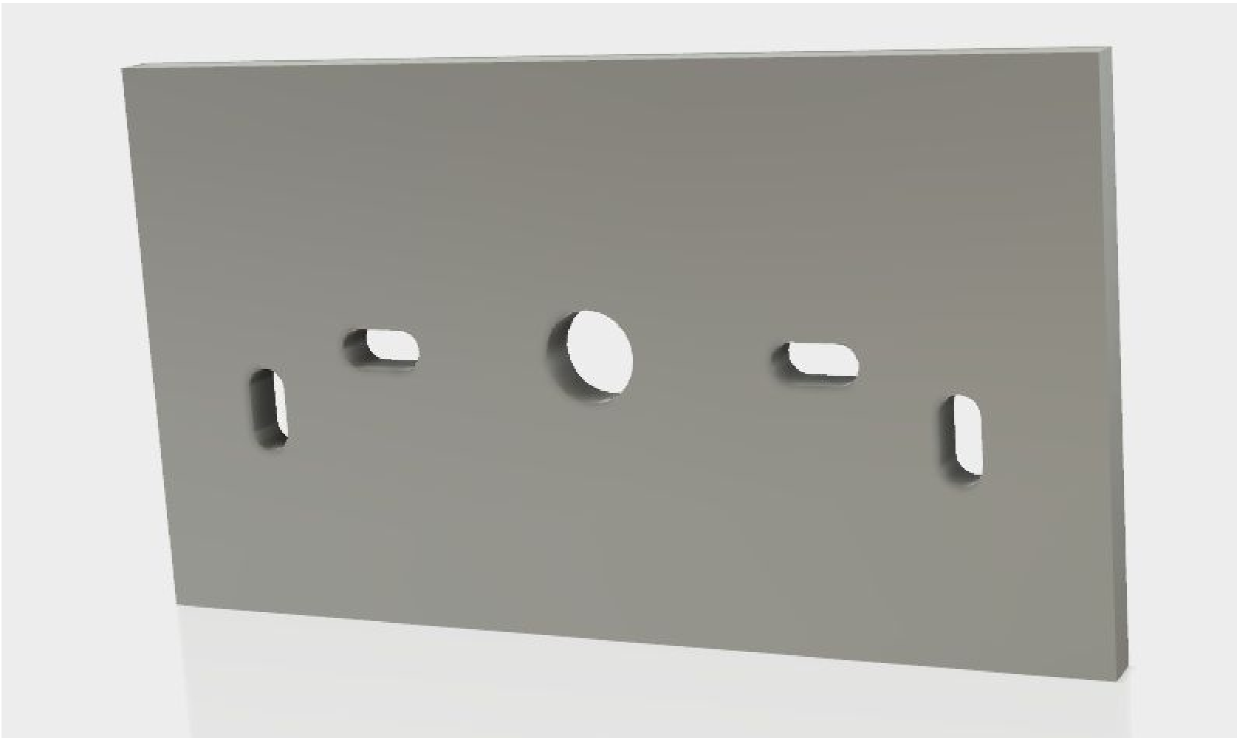
This section covers some useful suggestions for working with sketch profiles. For creating solid models, profiles are the ultimate goal of a sketch, so understanding how to best work with profiles is critical.

Use construction geometry to simplify your profiles:

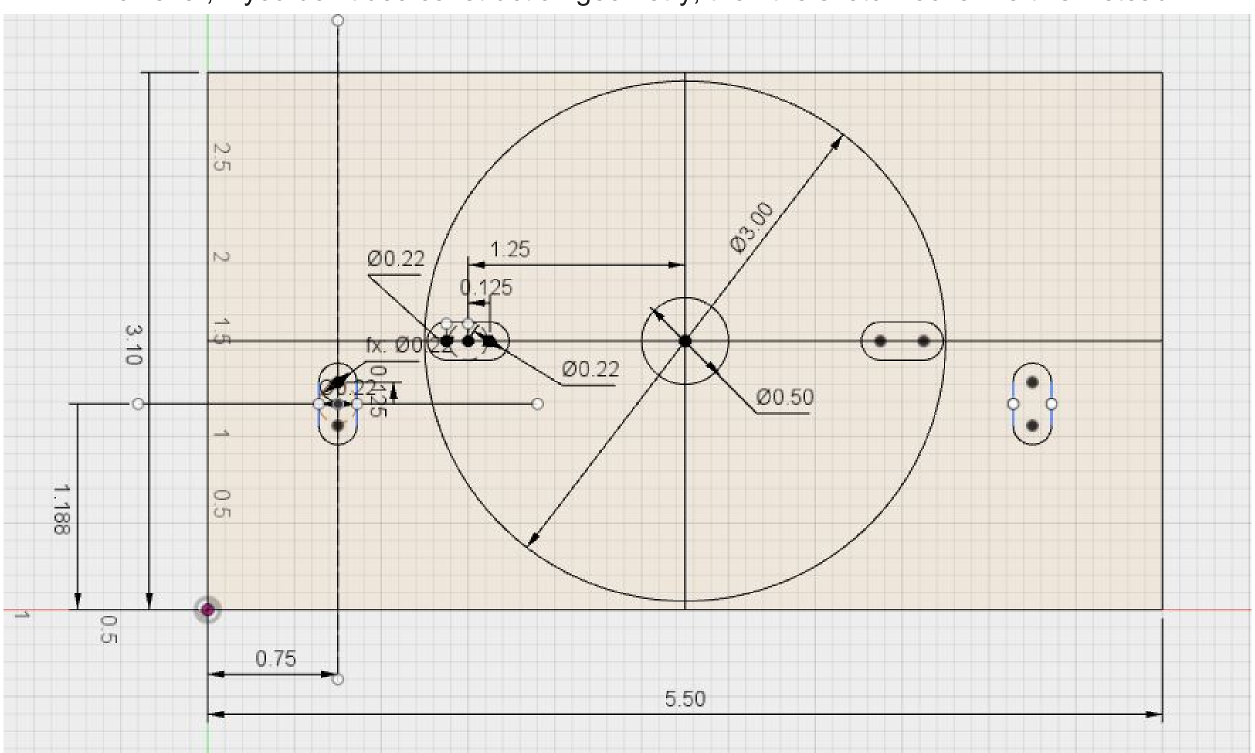
The process of converting sketch geometry regions into profiles can be expensive. To help with this, use construction geometry:



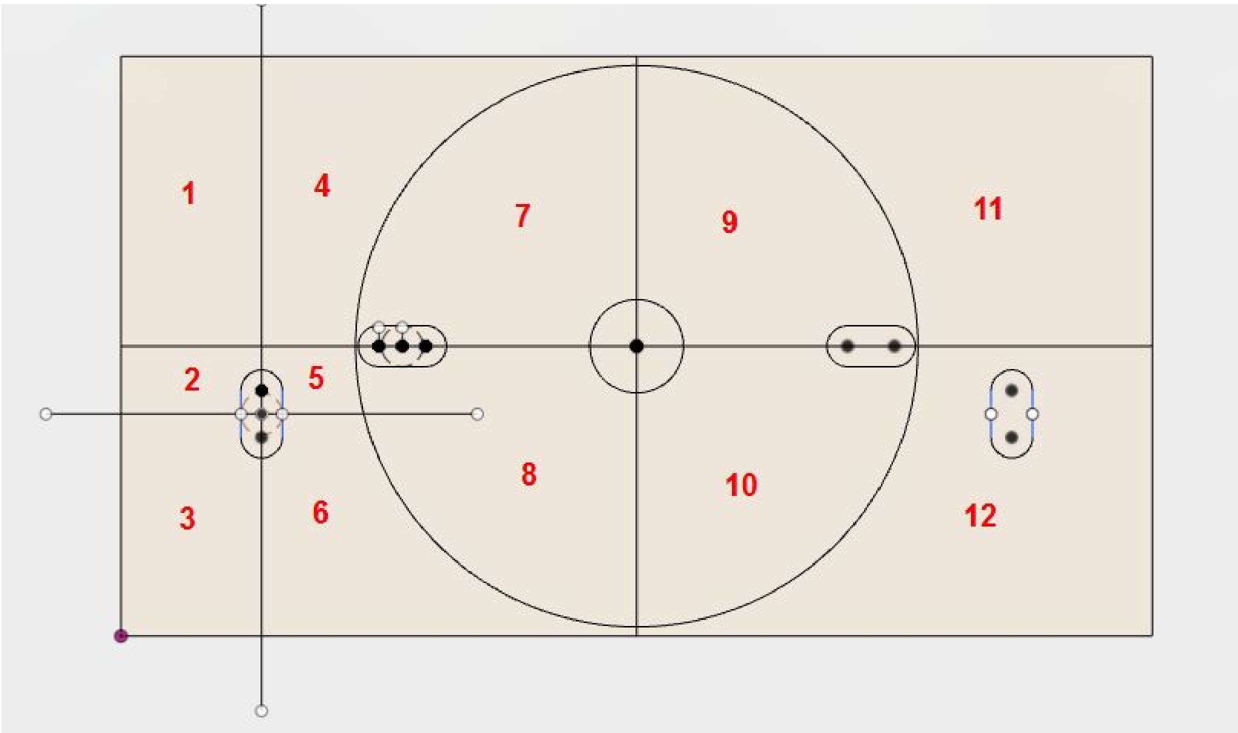
In this case, to get the result below, you just have to select a single profile in the above sketch:



However, if you don't use construction geometry, then the sketch looks like this instead:



And, to extrude the result, requires you to select 12 different profiles:

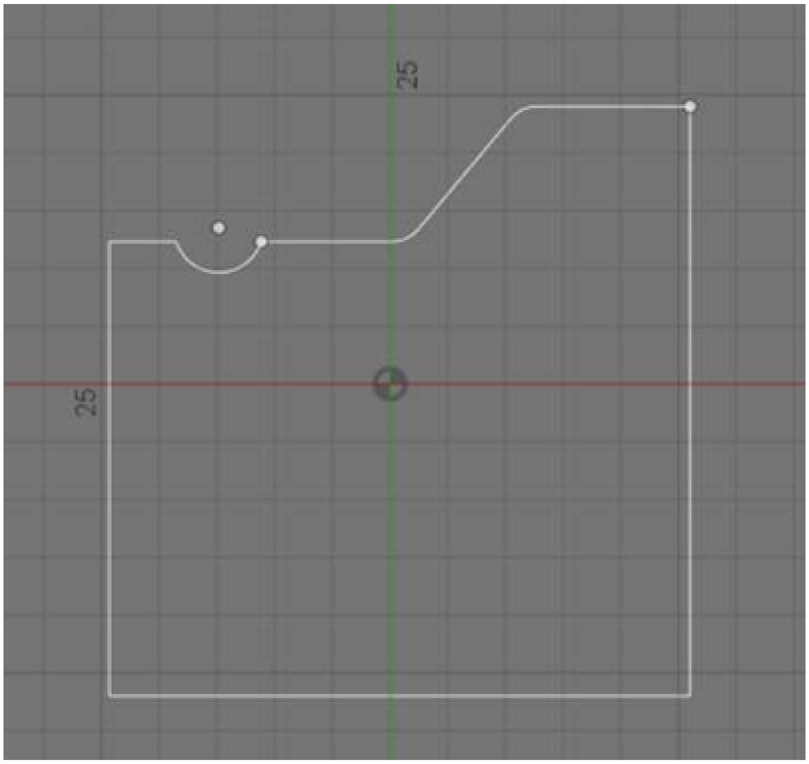


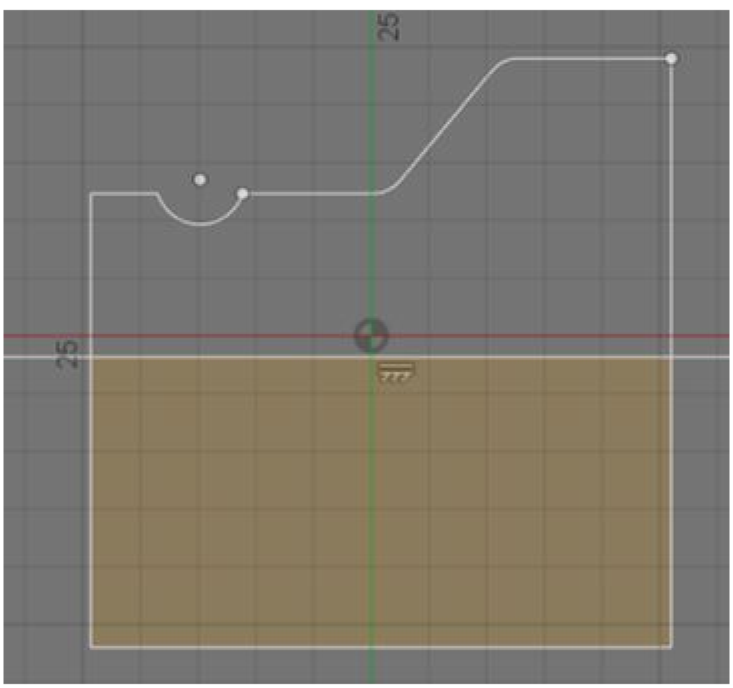
Also, the use of construction geometry will help make your features easier to create, and more stable when your sketch changes.

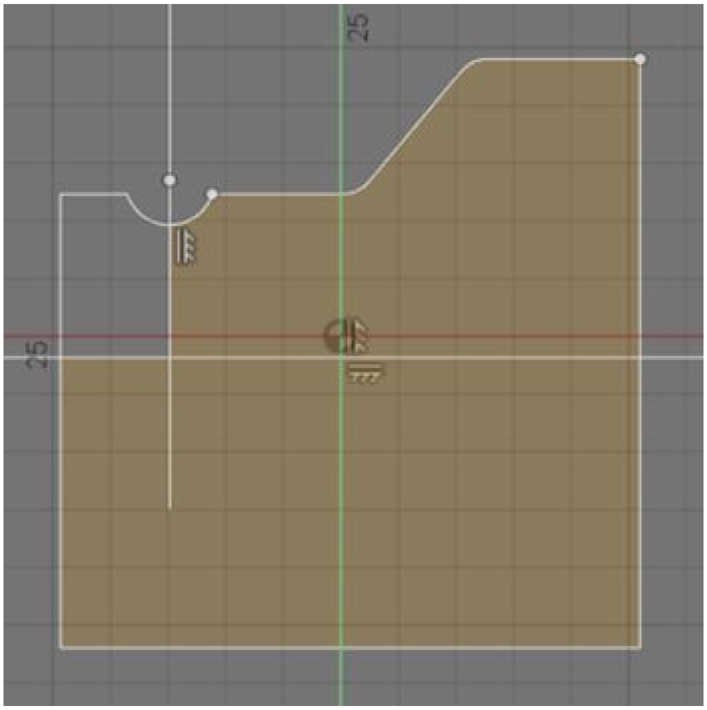
How to find a “leak” in a profile:

Fusion, unlike some some CAD systems, allow profiles to be recognized even when the curves are not explicitly connected. However, sometimes this leads to situations when profiles are not recognized. This can be caused by a “leak” in the profile.

See this excellent post by @HughesTooling: find-break-in-sketch-geometry. This is one of my favorite techniques for debugging a sketch. The basic idea is to add geometry to your sketch and see where the profiles are identified:


25
25


25
25

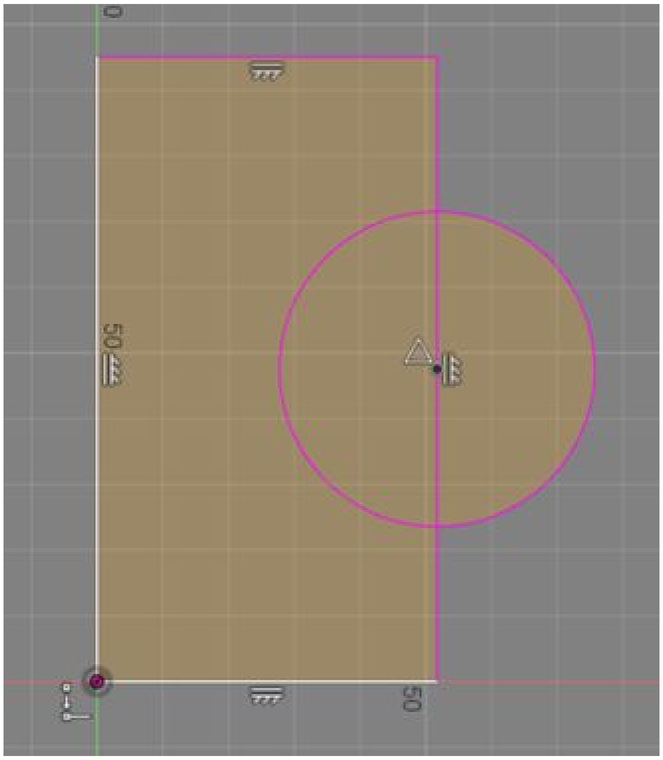


Tip 7: Miscellaneous Sketch Tips

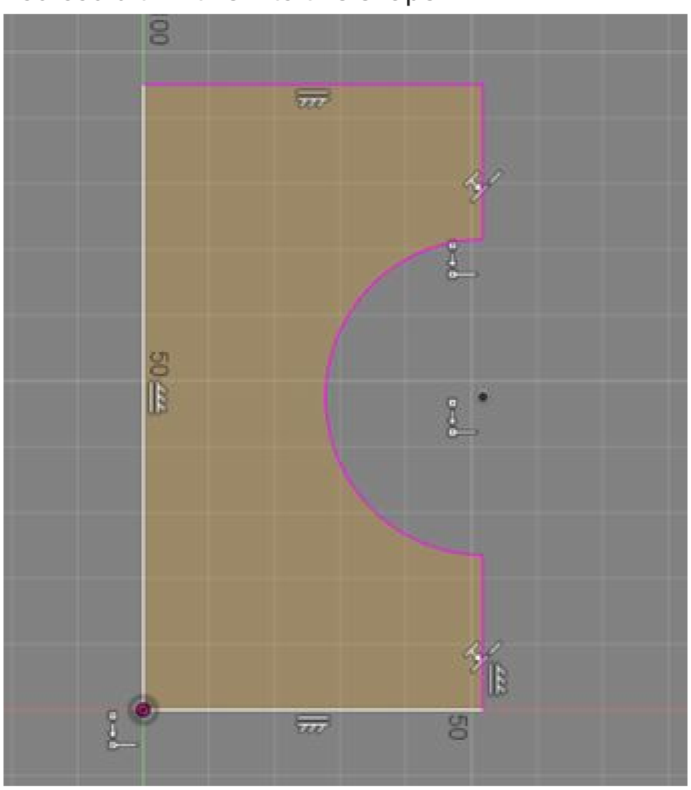
This section covers some useful suggestions for working with sketch profiles. For creating solid models, profiles are the ultimate goal of a sketch, so understanding how to best work with profiles is critical.

You should almost never need to use sketch trim:

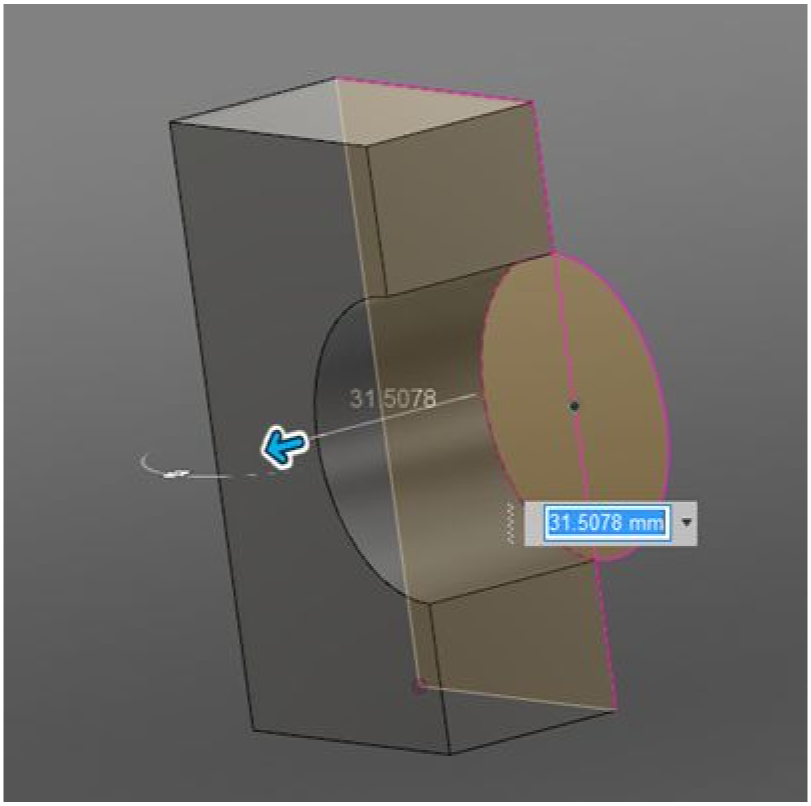
Because Fusion can extrude sub-regions of a design, you should never need to trim your sketch. For instance, if you have this sketch:



You could trim this into this shape:



However, you can very easily Extrude this shape, without doing any trim operations:



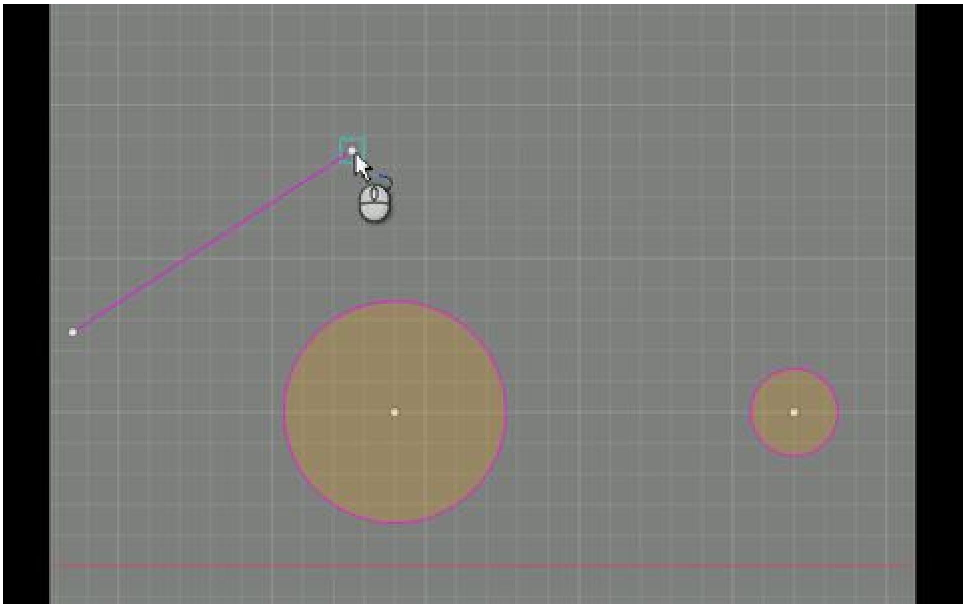
Drag to create tangent lines from circles, or tangent arcs from line endpoints:

This one is really just a little productivity tool that saves a few clicks. If you come from Inventor, you will already know this. But, the line command can do more than just create connected line segments. It can also create arcs tangent to lines and lines that are tangent to arcs.

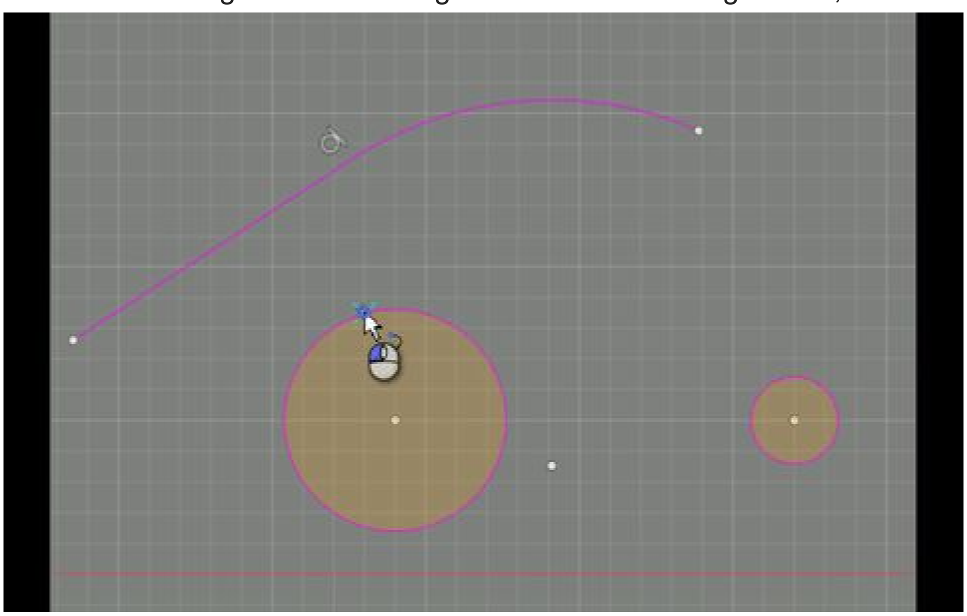
If you click and drag on a line endpoint, you can create a tangent arc at the end of a line, either an existing line, or one that you just created in the line command:

Screencast: http://autode.sk/2elupbz

Drag from a line to create an arc:



You can also drag from an existing circle to create a tangent line, at both endpoints:

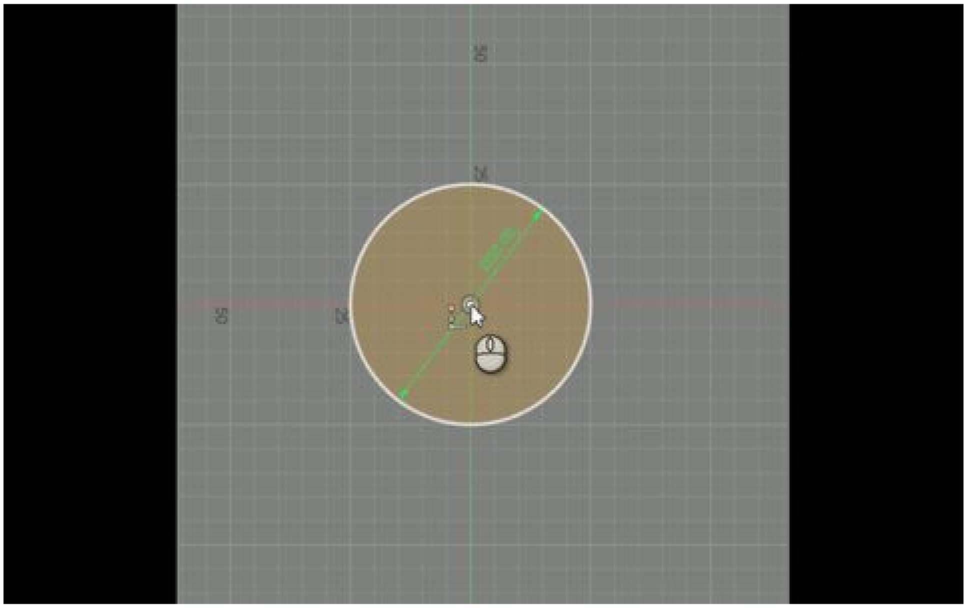


Beware of the fixed point at the sketch origin

This feature was added at customers' request, when Fusion added support for geometry coloring based on whether it is fully constrained or not. In order for a sketch to be fully constrained, at least one point must be fixed. So, Fusion introduced a fixed point at the sketch origin. But, this is something you need to be aware of. Many times, new users will be confused when, for instance, they create a center point circle at the sketch origin, and find they cannot move this circle afterward.

To fix this, you will have to break this constraint using the Delete Coincident command:

Screencast: http://autode.sk/2dSN9MZ



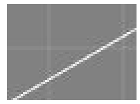
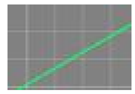
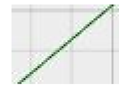
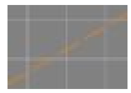
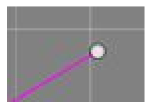
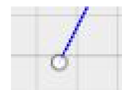
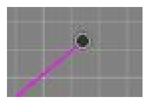
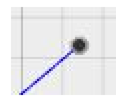
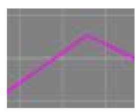
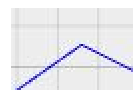
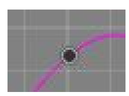
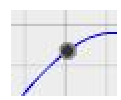
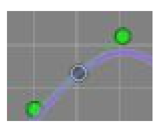
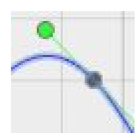
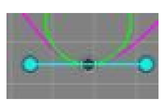
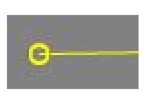
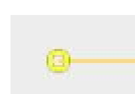
Future direction: In upcoming releases, we hope to add the ability to also drag perpendicular arcs and lines:

- Drag a perpendicular arc from a line endpoint
- Drag a perpendicular line from an arc

What do all the colors mean?

This tip is to explain what the different colors in sketch are trying to tell you. The problem with this topic is: the colors are different for different environments. I will try to show a couple of the common environments.

Condition	**Color in Grey Room**	**Color in Photo Booth**
Underconstrained curve		

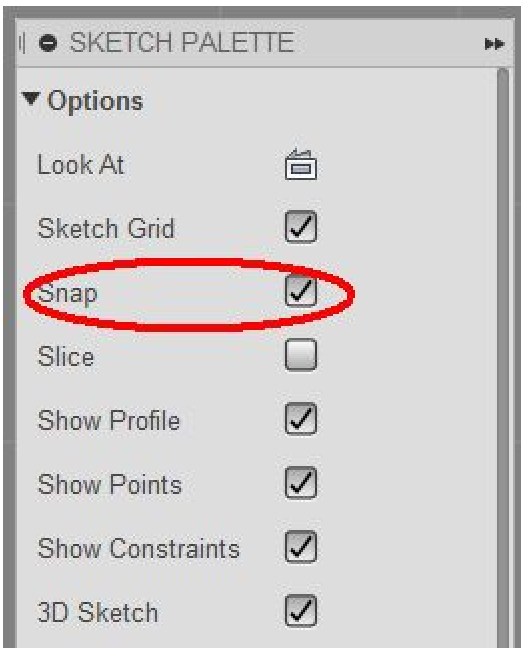
Fully constrained curve		
Fixed curve		
Construction curve		
Free point (connected to 0 or 1 curves)		
Stacked point (2 or more)		
Shared point (not drawn)		
Spline fit point		
Unactivated spline handle		
Activated spline handle		
Projected geometry		
Failed projected geometry		

Sketch snapping and constraint inferencing explained

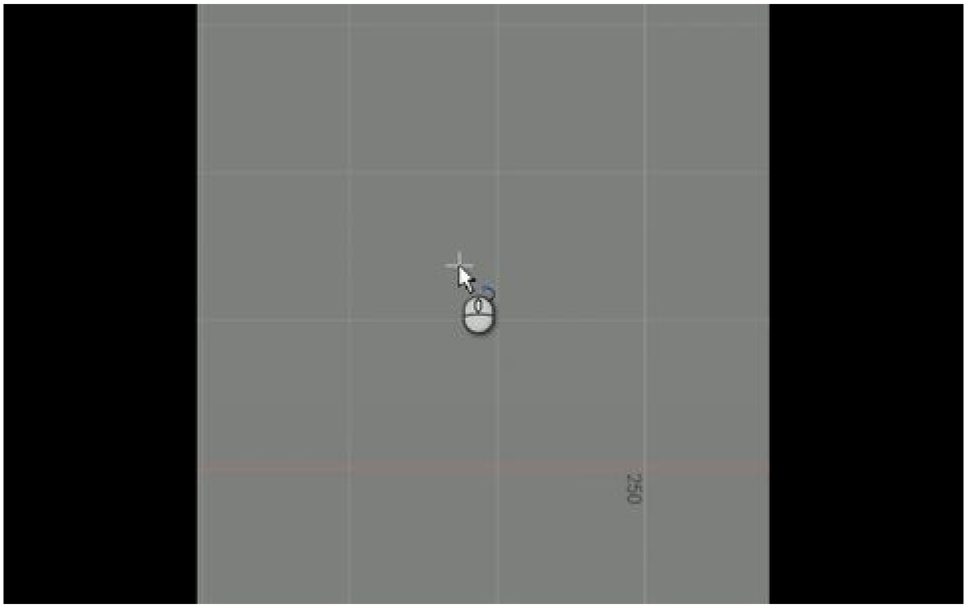
This section talks about using sketch snapping and constraint inferences to help smooth your workflow. There are several different concepts here:

1. Sketch grid snapping
2. Sketch alignment snapping
3. Sketch dimension creation
4. Sketch constraint inferencing

Sketch grid snapping. This one is pretty straightforward. With sketch grid snapping turned on (the easiest way is via the sketch palette):

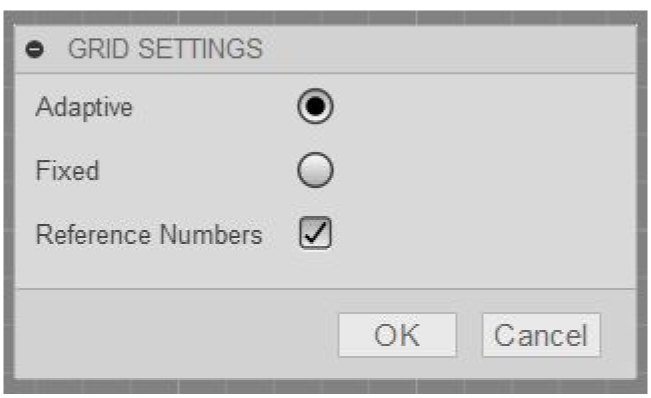


When you are in any creation command (line, arc, point, etc), the cursor will snap to a grid point. The video below shows a large grid to illustrate the behavior. You will snap to a grid point only when the cursor gets near to intersection, but is free to move elsewhere in the space in between grid points:

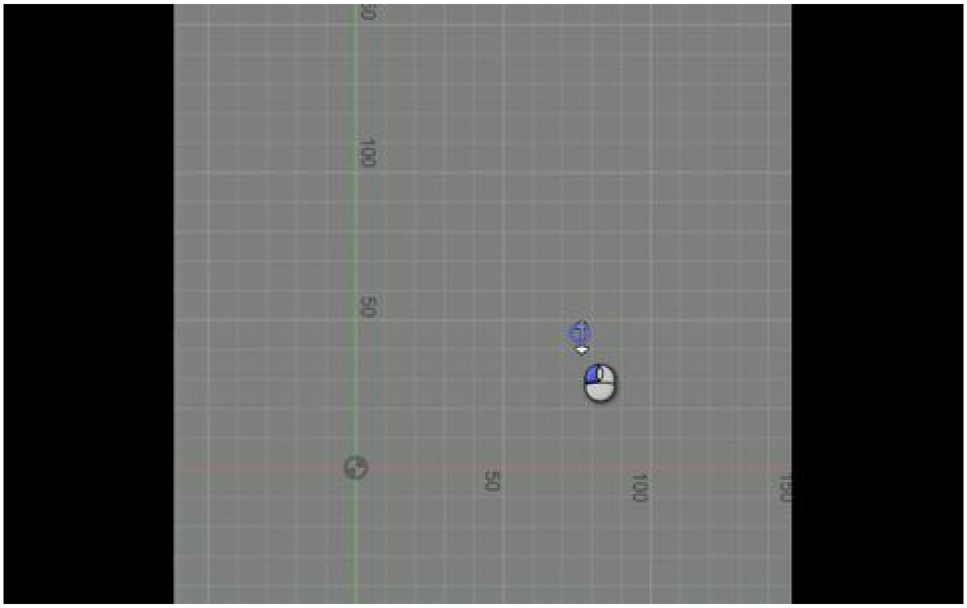


Grid settings: Use the Grid Settings command to change the grid:

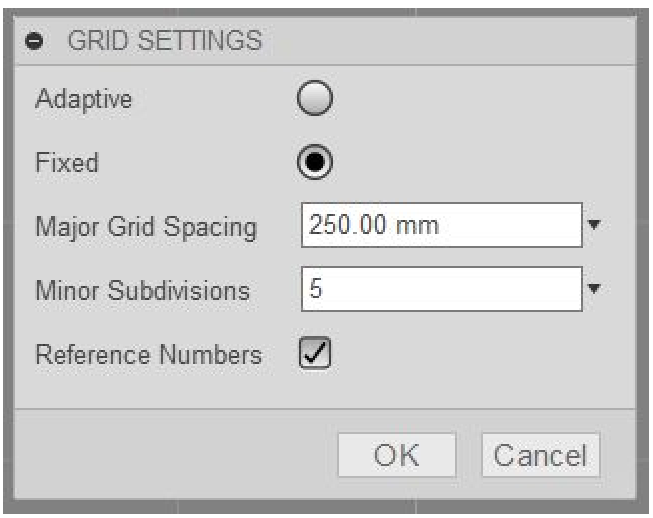
- How it reacts to zoom operations
- Grid distances
- Grid display



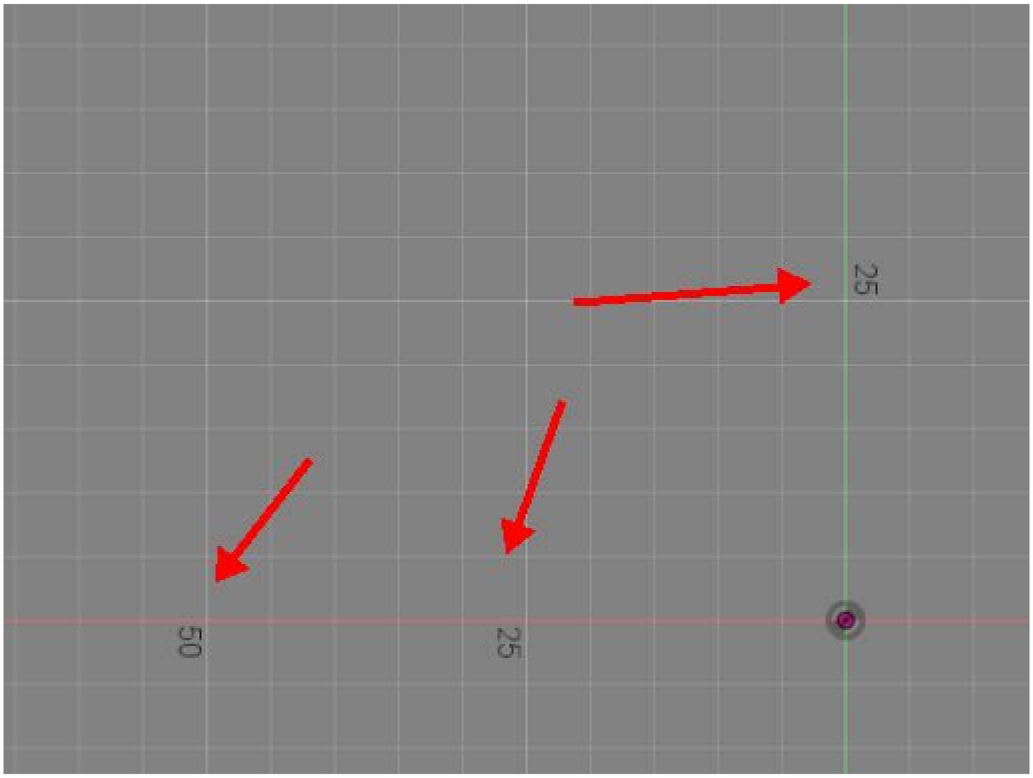
The first setting (Adaptive) is the default. When the grid is set to Adaptive, the grid will adjust to your zoom level:



In Fixed mode, you can set the grid spacing manually, and it will persist regardless of zoom factor:

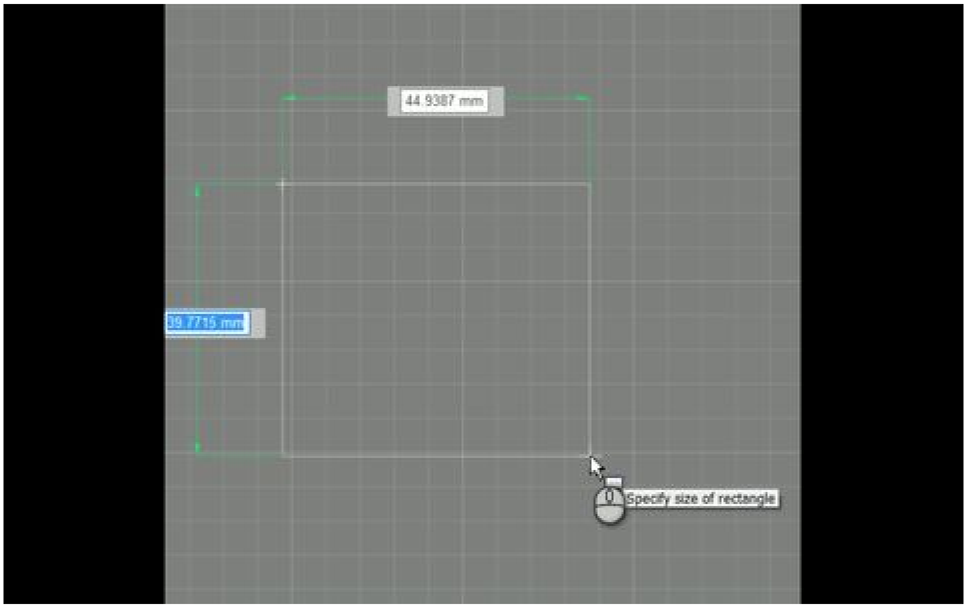


The “Reference Numbers” setting simply controls whether you see the grid numbers on the grid:



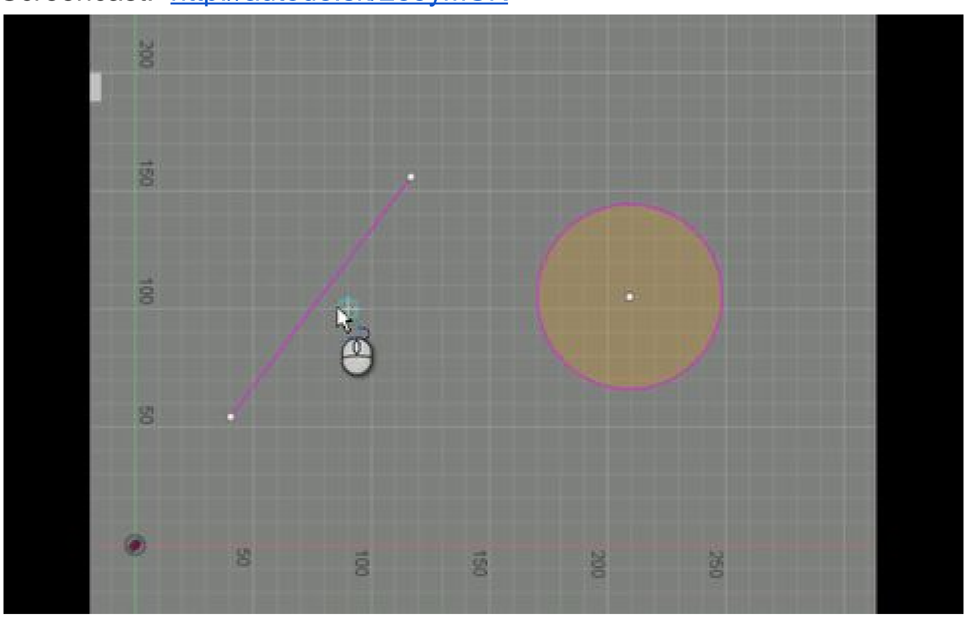
Dimension creation during curve creation. This item refers to the ability to create dimensions for some sketch commands while creating sketch geometry. For Line and Rectangle, dimensions are previewed. If you type in a value, that value can be locked using the Tab key. If you do not lock the value, it will be changed if you drag the mouse to a different location. If the value is locked, when the command is finished, a dimension will be created for that value. Unlocked values are left undimensioned:

Screencast: http://autode.sk/2e5vocA

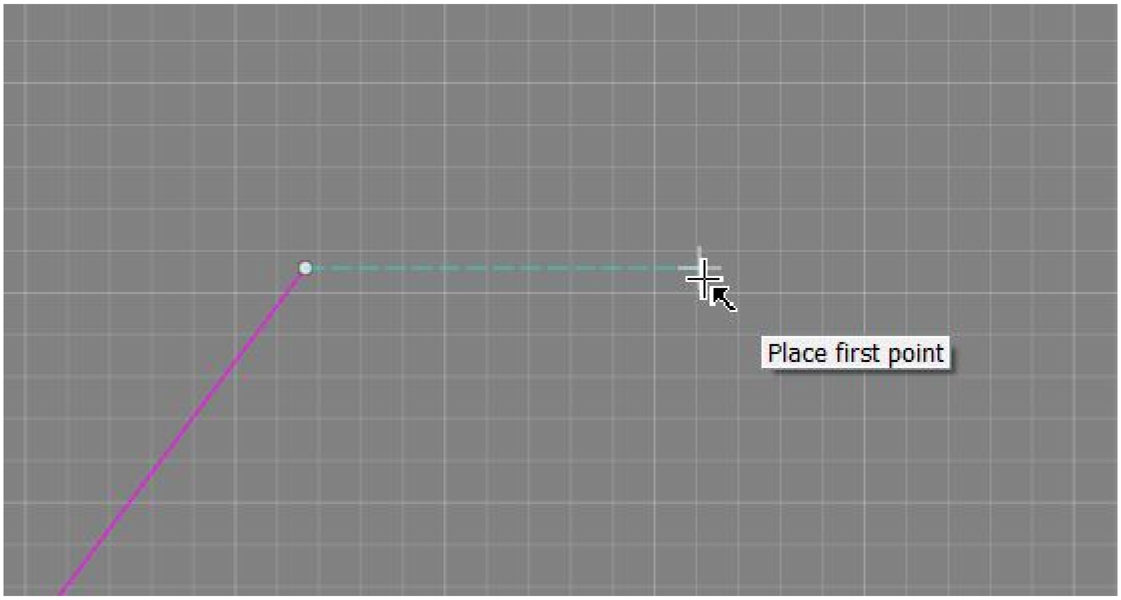


Constraint Inferencing. This term refers to the fact that, when you are creating sketch geometry, you can preview constraints that can be automatically created based on its relationship to other geometry in the sketch. In commands such as Line, you can infer constraints such as perpendicular, tangent, midpoint, vertical/horizontal, etc:

Screencast: http://autode.sk/2e5yMUK



Alignment special case: The one caveat to this is the "alignment" inference. This is the "dotted line" inference:

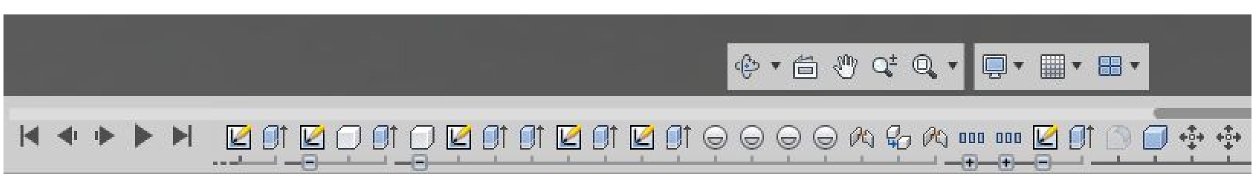


This is a special case, because this does not create a constraint. It is just an alignment hint that is used in the command. So, if you want this one to stick, you will have to apply it manually, by creating a vertical or horizontal constraint between the two points.

Parametric Modeling Tips

Parametric modeling is one of the most powerful tools in the CAD world. But, as is usually the case, powerful tools come with some danger and require some learning, some planning, and some care to get the most out of them.

Parametric modeling is a way to record your design creation recipe in such a way that it can be replayed. Think of this as a "recipe" for your design. This recipe can be re-executed, with different parameters. This recipe is called the design history. In Fusion, this is represented by the Timeline UI:



The timeline is a graphical representation of your design history. It represents the features, sketches, component operations, etc, in the order that they are executed in the design. This is usually the order that they are created, but as we'll see, you can change that.

Tip 1: The Timeline is a time-travel machine

You should think of the timeline as a time machine. Moving the end of design marker:

to a different location in the design (which we call "rollback") is like traveling back in time. Fusion is put into the same state as that point in time. The graphics and browser will only reflect objects that existed at that point in time. This same mechanism is used when editing features and sketches. This is an important concept to remember when editing a design. A common Fusion forum question we get is something like: "I want to project an edge of this Extrude, but when I edit the sketch, I don't see this geometry. How can I do this?". The answer is: "you can't". And, the reason is almost always that the geometry that they want to reference was created after the sketch. So, when Fusion rolls back, that Extrude has not yet been born, so you cannot reference it.

Tip 2: Fusion assembly operations are timeline features, too

This is something unique to Fusion. Most other parametric CAD systems have a separate Assembly and Part documents/environments. Fusion combines them into one single environment. We think this is pretty powerful, but it takes some getting used to. Unlike applications you may be used to, you have to start thinking of some assembly operations as history features as well. Things like creating a new component, adding an instance of an existing component, joints, even assembly hierarchy changes, are all recorded as timeline features. So, as above, if you roll the end of design marker before a "New Component" feature, that component will not exist in the browser or the graphics area.

Screencast: http://autode.sk/2dNsbjR

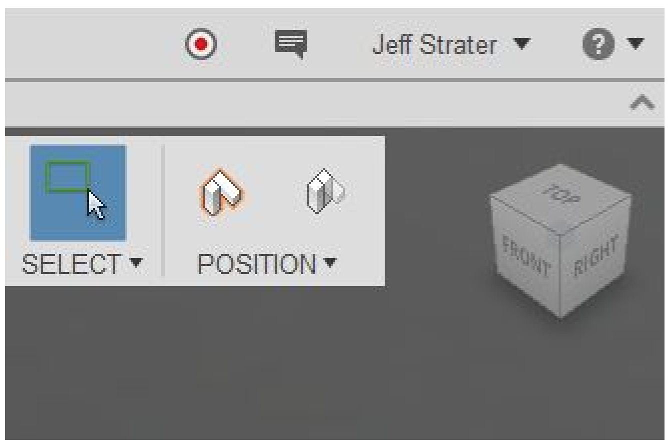
Similarly, Joints are also timeline features. So, if you roll back before a Joint has been created, any motion between the components referenced defined by the Joint will also be unavailable.

Screencast: http://autode.sk/2f6TeIL

Even assembly restructure operations are timeline features. So, if you move a top-level component underneath another component, that will show up as an item in the timeline. If you roll back before a restructure, that operation will also be "undone", meaning that the component will be moved back to the top level. This is unusual for those with Solidworks or Inventor experience.

Component position and Snapshots. Because of this unique way of handling assemblies, there is one more unique aspect to assemblies in Fusion: Component position. Component position is the one exception (kind of) to the “assembly operations are timeline features, too” rule. I won’t go into this much, see Kevin and Mike’s “Fusion 360 Assemblies - Master Class” (AU class link) class for more information. But, this one is worth touching on for just a few minutes.

Component position is not captured as a timeline feature, unless you want it to be. When you move a component, by dragging, using Move Component, or by driving a joint, Fusion will be put into what we call a “Snapshot Pending” state. I’m sure you’ve seen this UI pop up in Fusion:

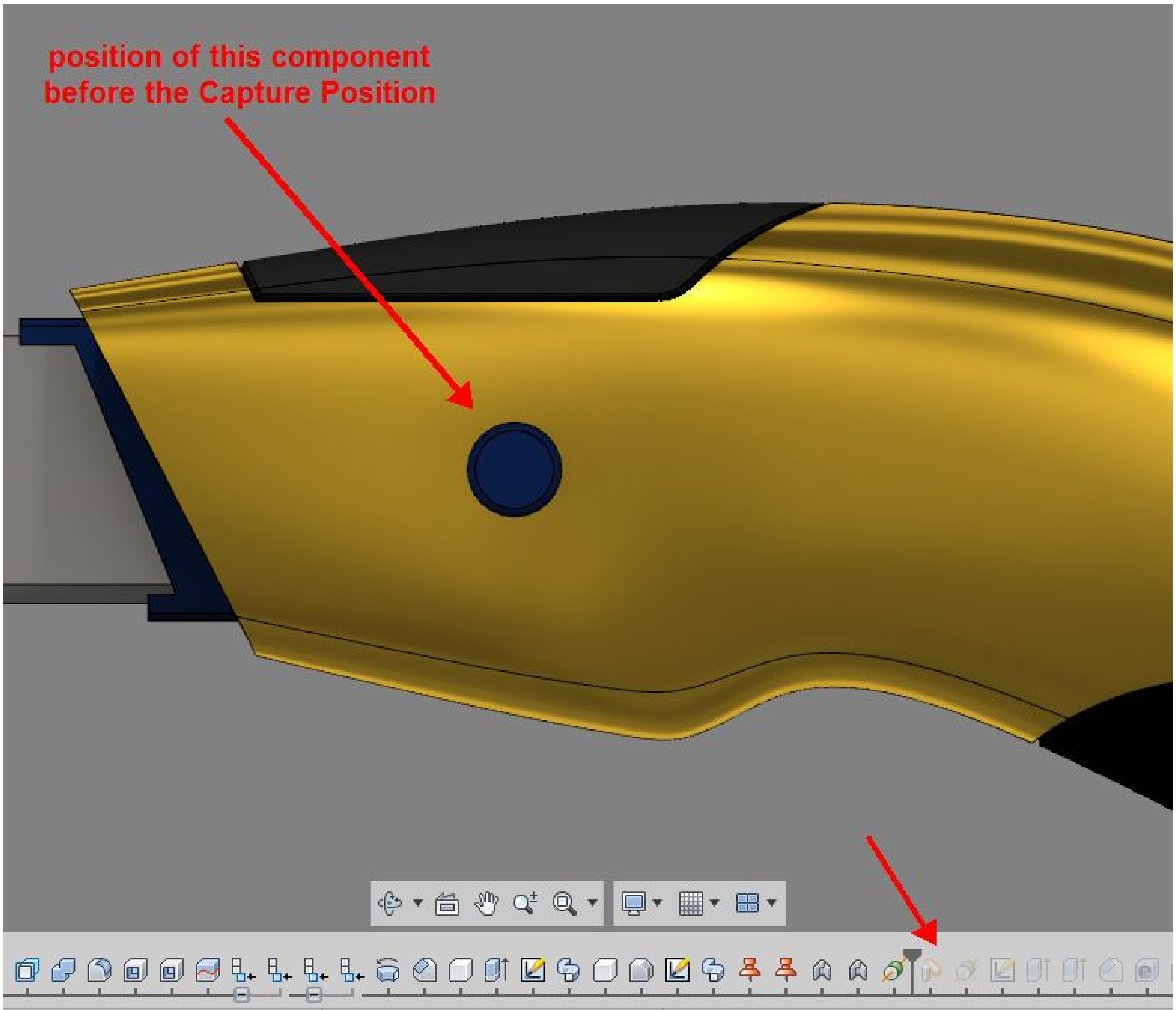


This UI is telling you that one or more components have been moved. You can:

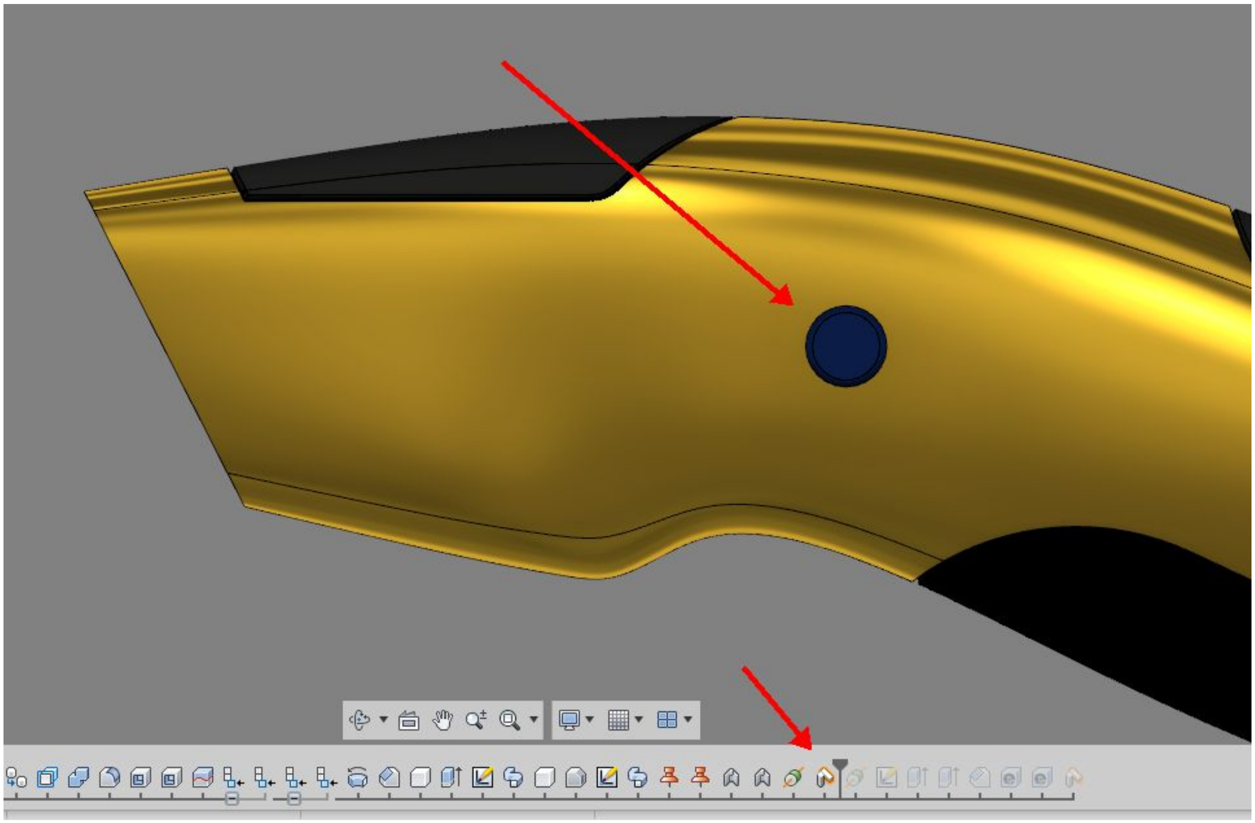
- Ignore it. Fusion will bug you when it needs to about this state
- Reset the state to before the component move happened
- Capture the component position with a “Capture Position” feature

That last option does make the component position a timeline feature. Why does Fusion do this? This allows you to support “position-dependent” modeling. A good example of this is in the Fusion sample “Utility Knife” model.

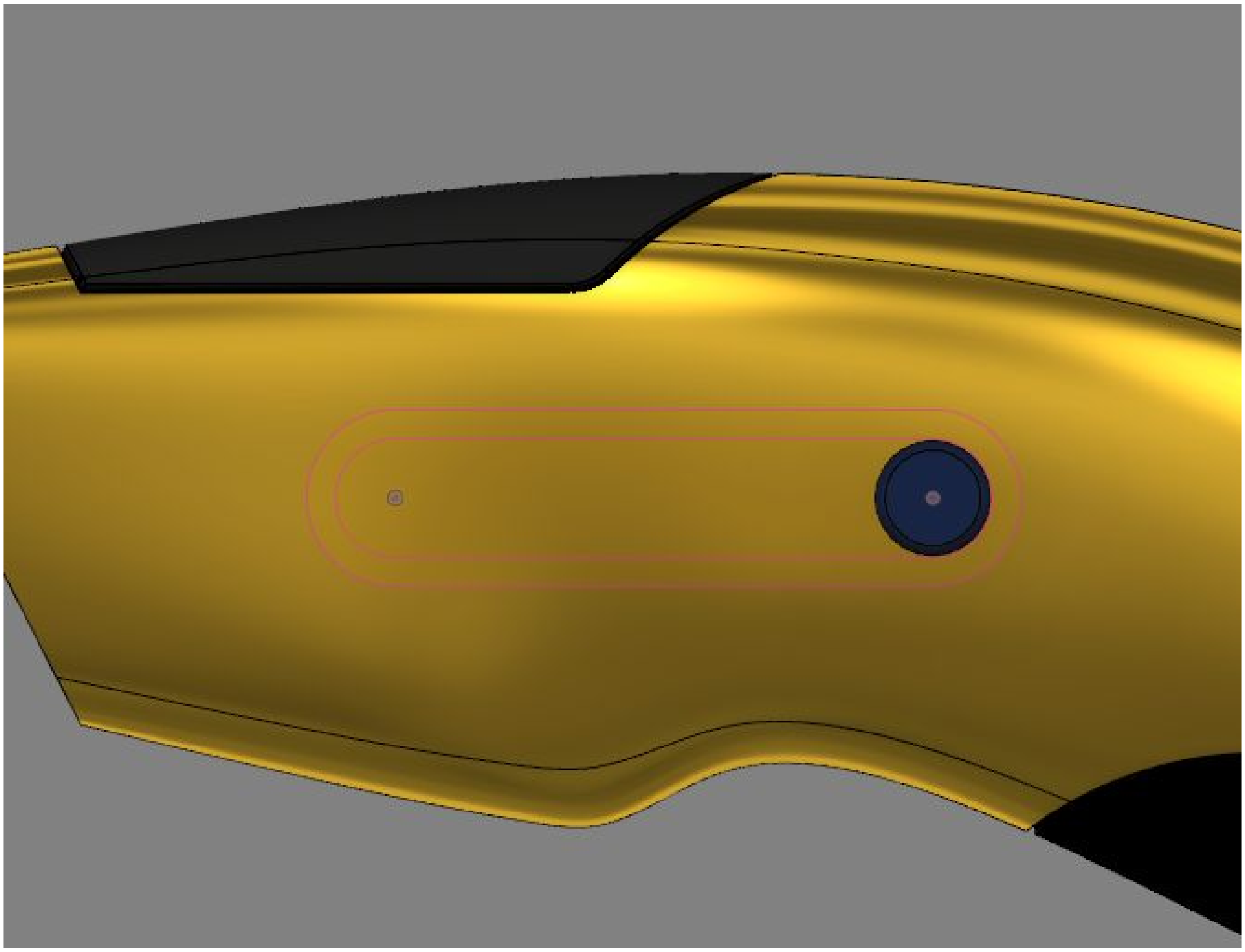
Here is the model before the Capture Position feature:



And here is the position after:



Note that the button has changed position. The position of this component is used (by projecting) to define the profile in the sketch for the slot:



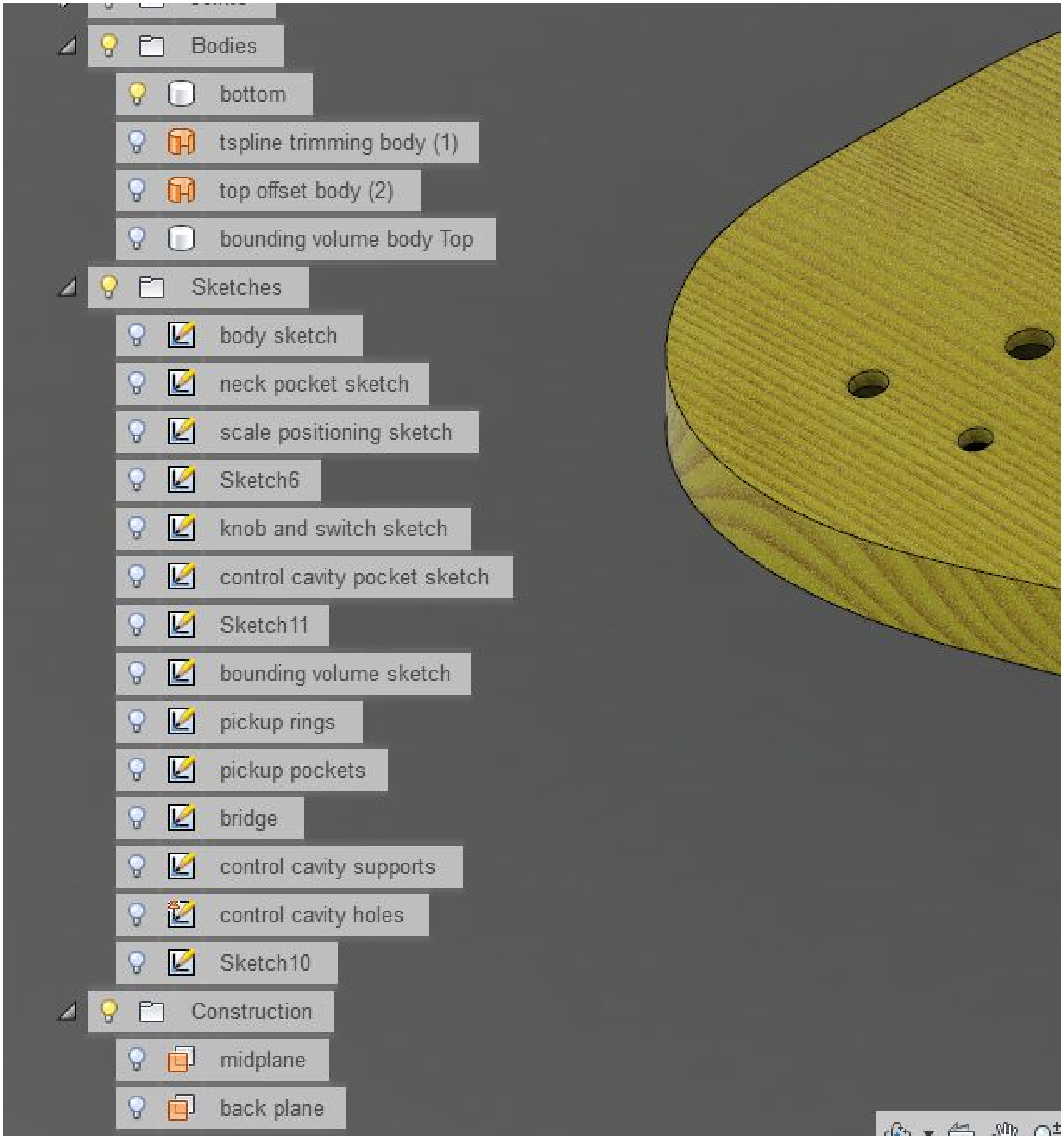
This is a very important concept: You can use the position of a component in two different positions to drive geometry. This is unique to Fusion.

Tip 3: Organize your timeline - you will thank yourself later

I can't emphasize this enough: A well-structured timeline will save you a lot of time when it comes time to edit your design.

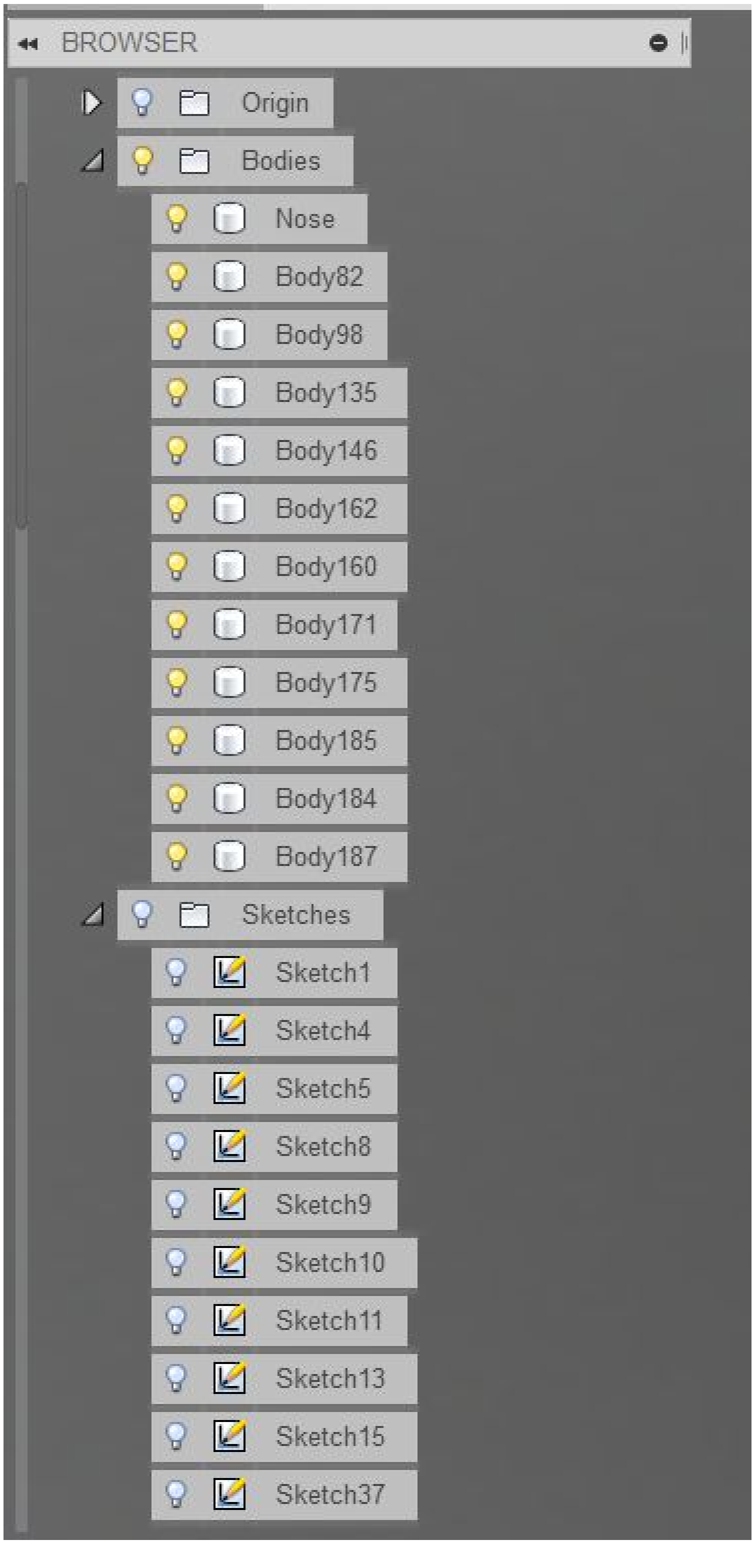
Name everything. Yes, everything.

Nothing is more frustrating than looking at your own design a couple of months down the road, and trying to remember: "now, how did I make that little section of the geometry?" and not being able to remember. The fix for this is to name your features. Name your sketches, work geometry, features, everything. Fusion lets you name every item, so you should use it. The image below is from a guitar design I made last year:



This make is fairly obvious what each body, sketch, and workplane in this model does.

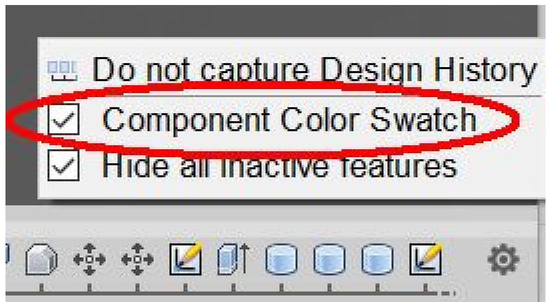
As a counter-example, look at this browser:



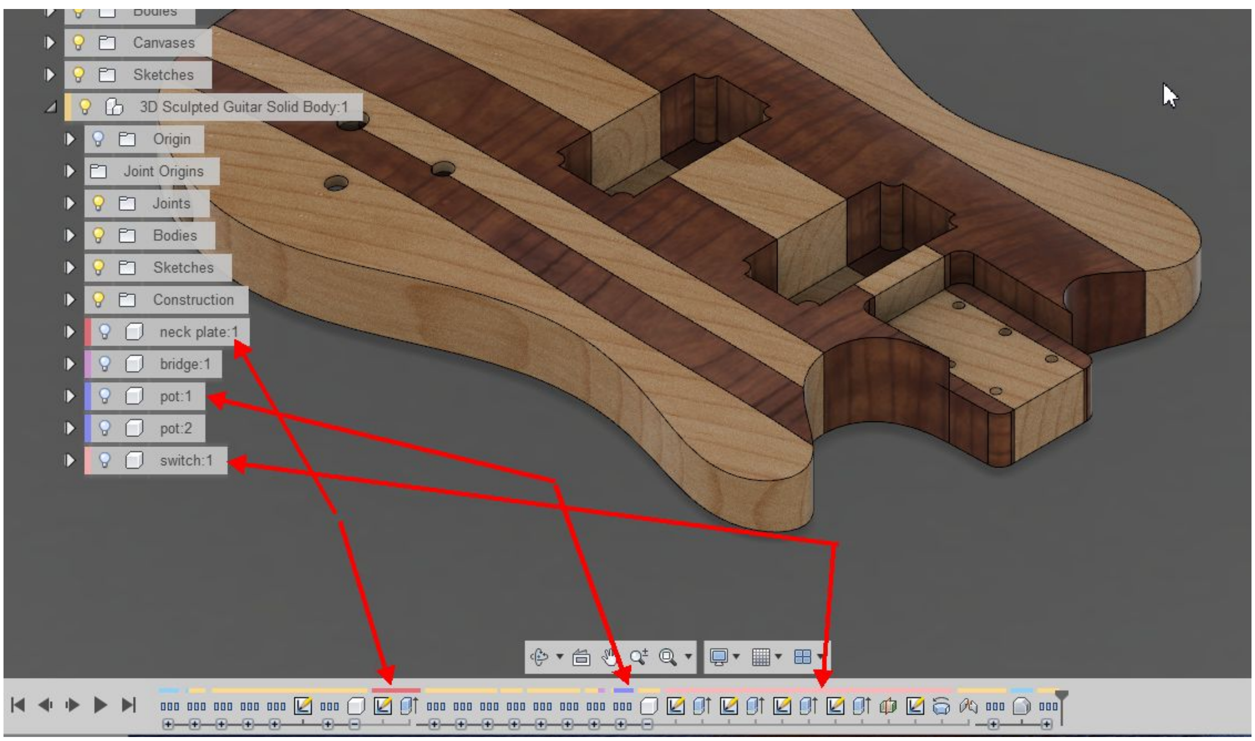
Doesn’t tell you very much, does it?

Group operations for a component as close to each other in the timeline as possible

Yes, this is a hard one to keep, and I am guilty of violating it all the time. It's natural to jump from one component to the other, and Fusion actually makes it easy for you to do this, with its convenient component activation feature. However, you will end up with a timeline that is pretty messy. Use the "Component Color Swatch" command:



to turn on the color swatches in the timeline. The colors match with the color swatches in the browser:



This will help you if you want to know what state your design is in at any given time. I tend to run with this option on at all times.

Note that this is a different tip than R.U.L.E.#1. That rule is all about organizing your design so that your component owns the pieces which make it up. This tip is just about a clean timeline organization.

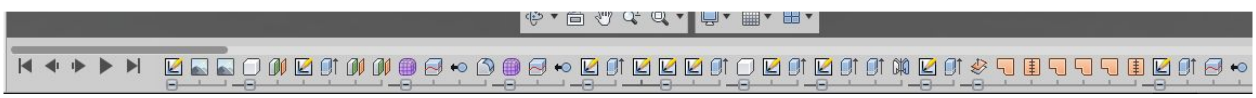
"I forgot to follow this tip. Can I fix my model up afterward?" The answer to this question is "maybe". It depends how your model was constructed. As we'll see later, it all depends on what relationships you have created. If your relationships are nice and clean, you can often reorder features in your timeline to achieve this. Below is a Screencast showing a simple example that illustrates. This model was constructed specifically to illustrate this point. You are unlikely to get this lucky in a real-world design. But, you get the idea:

Screencast: http://autode.sk/2eErtXK

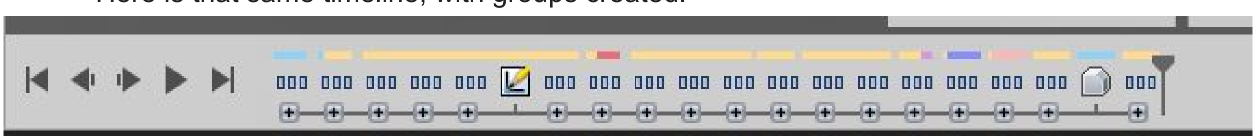
Create Timeline Groups to further organize your Timeline

This tip is used to compress your Timeline to something that is easier to digest. Again, for my guitar body design, here is what the timeline looks like without groups:

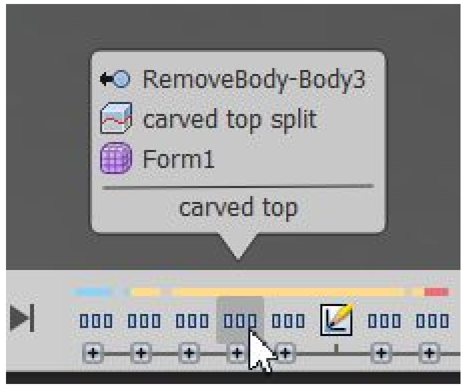
Screencast: http://autode.sk/2eEsrn2



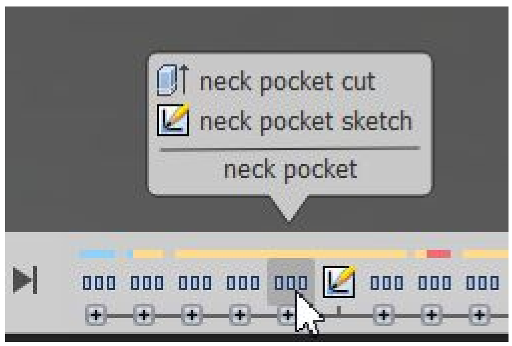
Here is that same timeline, with groups created.



Note that you don't need (or probably even want) a single group for a whole component. But, instead, create groups for "meaningful" sets of features. For instance, I have this group for the set of features that turn a flat top into a shaped top:



And this group, to create the cut for the neck:



Both of these groups are in the same component. These are examples, though, of a meaningful set of features to be grouped together. As mentioned above, you should always name your group as well. You can see above that it's going to be easy to figure out what purpose this set of features serves in the design.

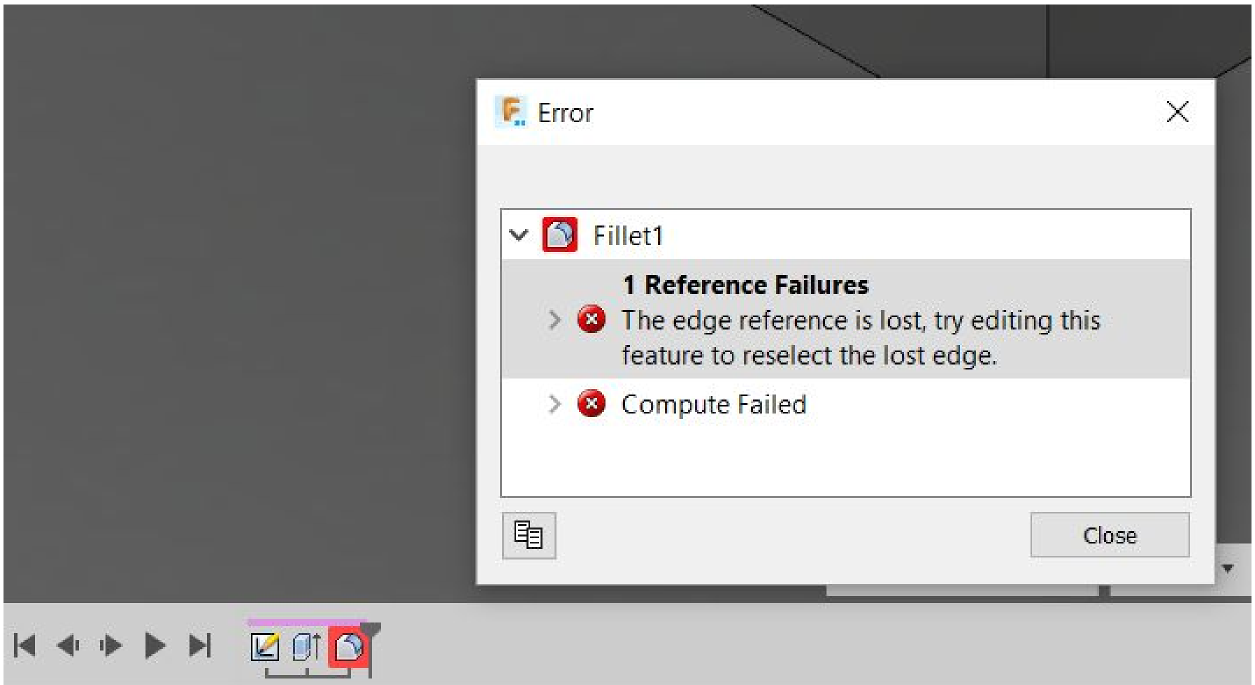
Tip 4: How not to blow up your design when you edit it

Probably the hardest thing about parametric design is dealing with errors that happen when you edit your design. It's ironic: The biggest advantage of parametric design (the ability to edit a design and have it regenerate with new values of parameters) is also its weakest point, because those same edits are what cause the errors.

Fortunately there are a few techniques that can help minimize these problems. The key is to have just a bit of insight into how Fusion works "under the hood". Once you understand some of the theory behind Fusion's compute, you can predict which edits will succeed, and which will fail, and further, how to fix those failures when they do happen.

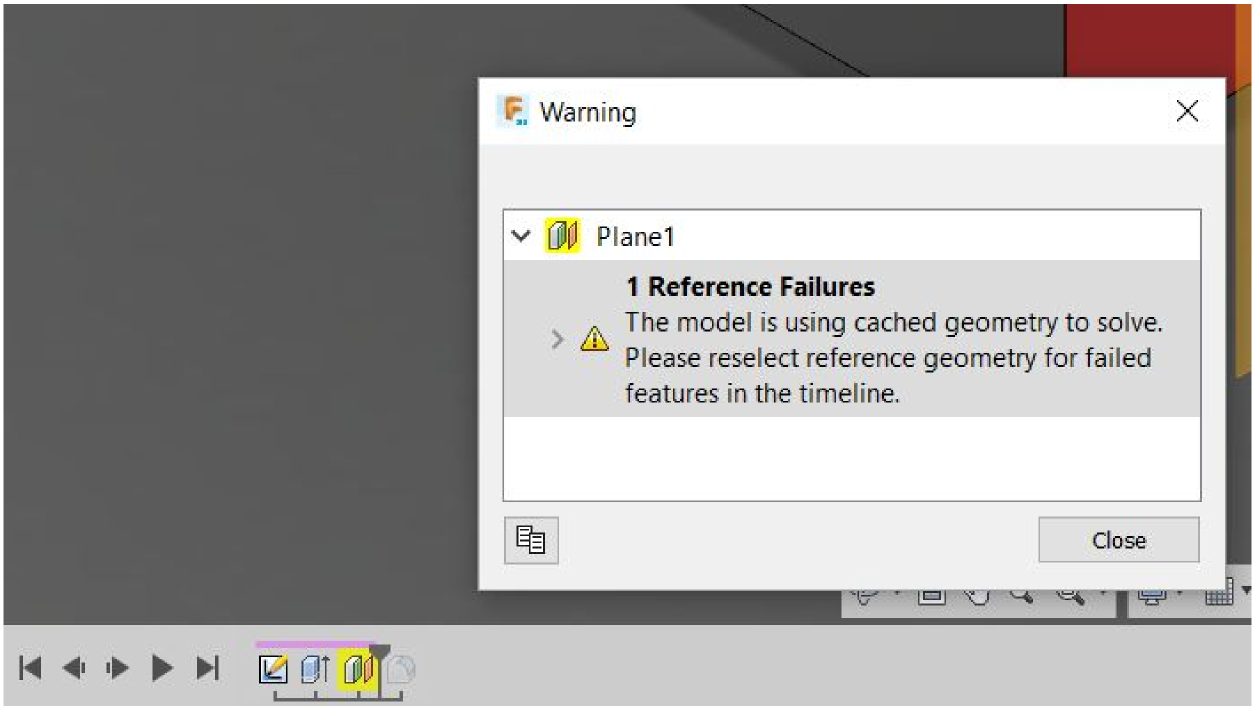
Understanding errors/warnings in Fusion

What do the different color in my timeline mean? This is also good to understand. When failures occur in Fusion, there are two different severities of result. The first is an error status:



This red status means that the feature has failed, and cannot be computed at all.

The second is a *warning* status:



This yellow status means that something is wrong with the feature, but Fusion can proceed anyway. Most of the time, the feature is able to compute using cached geometry. We put this into Fusion in an attempt to make designs more robust. I now believe that this was a mistake on our part. The fact that this is just a warning is giving false comfort that your design is OK, when in fact it is not. Ignoring these warnings can be a very bad idea, unless you understand the warning in great detail.

Let’s dive down into these status types a bit more.

Error status: As above, an error occurs when a non-recoverable failure has occurred. Meaning that the feature just cannot compute at all. Examples of this include: A Fillet feature that cannot find the edge to fillet. Without that edge, there is no fillet. Or, a Shell feature where the shell operation itself cannot succeed because of the geometry.

Warning status: This is the tricky one. In these conditions, Fusion does keep enough information around to allow the feature to continue. For example, if you have an offset workplane from a face, and that face cannot be found. Fusion keeps around a cache of the plane from that face on the last successful compute. If the face cannot be found, the workplane can still succeed, and will compute at its last known location. However, (and this is extremely important to understand), that offset plane is no longer tied to the face, so subsequent updates to the position of that face will not correctly update the offset plane, and any downstream features from that plane are suspect at best. Here is an example screencast (this one has audio explaining it):

Screencast: http://autode.sk/2eGIrF4

As you can see from the Screencast, though these warnings are not fatal, they can have a profound impact on your design, and it is good to be aware of what these mean.

Relationships are the source of most problems

I'm resisting the temptation to extend this tip to life in general… But, certainly in Fusion, most of the errors and warnings in your designs are caused when you create a relationship from one object to another object, and the referenced object cannot be located. You have filleted an edge, and when that fillet computes, it cannot find the edge. Or, you created an Extrude with a "To Object" termination type, and the face you selected cannot be found. A good percentage of failures are in sketch projected geometry, I believe: You projected an edge into your sketch and that edge cannot be found.

As a result, you need to think a bit about whether you need a particular reference to an external object, and whether there is a way to create a reference to a more stable object, or avoid it altogether.

Tips for more stable relationships

We are not giving marriage counseling here. But, it is true that some references are more likely to succeed when your design is edited, and you should use those whenever possible. Assuming you do need to have the reference at all, then:

1. **Origin work geometry is the most stable of all.** These guys never change. If you can get away with projecting an origin plane from your component into a sketch, and dimensioning from that, instead of projecting a body edge, I can guarantee that this reference will never go bad.
2. **Work geometry that only references origin geometry is also good**. If you can create an offset plane from the YZ plane, and project that into your sketch, you also will have a very stable relationship.
3. **Not all work geometry references are inherently stable.** This is worth noting. The above two hints might seem to imply that somehow work geometry is more stable than BRep geometry. This is not true if that work geometry also contains a BRep reference. For instance, if you offset a workplane from a face, you might think that it is fine. However, all you are doing is adding one degree of separation from the BRep reference.

If the BRep reference inside of the offset workplane fails, then your workplane is no good, either.

4. **For BRep references:**
    a. **Body references are more stable than other BRep references.** Select an entire Body as a reference, rather than an individual face or edge, if you can. The example in the screencast below illustrates why:

       Screencast: http://autode.sk/2ecMoxN

    b. **Face references are more stable than an edge reference.** An edge is relatively easy to fail, but an entire face is likely to succeed. So, for instance, if you are projecting BRep geometry into a sketch, if you can, project the entire face into the sketch, not just one edge.
    c. **Edge references are less stable that face references.** The underlying reason for this is that in order to resolve an edge reference, Fusion needs to resolve two face references (an edge is the intersection of two faces). So, it is twice as likely to fail as a single face reference.
    d. **Vertex references are the least stable of all.** Avoid these if you can. For a vertex reference to succeed, Fusion needs to match 3 different faces.

5. **Sketch references can also be very stable, as long as you are careful.** Often times, you can just project from one sketch into another. Unless you go into the source sketch and manually delete a line, that line will be there forever.

What can cause a reference to fail?

The other bit of "Fusion theory" that is worth a few minutes to understand is: What sort of edits are likely to make my references fail? How can I prevent some of these failures from happening in the first place?

Without going into too much technical detail, you need to understand just a bit about how Fusion tries to resolve references, especially BRep references. BRep entities are "named" using information that primarily comes from three sources:

- The ID of the sketch entity, for sketch-based features (Extrude, Revolve, Sweep, etc)
- The ID of the BRep entity for non-sketch-based features (Fillet, Shell, Draft, etc). This is just one level of indirection, since most geometry is based on sketch-based features. So, an edge for Fillet has most likely been produced by a sketch-based feature.
- Information about the type of feature that created that entity

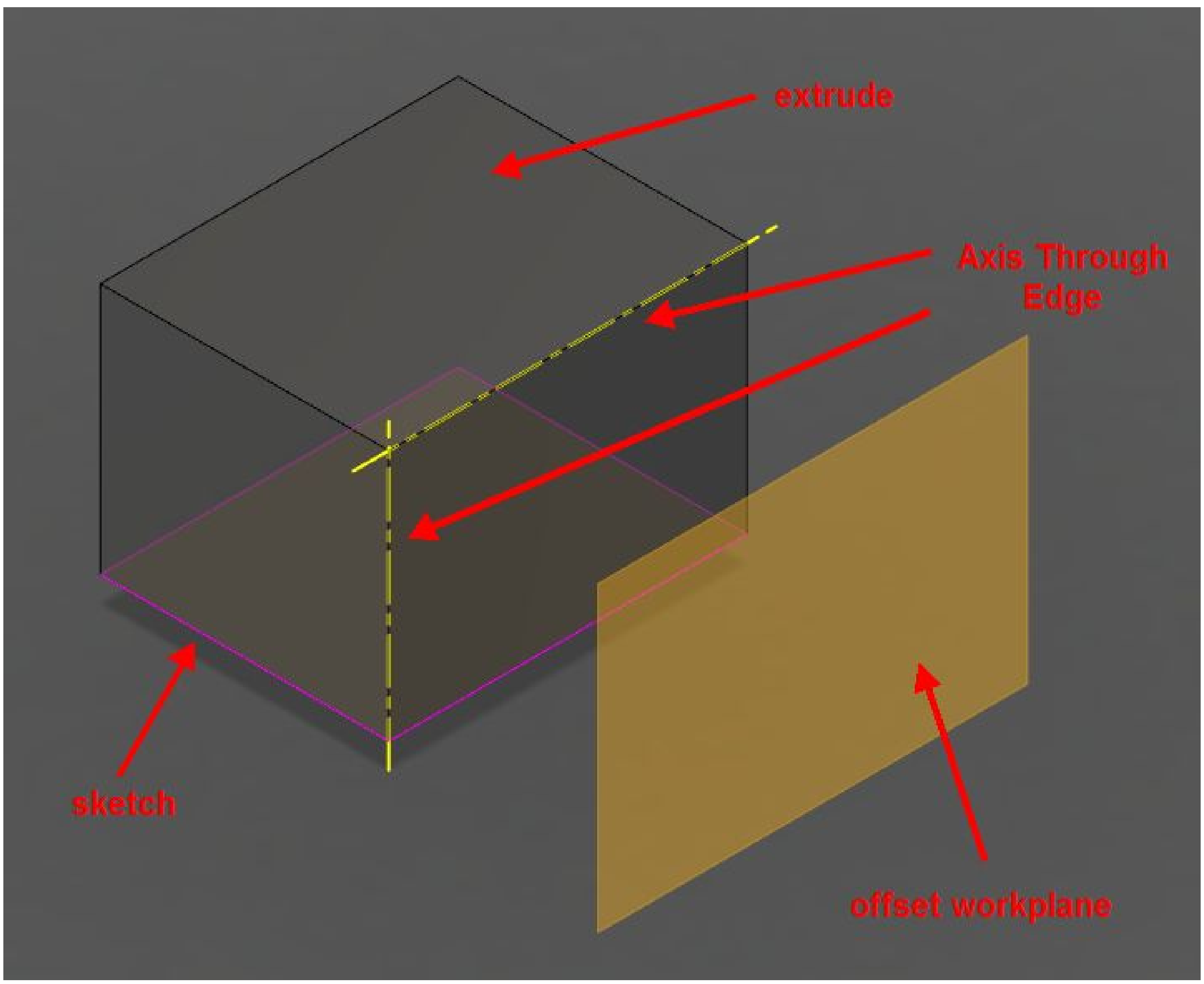
I'll run through a couple of examples of how failures can be introduced. This won't cover all cases where failures can happen, but it will give you an idea of types of edit to avoid to prevent some failures.

The ID of a sketch item is changed

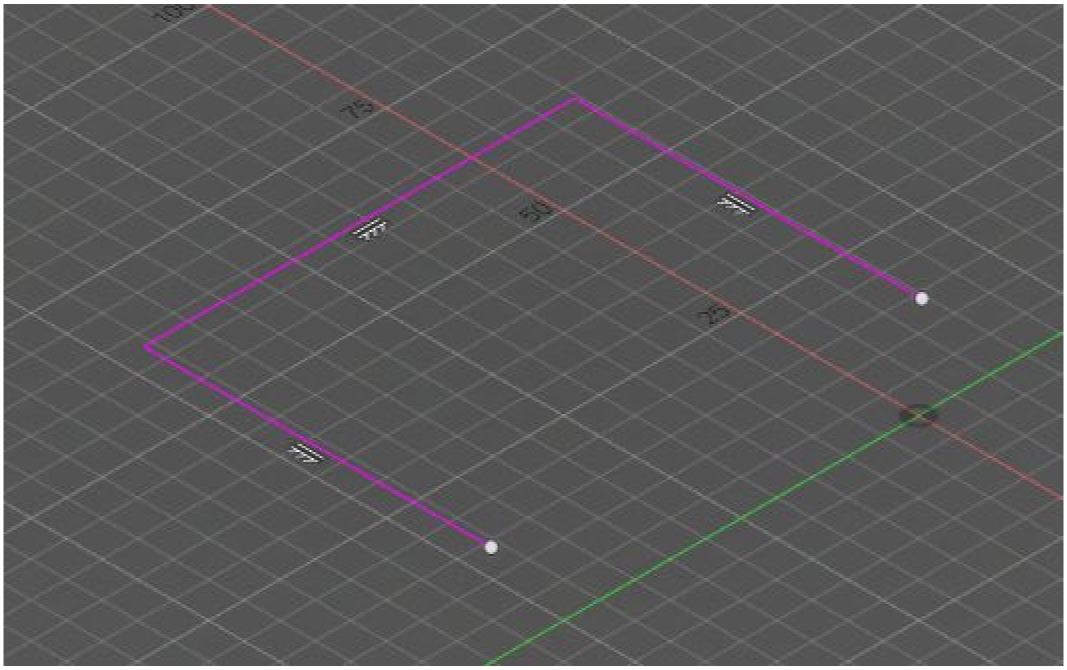
This is probably the most preventable reference failure type. As we said above, BRep geometry is named from the ID of the sketch geometry that created it. So, if you change that ID, bad things will happen. Here is a screencast that illustrates what can happen with a very simple edit:

Screencast: http://autode.sk/2ecPJg5

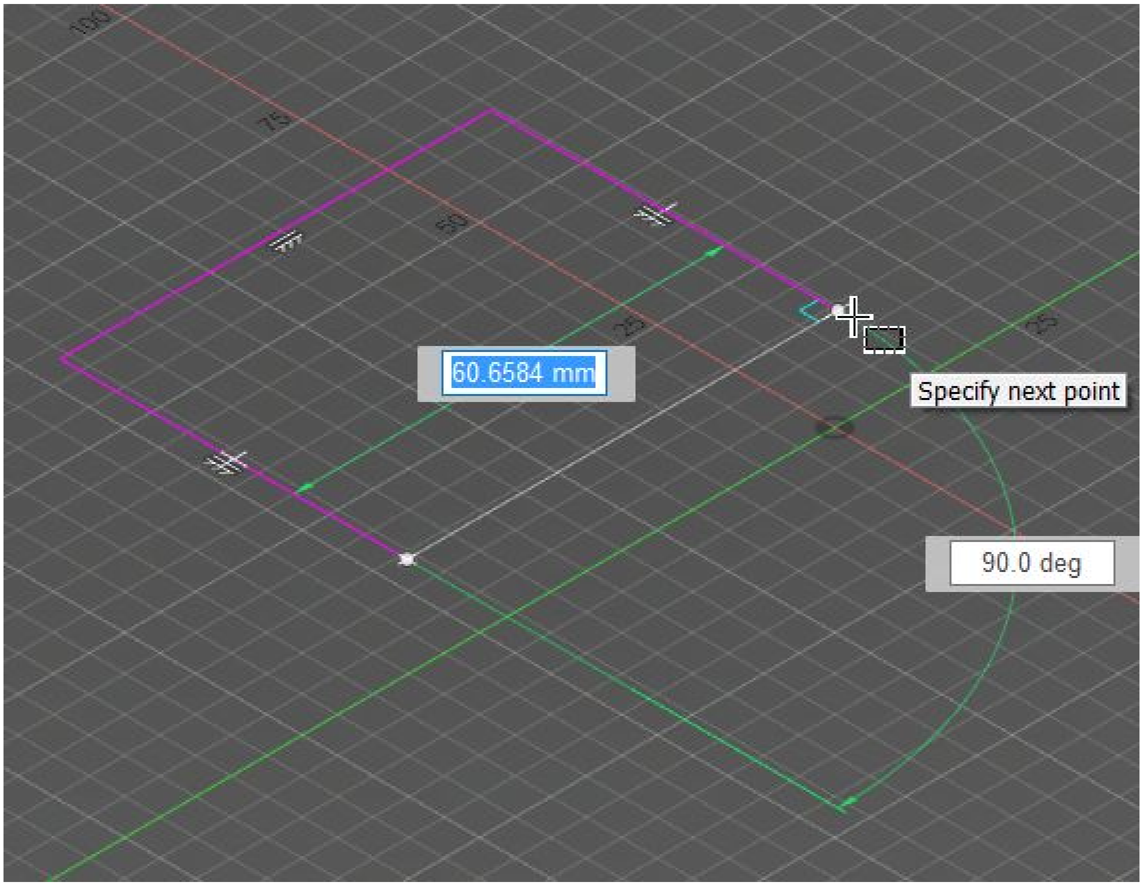
This case starts with a simple design: One sketch, one Extrude, and a few BRep references to a face and edges of that Extrude:



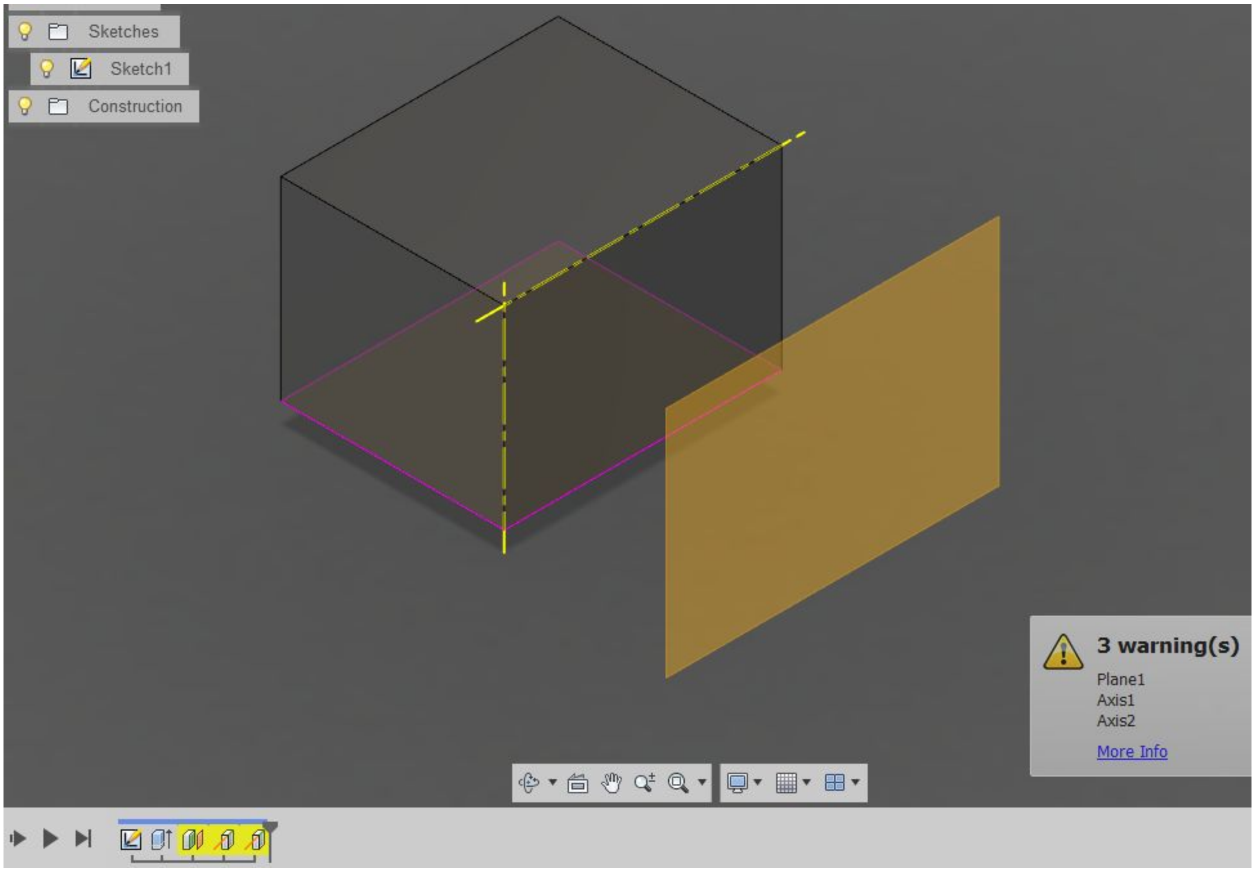
Then, edit the sketch, and delete a line:



Then, re-create the line:



When you exit the sketch, all 3 work geometries will fail:



How to avoid this kind of error: Don't delete sketch geometry without some careful planning. If you can edit it, you will get much better results. While the above example is pretty simple (who would ever delete a line and re-create it later?), there are cases where this is less obvious. For instance, a sketch spline curve. It is easy, sometimes, to get your spline so screwed up that you just want to delete it and start over. If you have BRep geometry derived from that spline, it would not be a good idea.

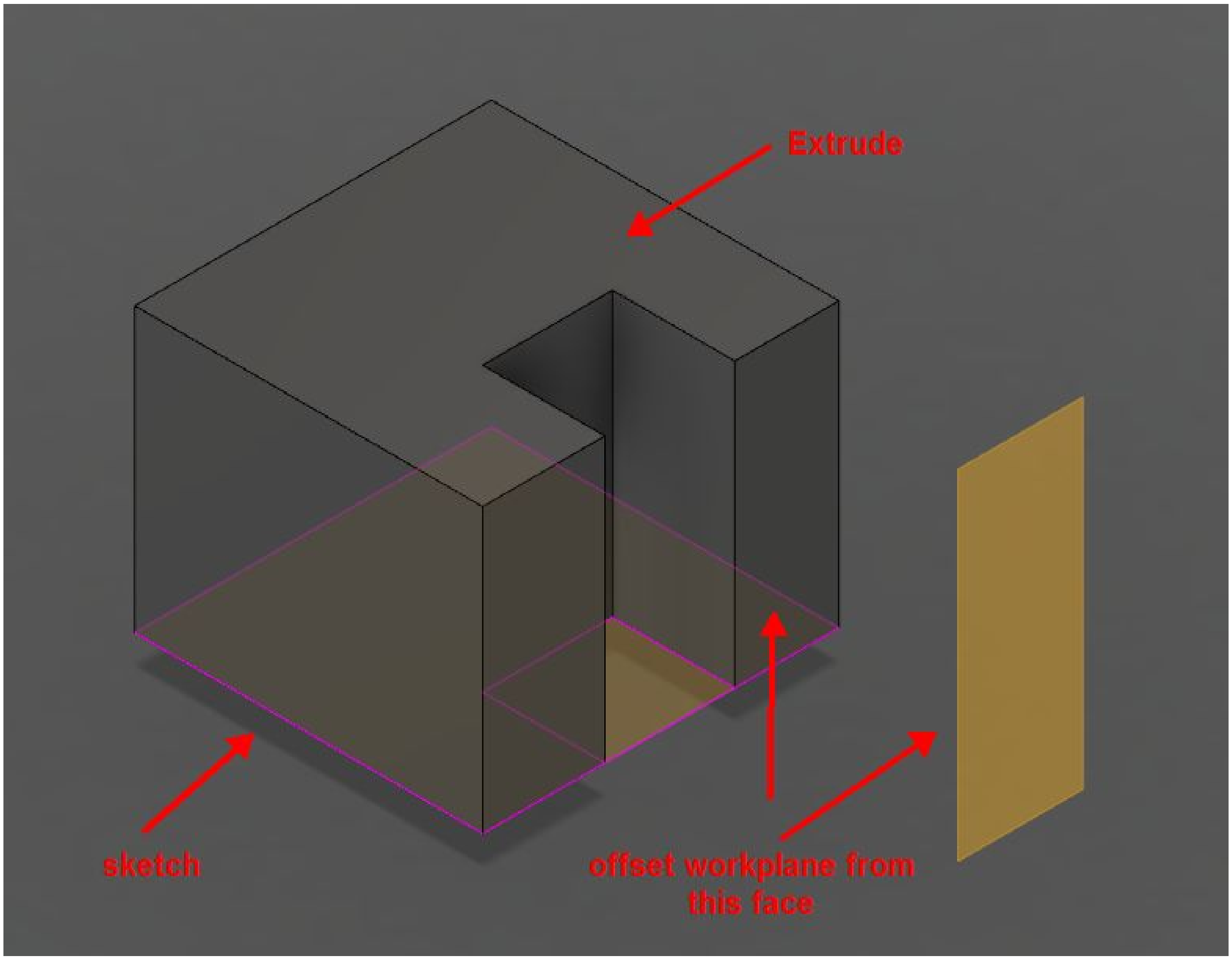
Also, remember that there are other ways to change a sketch curve's ID. Operations such as Break Curve can be dangerous. Break Curve splits a curve into two pieces. One will get the original ID, but the other will get a new ID. If you are not careful, this can lead to surprising results.

Edits to sketches or features that occur before the reference

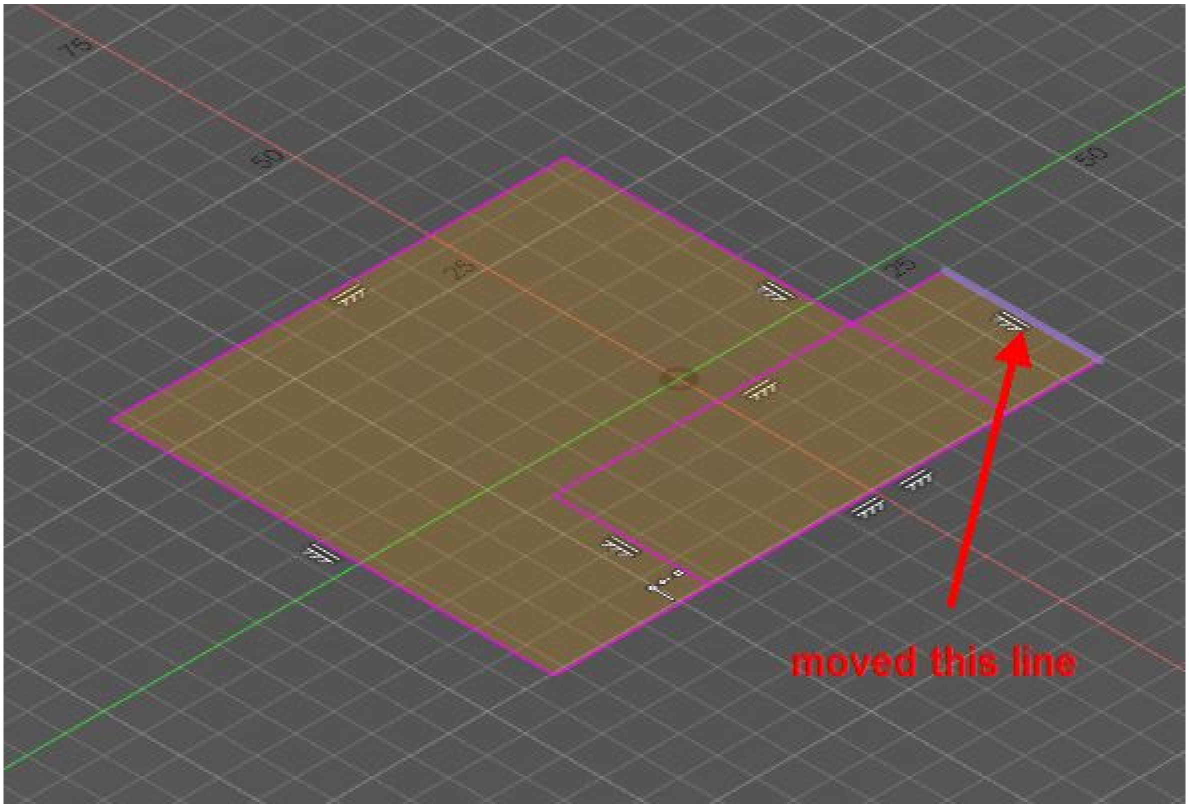
In these types of failures, changes have caused some BRep entity to no longer exist. Here is another simple case:

Screencast: http://autode.sk/2ejZvwc

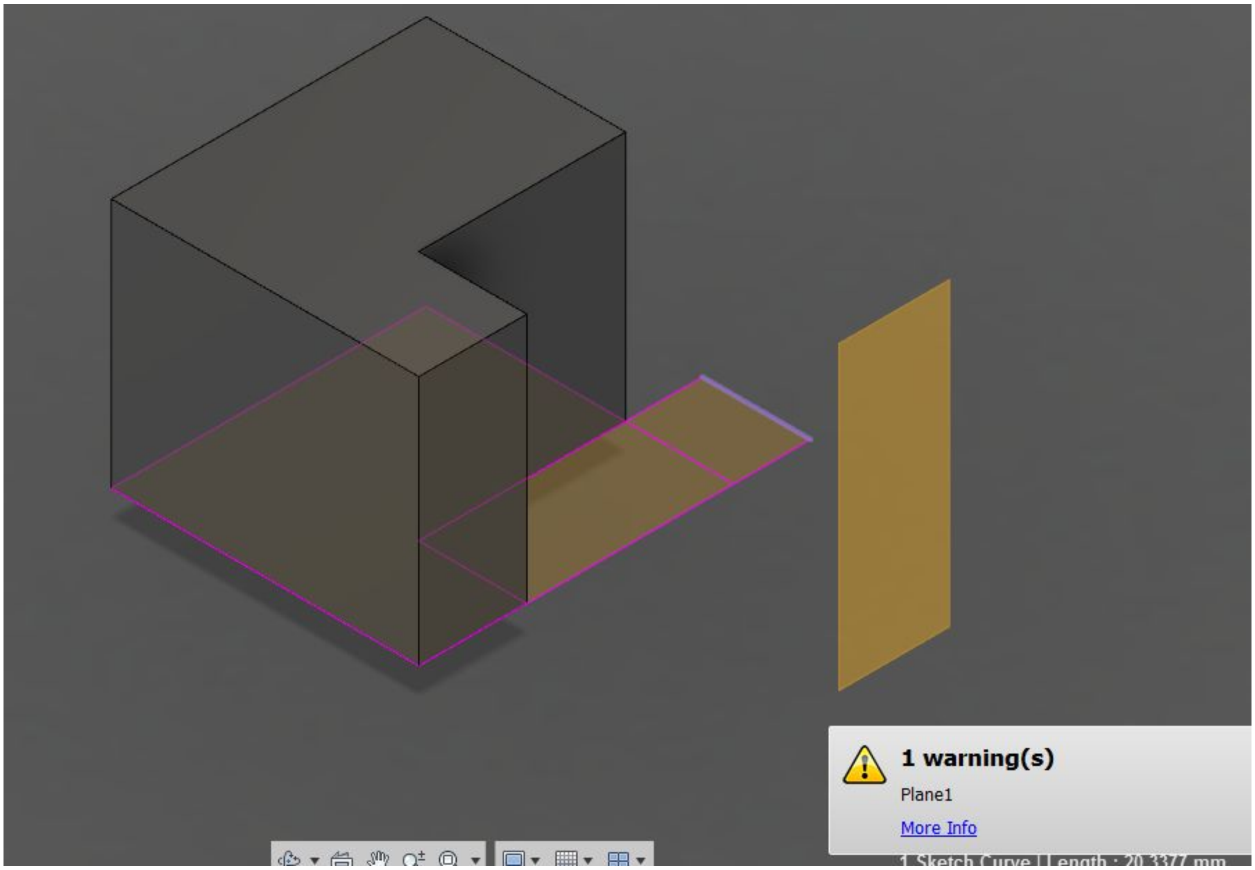
Start with another simple sketch and Extrude:



Edit the sketch, and drag the small rectangle so that it extends outside of the larger one:



When you exit the sketch, the workplane fails:



In this case, the face on which the workplane is based is eliminated by a sketch edit, so the reference can not be resolved.

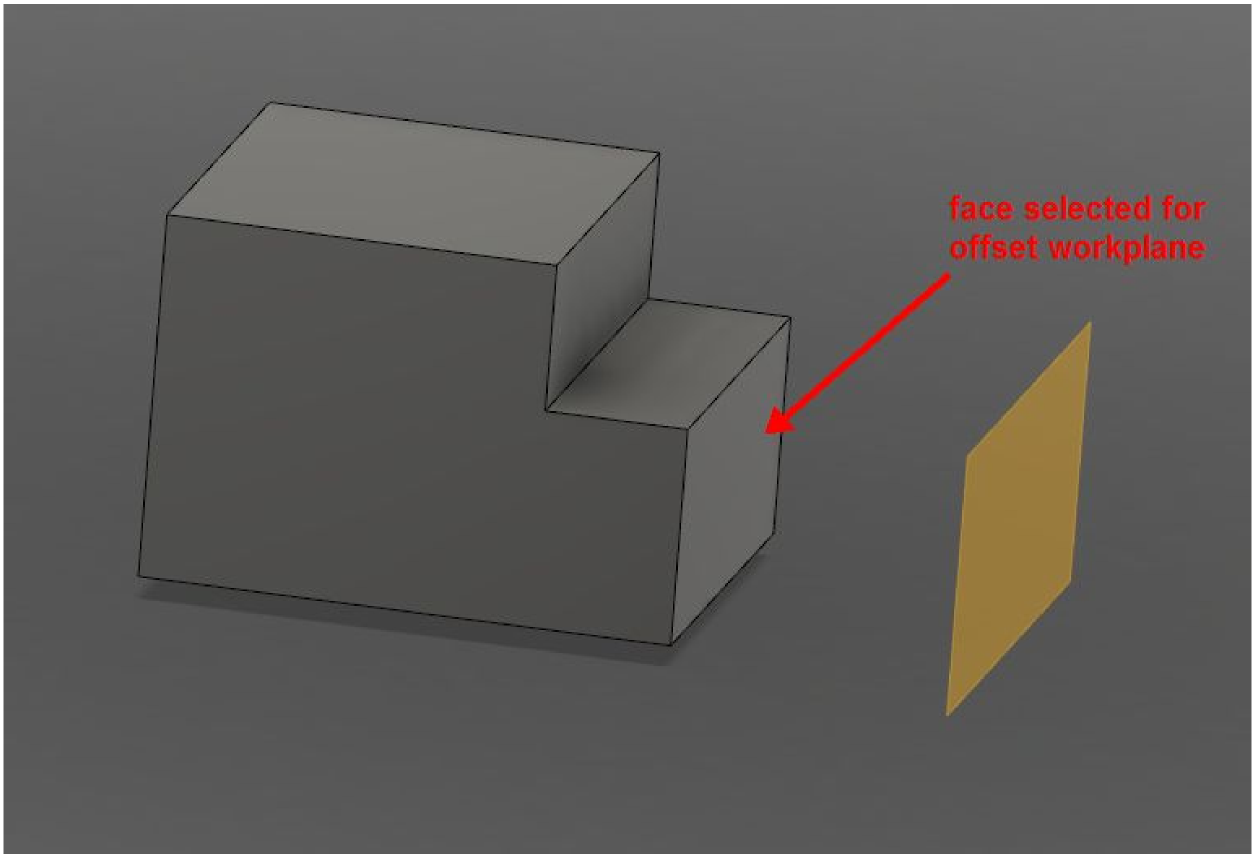
Why does this not cause an error?

This is an interesting example. Take a look at the screencast:

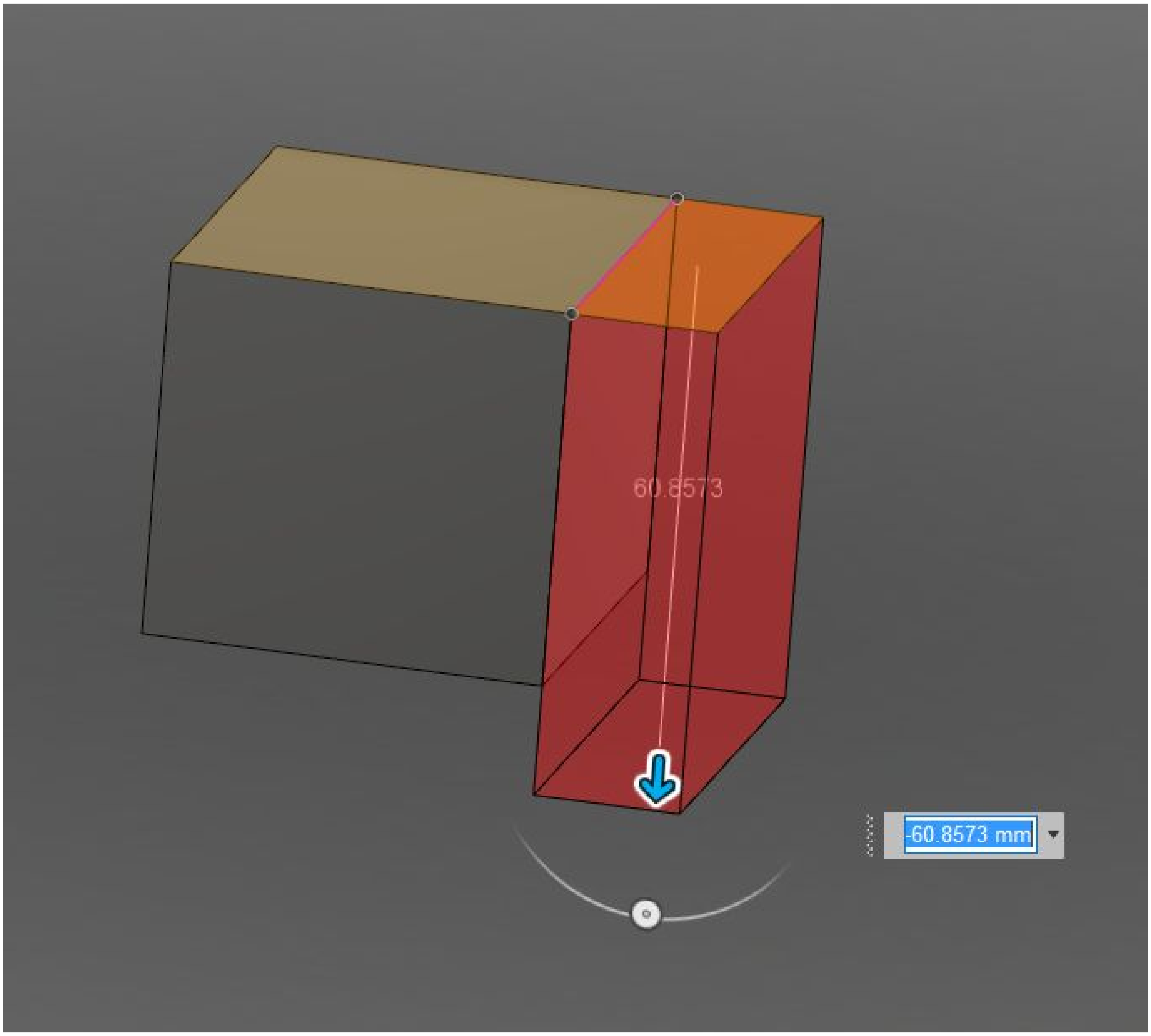
Screencast: http://autode.sk/2eGWVF2

In this case, I edited an Extrude feature that caused the face that the workplane was based on to no longer exist.

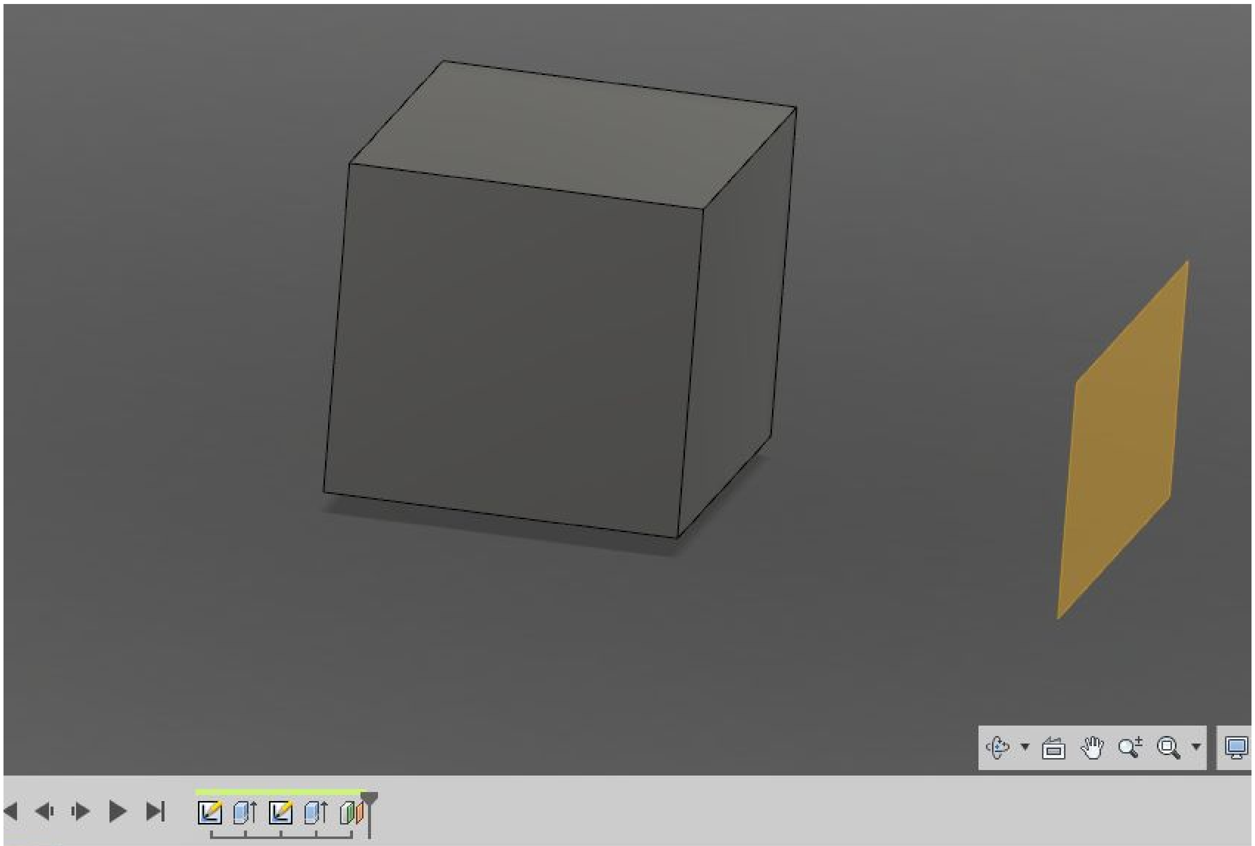
Starting condition:



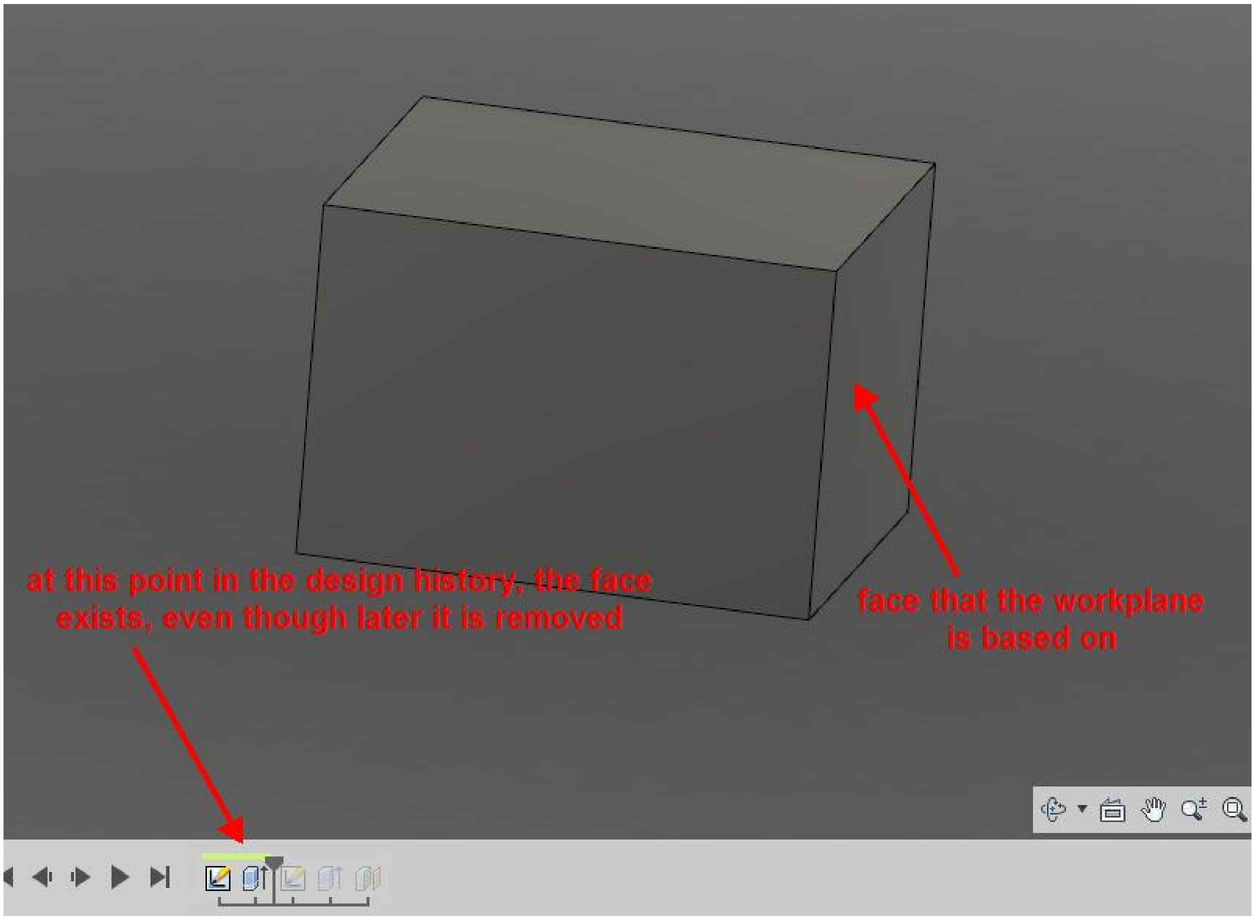
Edit the second (cut) Extrude, so that it wipes out the face:



Yet, the workplane does not complain:



Why did this reference not fail? The difference between this and the previous example is: In this example, that face existed *at one point in the timeline*:



Even though the next Extrude kills that face, Fusion has already resolved this reference, so can happily continue.

New features have been inserted in the timeline

This one is less common, but can happen.

Use Compute All as a design checker

We call this the “Compute All Paradox”. You should never need to do a Compute All. If Fusion was a perfect application, this command would be useless. In fact, it was not part of the original parametric release. But, we had this command for our own internal use, and we realized that it is still a useful tool, so we included it.

What we recommend is to use this command like a “design checker”. If you can do a Compute All, and your design shows no errors, then you know your design is solid, and that is a good time to create a new version, just in case subsequent edits do introduce errors.

How can I fix my references when they do fail?

OK, so now we've shown you how to avoid some reference failures, and you have some "theory" of how these failure can occur. So, you should see much fewer of them. However, they do still occur. So, how can you fix them?

Unfortunately, Fusion is a bit immature in this regard. There are not a lot of automated tools to help with this. Most of it is purely manual.

The easy cases: Feature failures

Some failed references are easy to fix. Failures such as "To Object" failures in Extrude, you can often fix these just by editing the failed feature. For example, in the following Screencast, I show how to repair the failed work planes and work axes from an earlier example:

Screencast: http://autode.sk/2e3cJmA

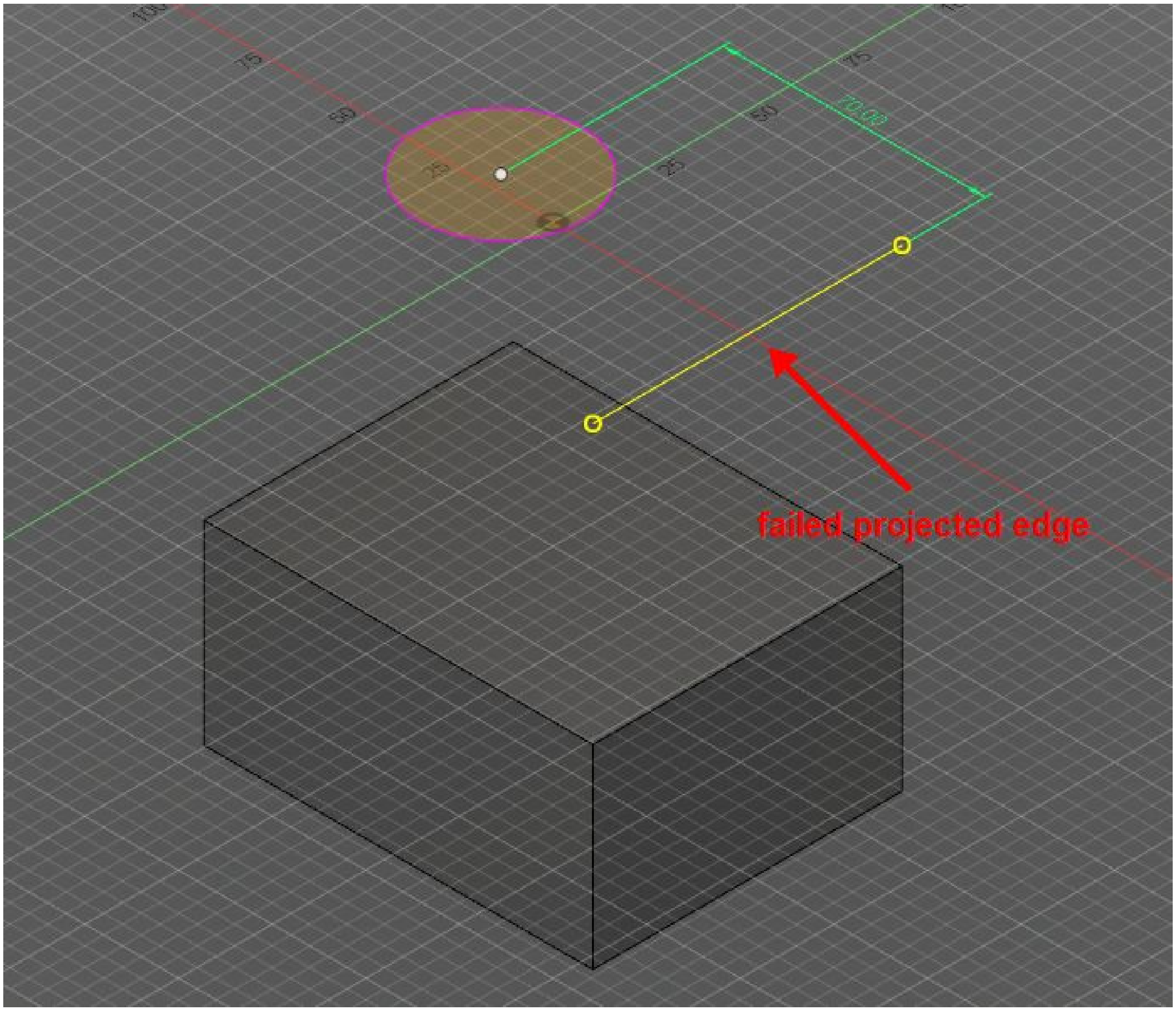
A harder example: Broken sketch Projections

The most common failure is also the hardest to fix, unfortunately. If a projected sketch entity fails because of a missing reference, it can be very challenging to repair. There are a couple of reasons why this is a challenge. First, there is no obvious visible relationship between the sketch curve and what it was projected from, so unless you remember, it can be hard to figure out where the original was. (we are working on this, by the way, and hope to be able to have a mechanism to show this for failed projected geometry soon). Second, there is no "repair projected geometry" mechanism in Fusion. Your only options are really to:

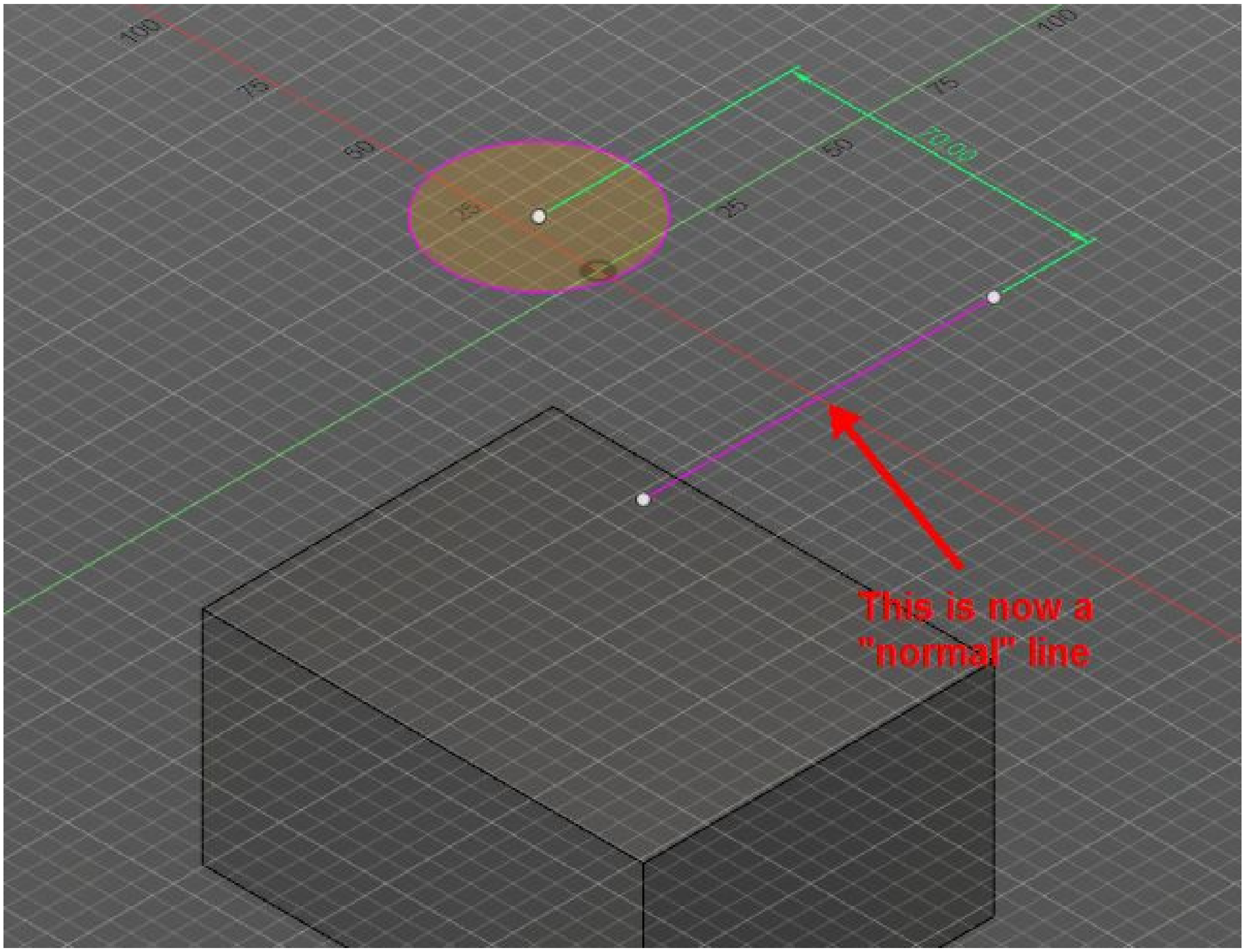
- Use Break Link to convert this into a regular sketch object. This will make the error go away, but will leave you with an unassociated sketch curve, which will not update when its source.
- Delete the projected entity and re-Project (assuming you can find what to project). This will "repair" the projection, but deleting the original will delete any references - dimensions/constraints. These will need to be re-created. Also, because of the issues discussed here that comes from changing the ID of a sketch object, <u>this method could actually cause further failures downstream</u>.
- Hybrid. This is a bit of a hack. But, there is a way to make this work. If you do Break Link on the current projection, then re-Project the new entity, then make the new projection and the unassociated curve coincident, you can repair this, without loss of associativity, and without downstream failures. It doesn't work for all curve types (it's hard to make a spline "coincident" to another spline).

- Screencast: http://autode.sk/2eiDWxA

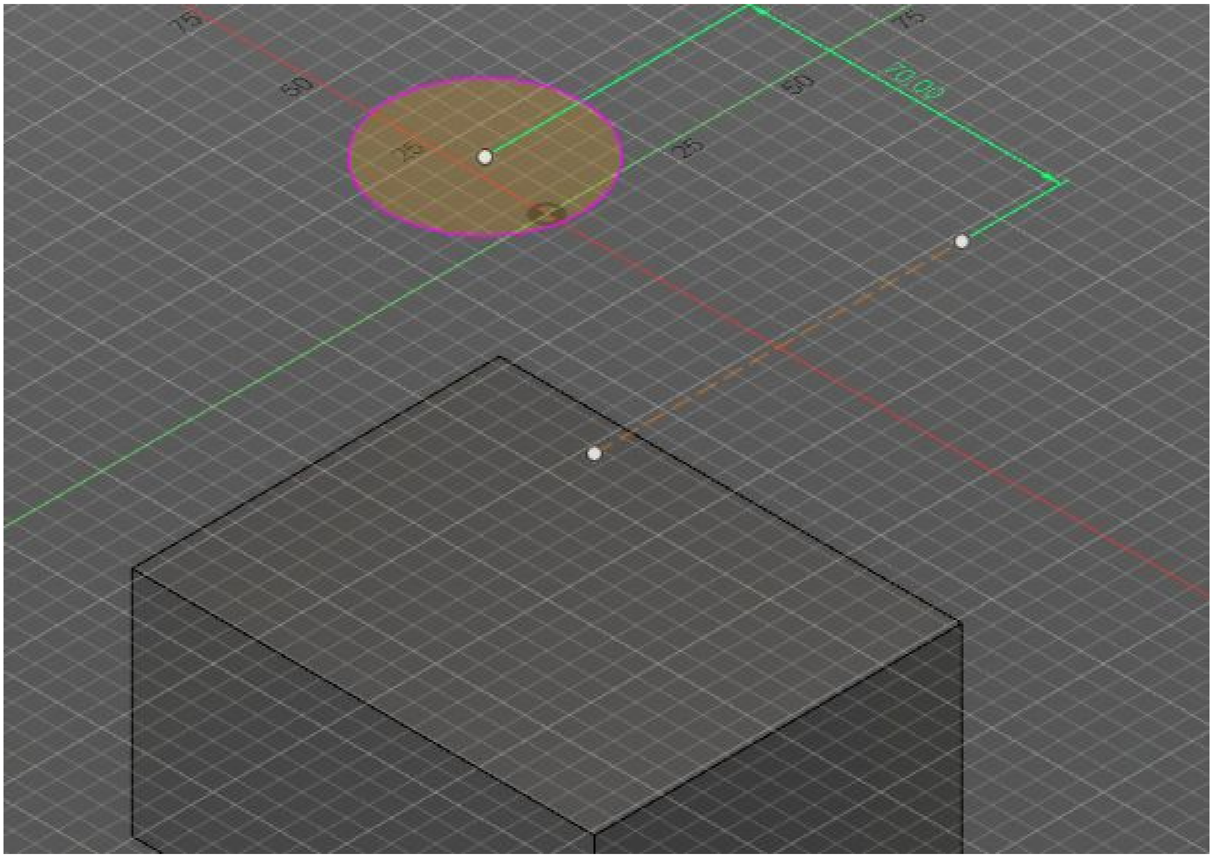
  Starting with a failed projected edge:



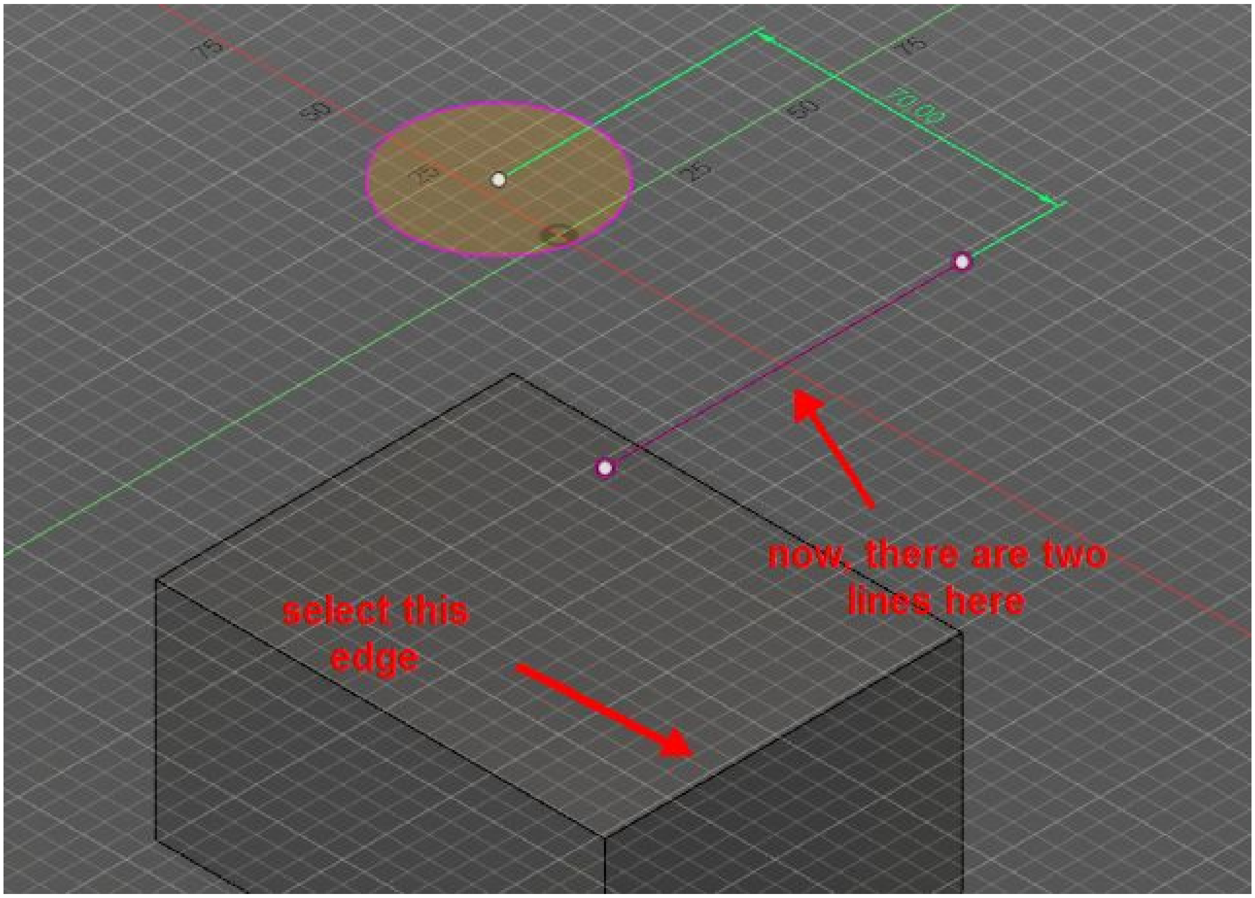
Break Link:



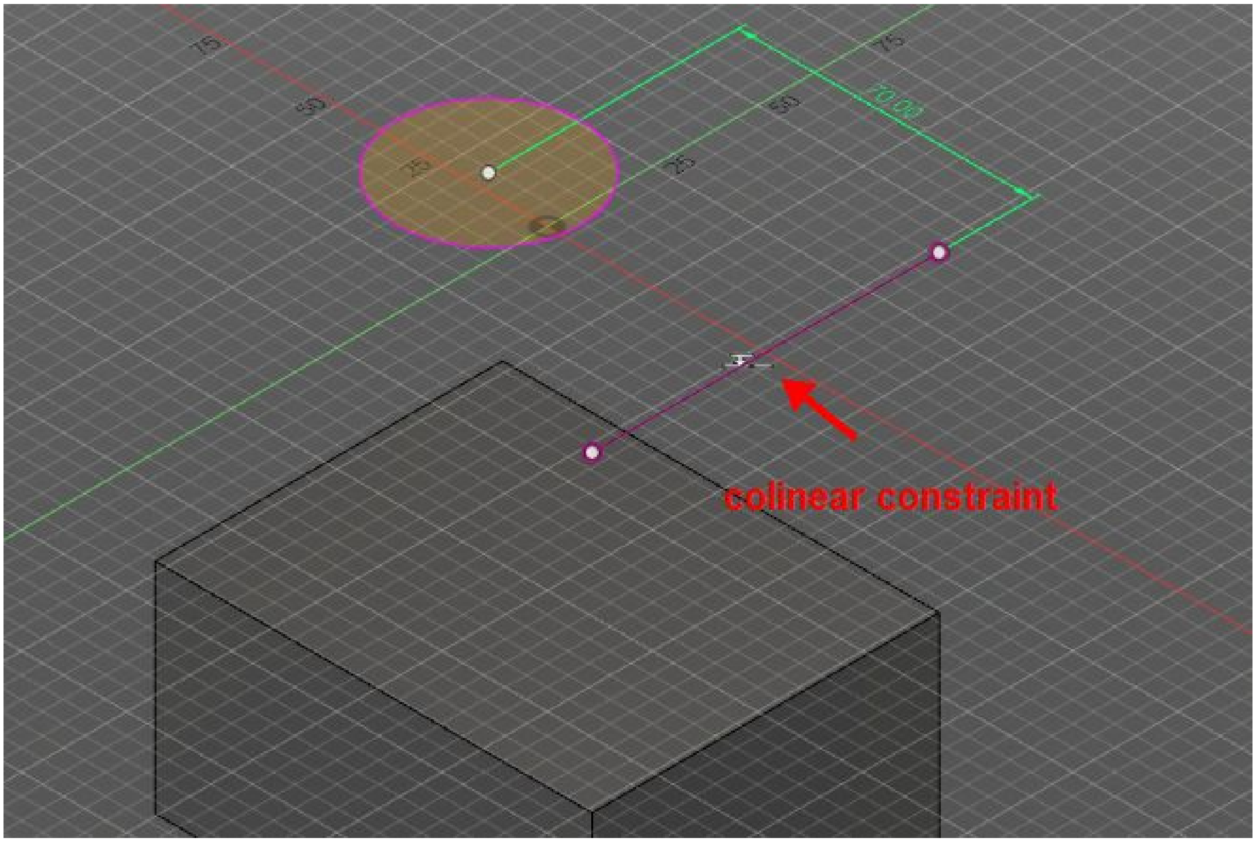
Convert to construction. This is not strictly necessary (and unwise if the projected curve is used in a profile). I did this here just to make it easier to apply the constraint later:



Re-Project the edge:



And, finally, make the two lines colinear:



And the system will update correctly

Future consideration: Everyone realizes this is ugly, as it is implemented today. We do have a project in the backlog to implement a “Repair Sketch Projection” command. This would simplify the workflow, we believe. This is the Idea Station item for it: provide-an-interface-for-fixing-broken-projections-misplaced. Unfortunately, this idea only has 6 votes currently… So, cast your votes!