

**ZW3D**  
***Advanced Modeling Training Guide***

***Copyright and Trademarks***

©Copyright 2017 ZWCAD Software Co., Ltd. All rights reserved.  
32/F Pearl River Tower, No.15 Zhujiang West Road,  
Tianhe District, Guangzhou 510623, P.R.China  
(8620)38289780

**ZW3D™ V2017 Advanced Modeling Training Guide**

This documentation may be reproduced provided it complies with the terms presented on the LICENSE AGREEMENT supplied.

ZWCAD Software Co., Ltd and the program authors have no liability to the purchaser or any other entity, with respect to any liability, loss, or damage caused, directly or indirectly by this software and training materials, including but not limited to, any interruptions of service, loss of business, anticipatory profits, or consequential damages resulting from the use of or operation of this software.

Updates may be made to this documentation and incorporated into later editions.

ZW3D™ is a registering trademark of ZWCAD Software Co., Ltd.

The ZW3D™ logo is a registering trademark of ZWCAD Software Co., Ltd.

ZWCAD™, ZWSOFT™, the ZWCAD™ logo, and the ZWSOFT™ logo are all trademarks of ZWCAD Software Co., Ltd.

Printed in the P. R. China.

## Forward

The full version of the ZW3D documentation is available by selecting **Help Browser** from the ZW3D **Help Menu**. Go to the "**Program Folder\doc**" directory on the ZW3D program directory and open the file "**ZW3D.chm**".

This **Advanced Modeling Training Guide** is available as a printed manual and in PDF format. To view or print the PDF version, install the Adobe Acrobat Reader and open the corresponding files in the "**Program Folder\PDF**" folder in the program directory. Refer to the **HELP Manual** for information relating to specific tasks.

### NOTES:

1. For best results, view and print the PDF version of the documentation using the Adobe Acrobat Reader.
2. Since the **Advanced Modeling Training Guide** is only a portion of the complete documentation, it is highly recommended that you utilize the **HELP Manual**. You will find additional topics, more information on command options, input options, additional command notes, and cross-links to related tasks and topics.
3. It should be noted that the HELP Manual will always be more up-to-date than any printed or PDF material. If you find any discrepancies between the two, always refer to the HELP Manual.
4. While you are working with ZW3D, simply press **F1** to view the documentation about the command you select.

To help you customize your CAD/CAM requirements, please contact our Customer Support department on our website <http://www.zwsoft.com/> or call at (8620)38289780 ext 565. You can also contact your local sales representative or application engineer to discuss your requirements.

Thanks for being our customer!  
Team ZW3D

# Table of Contents

<b>Chapter 1</b>	<b>Hybrid Modeling</b>	<b>5</b>
	What is hybrid solid/surface modeling and what are the benefits?	6
	What applications is hybrid solid/surface modeling best suited for?	6
	Will your organization benefit from Hybrid modeling?	7
	Hybrid modeling example 1	8
	Hybrid modeling example 2	12
	Hybrid modeling example 3	22
	Hybrid Modeling example 4	31
<b>Chapter 2</b>	<b>Face Editing</b>	<b>43</b>
	Splitting shapes to curves or other shapes	59
	Curve List and Lofts with Continuity	70
	Classic example of editing the loft connection lines	79
<b>Chapter 3</b>	<b>Sweeps</b>	<b>83</b>
	Command option examples 1.....Z-Spine	86
	Command option examples 2..... X-Orient	89
	Variable Sweep	92
	Variational Sweep	99
<b>Chapter 4</b>	<b>Shell</b>	<b>113</b>
	Shell	114
	Volume Offset	117
	Face Offset	118
<b>Chapter 5</b>	<b>Family of Parts</b>	<b>123</b>
	Part Attributes Form and creating families of parts.	124
	Table-based family-of-parts	129
<b>Chapter 6</b>	<b>Morph</b>	<b>132</b>
<b>Chapter 7</b>	<b>Additional Exercises</b>	<b>138</b>
	Designing a Button	140
	Designing a Clip	153
	Designing a Bottle	166
<b>Chapter 8</b>	<b>Design Optimizer</b>	<b>178</b>

Design Optimizer (3D) .....	179
Design Optimizer exercise:.....	183
Chapter 9 Point Cloud .....	190

# **Chapter 1    Hybrid Modeling**

### ***What is hybrid solid/surface modeling and what are the benefits?***

Almost all Modelers today use a boundary representation to represent solids. In a nutshell, this means that the object is represented by the surfaces that define the outer boundary of the volume. Thus, the axiomatic principle that is key to an understanding of solid versus surface modeling is that given enough time and the proper set of tools, any object that can be modeled with solids can be also modeled with surfaces. The converse however is not true. In order to provide a reasonable set of operations, solid Modelers typically restrict the user to a finite set of shapes (i.e. spheres, boxes, etc.) and/or procedures (i.e. extrude, revolve, sweep, fillet etc.) that can be used to define the geometry of the object. This inherently limits the range of objects that can be described. In addition, they do not allow the representation of zero thickness partitions, such as the parting surface of a Mold. Because solid Modelers require that a true bounded volume be described at all times, manipulation and "fine-tuning" of the individual bounding faces is generally restricted.

Why then, with these limitations, has the use of solid modeling grown so dramatically? The primary reason is productivity. Since solid Modelers can perform in a single step what would generally take numerous operations with surfaces, most objects can be described much more efficiently. Also, these solid operations can generally be accompanied by describing dimensions and constraints in a fashion that allow them to be modified quickly and easily.

So we see that there are advantages to both methodologies that can be summarized as speed (solids) versus flexibility (surfaces). It would seem then that the ideal Modeler would provide both techniques, in an environment that allows the user to easily switch back and forth between them. This is known as hybrid solid/surface modeling. In practice, the synergy of this combination has proven to exceed the sum of its parts. A well-designed hybrid modeling system allows the user to leverage the power and efficiency of solids whenever possible, yet never force a design to be compromised for lack of an appropriate tool to achieve the desired geometry.

### ***What applications is hybrid solid/surface modeling best suited for?***

The ability to have a single CAD/CAM system that can be fast enough to design a gearbox on Monday and flexible enough to design a body panel on Tuesday is perhaps a compelling enough reason to have integrated solid and surface design. However, to truly realize the benefits of integrated solid/surface modeling one must look beyond this either/or paradigm. Consider, for example, the design of a camera or cellular phone. A true hybrid modeling system provides the functionality to design a beautiful, ergonomic body for the appliance with surface modeling tools and in the same environment apply a powerful and intelligent set of solid modeling tools to convert it into a thin shelled case, split it in half and add all the required bosses, ribs and flanges. If requirements dictate a geometric feature that cannot be achieved with one of the standard solid feature tools, in a hybrid modeling system surfaces can be used to "knife and fork" the desired shape. Beyond the design of products that don't easily fit the standard feature-modeling paradigm, there are several other applications where the integration of solids and

surfaces yields great benefits. One of these is for the repair or modification of models that have geometry problems, most often as a result of data translation or Modelers that allow geometry inaccuracies to accumulate.

Another is for the design of Molds & Dies, tools or fixtures where a shape needs to be partitioned into pieces along irregular boundaries, or faces of an object need to be extended or manipulated. Quite often supplemental geometry needs to be created for the purposes of NC tool path creation. A CAM system with an integrated hybrid Modeler is a tremendously powerful tool.

***Will your organization benefit from Hybrid modeling?***

Does your company do NC programming?

If yes, you need hybrid modeling.

If no, continue.

Does your company work with imported solid or surface data via IGES, STEP or VDA?

If yes, you need hybrid modeling.

If no, continue.

Does your company design parts that have free-form shapes? Any feature of the part should be considered and not just the overall shape. For example, your design is 98% prismatic in shape but has one feature such as a cam that requires a free form or variably swept face.

If yes, you need hybrid modeling.

If no, continue.

Does your company design Molds, Dies, tools, or fixtures?

If yes, you need hybrid modeling.

If no, continue.

Does your company use a solid Modeler that is limited in the functions available to create your designs? For example, do your parts have complex fillets or corner blends that your "state-of-the-art" solid Modeler is not capable of creating?

If yes, you need hybrid modeling.

If no, continue.

Are your designs primarily surface modeling based but you would like to realize the benefits of parametric, dimension driven modeling?

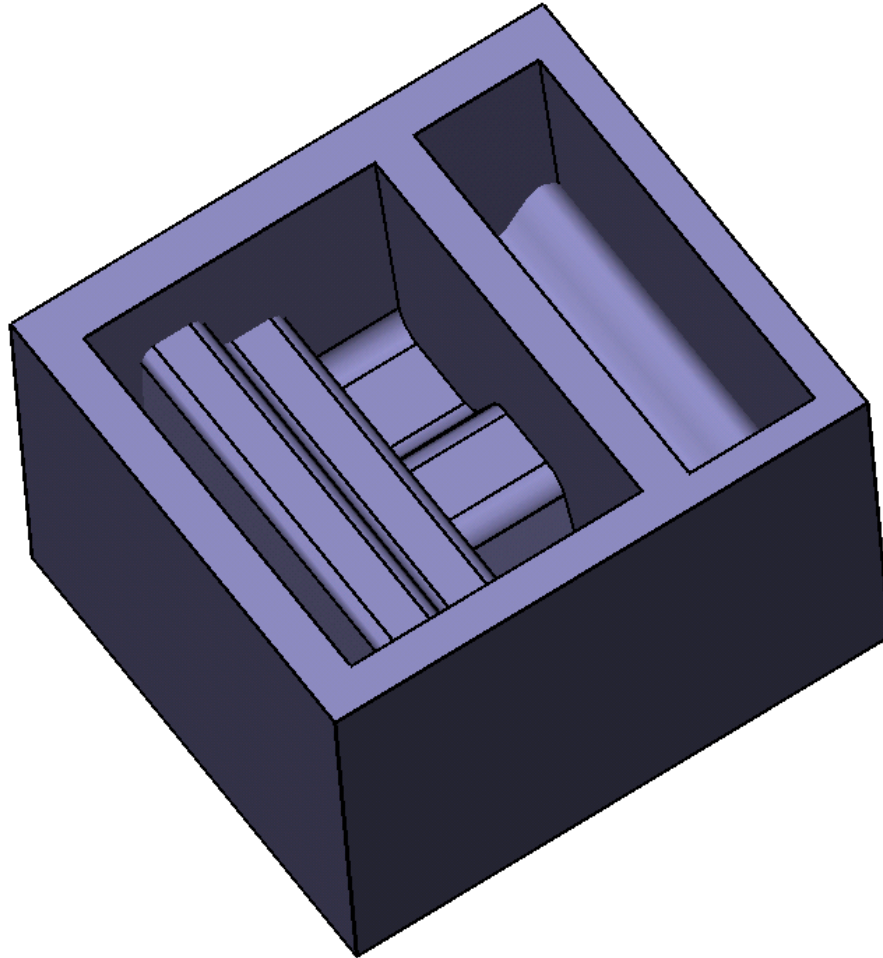
If yes, you need hybrid modeling.



### *Hybrid modeling example 1*

Open the file “**Advanced\_Modeling**” which is located on the training CD.

Edit the part “**01\_Hybrid1**”.



Step through the part history to see how the model was constructed.

Notice how the Bottom Face has been deliberately removed in order to make this a surface model or open shape.

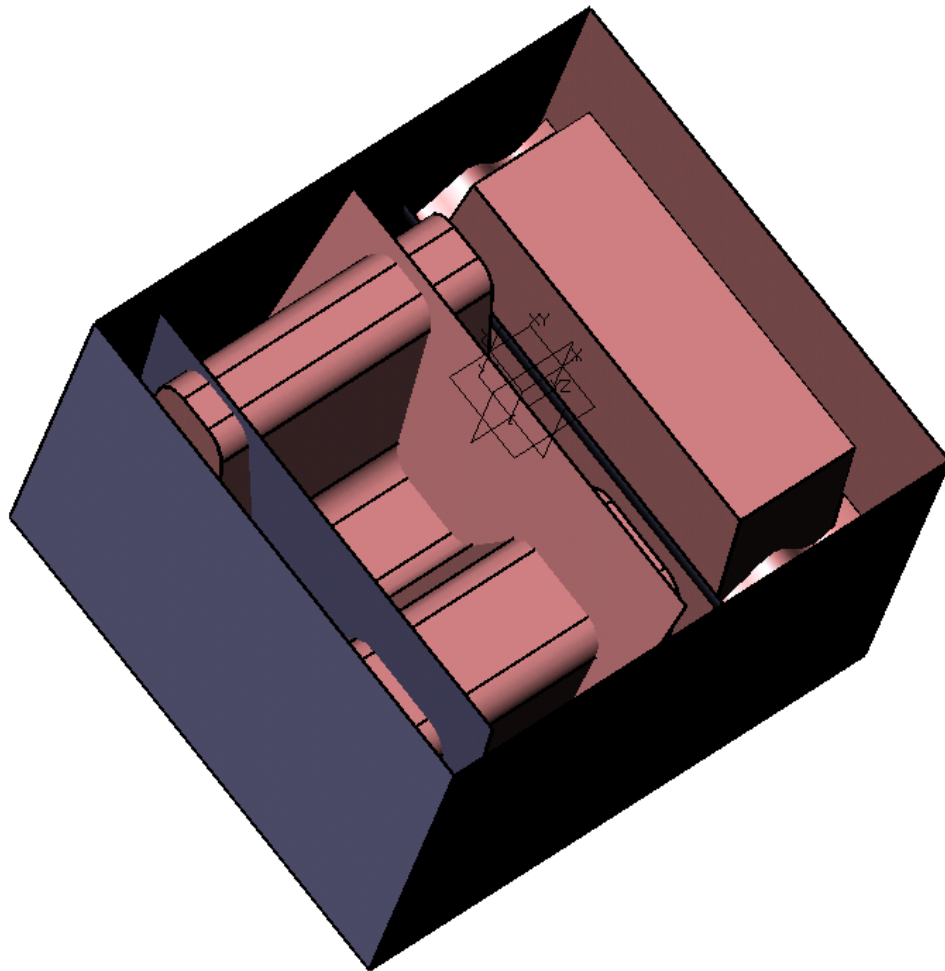
In spite of this you can see that the extruded cut operations work even though the part is not a solid.

In wire frame mode, some of the edges of the faces are shown as blue and dotted to indicate that they are not attached to other face edges.



In shaded mode ZW3D will show you the inside (or negative normal) faces clearly in a pink color. When creating faces that share common edges, the normals will be oriented in the same direction automatically. If the faces eventually form an enclosed volume (i.e. a solid) then all the faces will be orientated so that the positive normal points away from the solid body.

Rotate the view so that you are looking underneath the part as shown below.

Note: open the **face** attribute option under **Attributes** pull-down menu, then set the color of the front and back from the pop-up dialog box.



You can clearly see that the extruded faces and the main body need to be trimmed to each other. Although there are many powerful trimming options in ZW3D that allow the user to trim the faces, it is nevertheless a time consuming process. Instead we will use hybrid modeling techniques to quickly join the Shapes to each other.

From the **Shape Tab**, using the **Combine**  > **Add** command , join Pick the main body as the '**base shape**'. Middle-click to advance the menu.

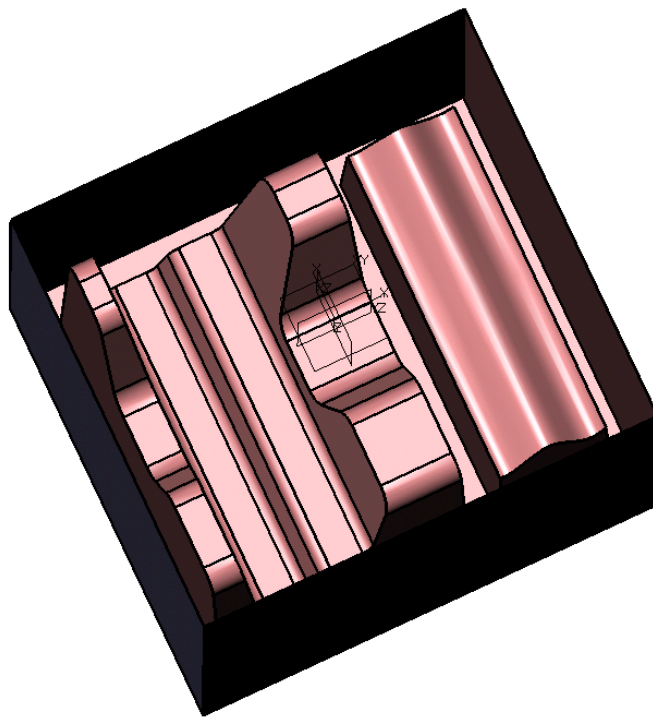
Pick all the other faces (with a window pick) as the '**operator**'. Middle-click to finish your selection.

Middle-click to end the command or Pick OK on the menu.

All the faces are now trimmed correctly using what is traditionally a solid type command.

The part should now be as shown below.

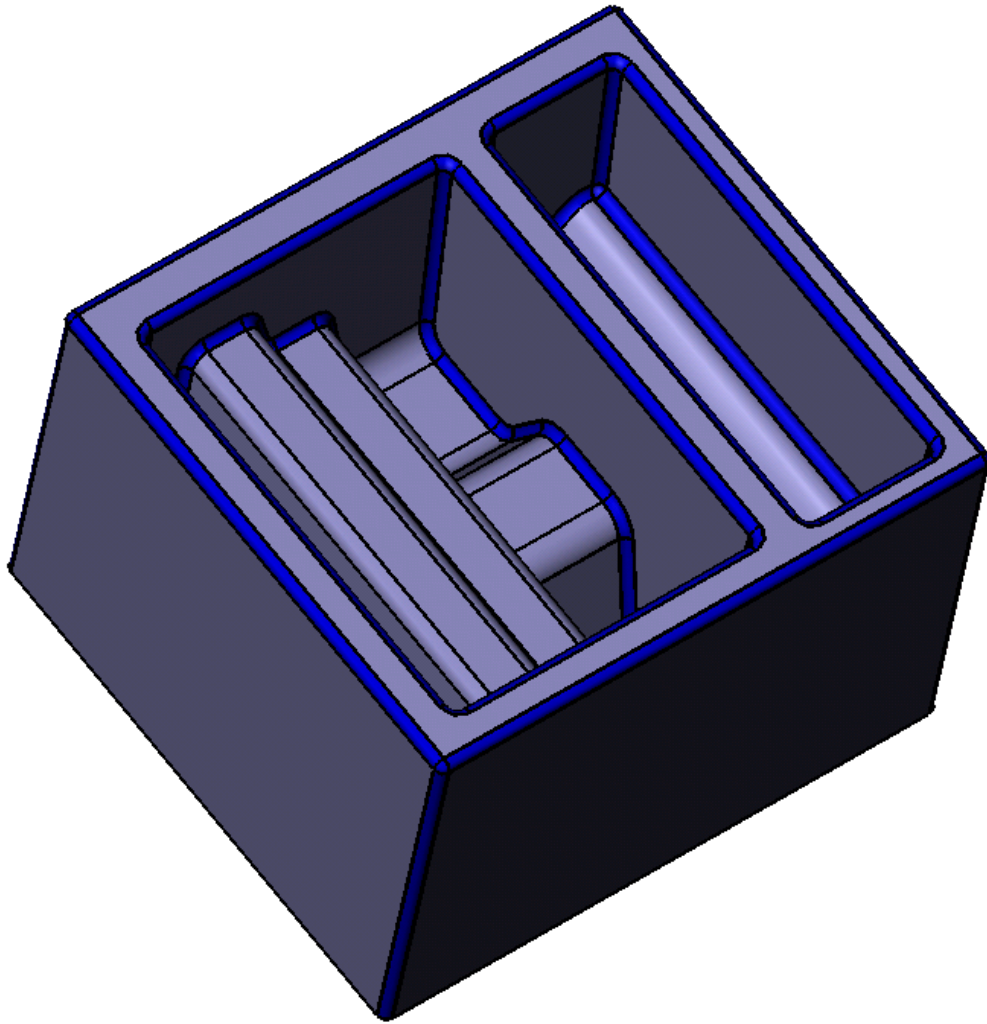
Next change the default color attributes for new faces to **blue** using Face color button



With ZW3D hybrid modeling the user can use commands, normally associated with Solid Modelers, on surfaces and or solids.

For example.


Using the **fillet** command  let's put a **5mm** fillet on all of the edges (window pick the entire part).

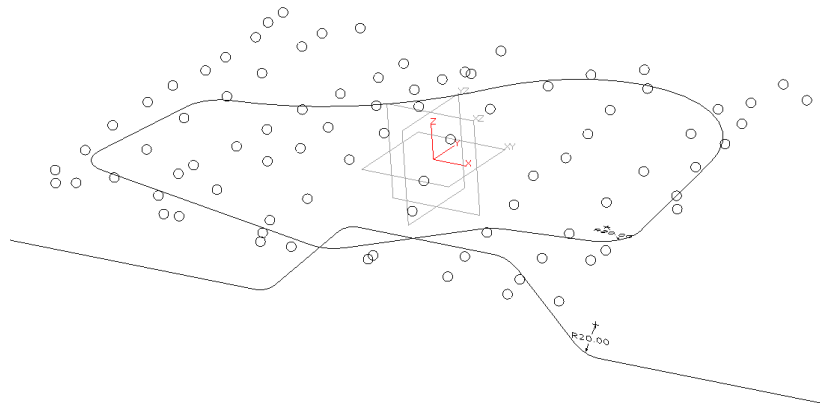


Remember all of the commands shown in this part were done on a surface model.

No need to save the file.

## Hybrid modeling example 2

Exit part  and edit the part “02\_Hybrid2”.



The part consists of a number of random points with various X, Y and Z values, and two sketches. **Sketch1** is on the XY-plane and the **Sketch2** is on the XZ-plane.

Create a surface through all of the points using the:

**Fit Face** command. 

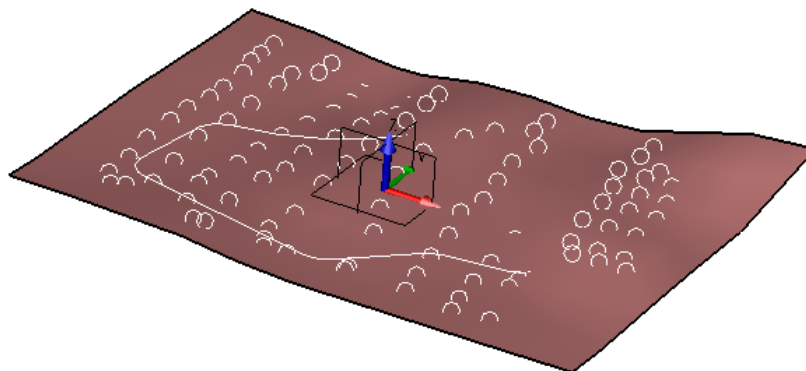
Pick all the points.

Set **Bending Resistance** to 6.7.

Pick **OK**.

(Raising the bending resistance will make a flatter surface.)

Then **erase** the points; don't forget to set the **pick filter** to “**Points**”.



Create a pocket in the face.

Using the **Extruded**  > **Remove**  command.

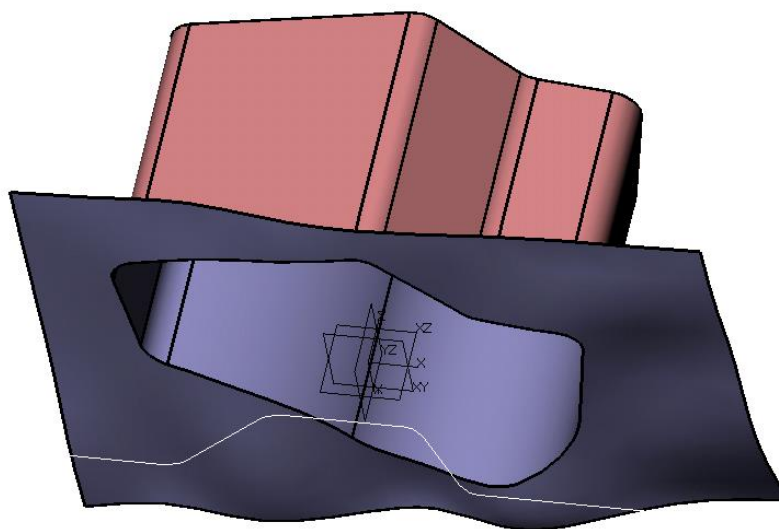
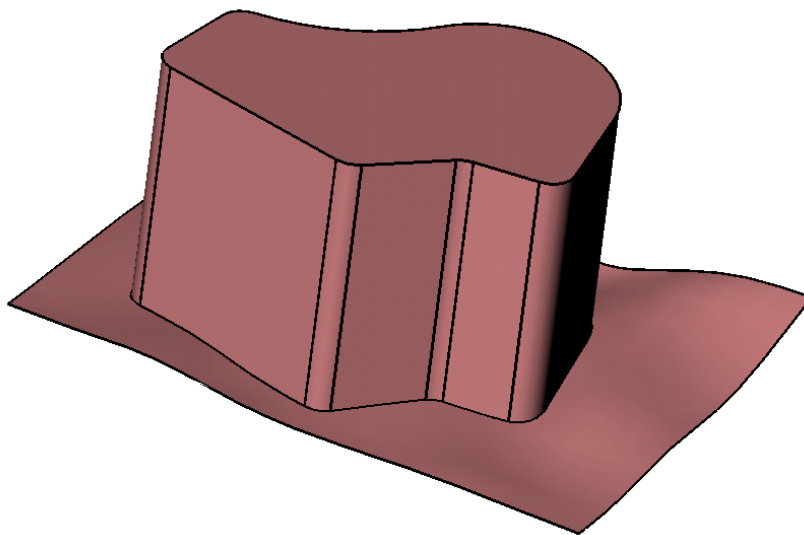
Pick **sketch1** on the **XY-Plane**.

Start at **+200**.

End at **-50**.

Add an **endcap** at the start point.

Pick **OK**.



## Create Constant radius fillets.

Using the **Fillet** command. 

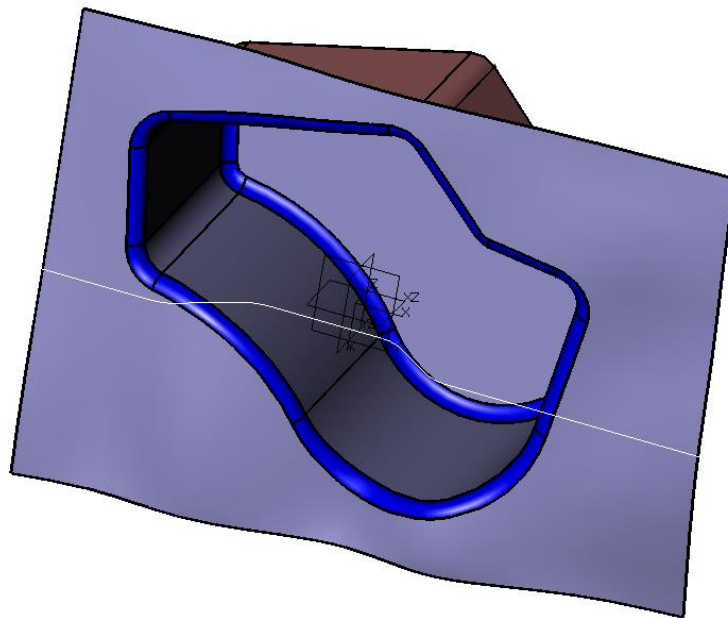
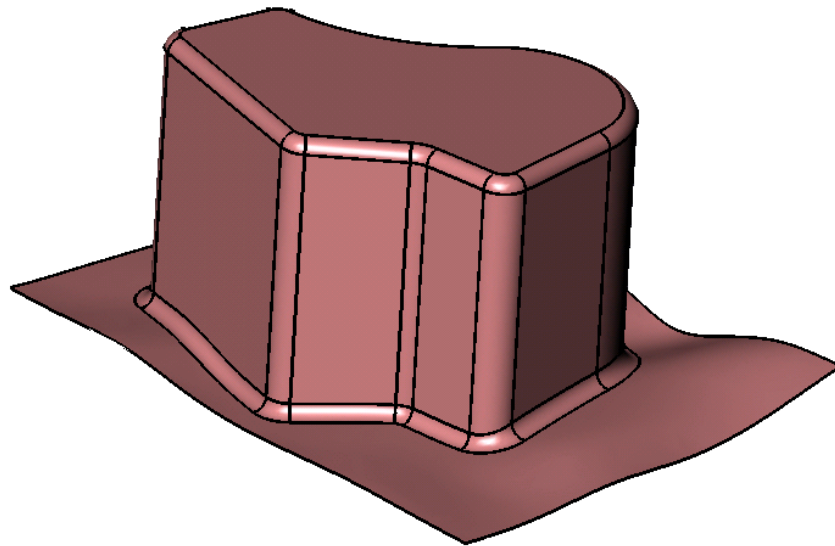
Chain select the edges on the flat bottom face of the part.

(Hold the <Shift> key and pick one edge)

Chain select the edges at the intersection with the curved face.


Create a **10mm** fillet as shown.

Middle-click to finish the command or Pick **OK**.





## Create four extruded faces

Pick the **Face color** button and select a new face color of you want. (Ex: )

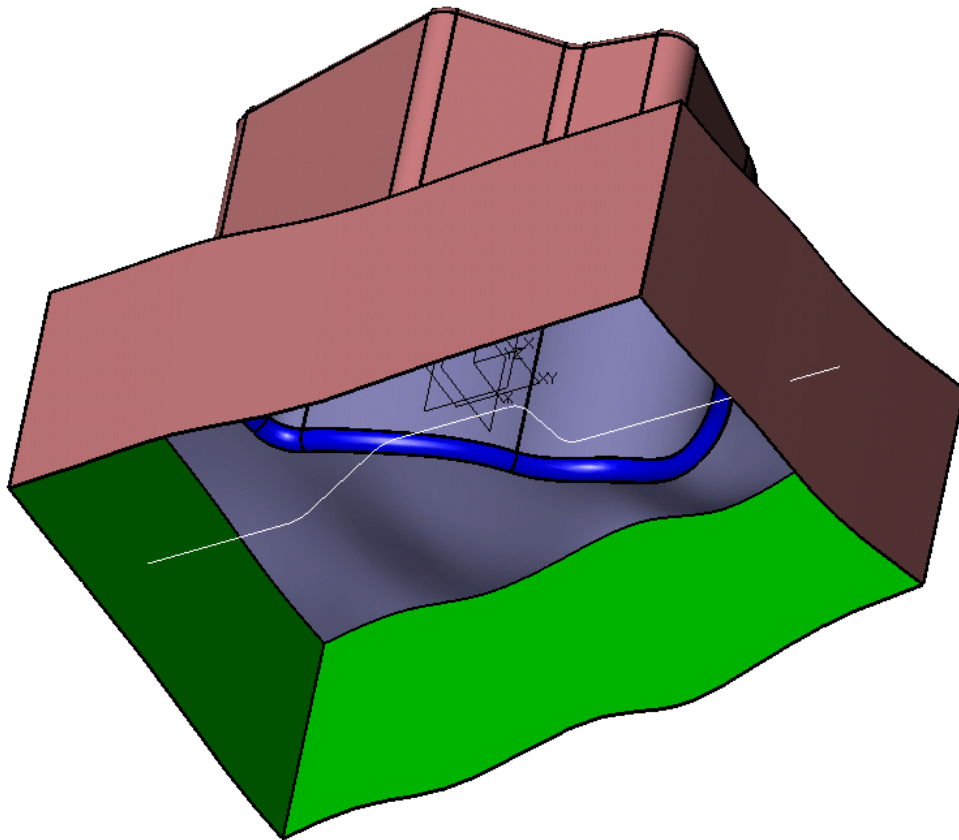
Select the **Extruded**  **> Add**  command .

With the right mouse button choose Insert Curve List and pick all four edges of the curved face.

Start value **0**.

End value **-170**.

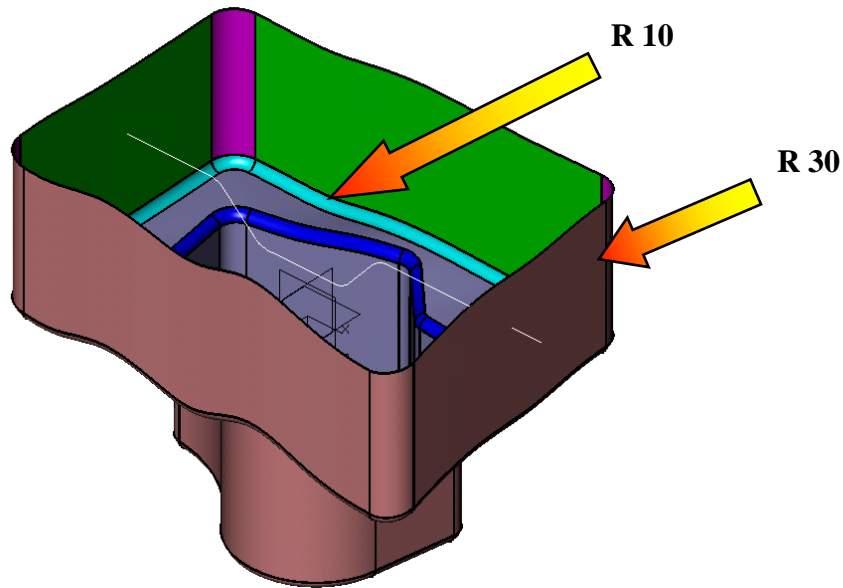
Pick **OK**.





Create two constant radius fillets. 

**30mm** fillets on the four vertical edges of the newly created **extruded** faces, then...  
**10mm** fillets at the intersection between the extruded faces and the **curved** face.



Shell the part **-5mm**. No openings.

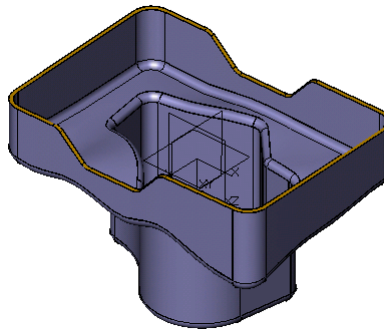
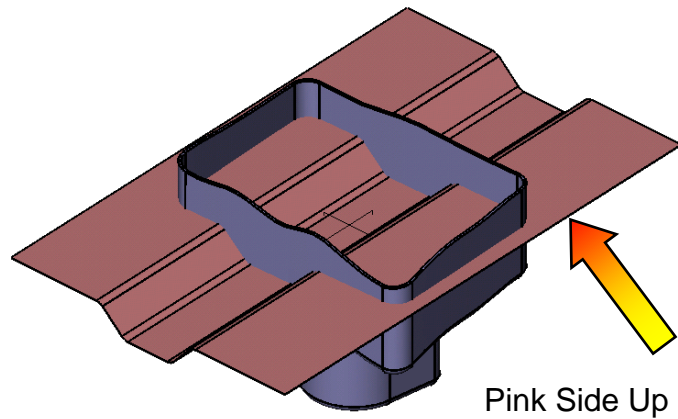
Create a Base Extrude for trimming 

Pick the **Sketch2**.

For the extrude direction, select the **Y-axis** option (**right mouse button**).

Start at **+500**.

End at **-500**.



Desired results

We are going to use this surface to trim off the top portion of the solid shape.

To rationalize which operation will give you the desired results, think of the open surface as a solid extruded in the direction of the red face. See below. Remember, red is the default color for the inside of a solid or back side of a surface.


In the image, the solid is extruded up and envelops the part of the shape we want removed.

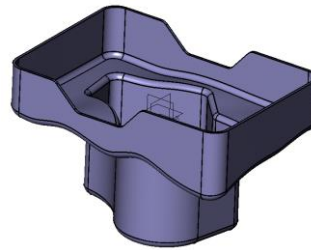
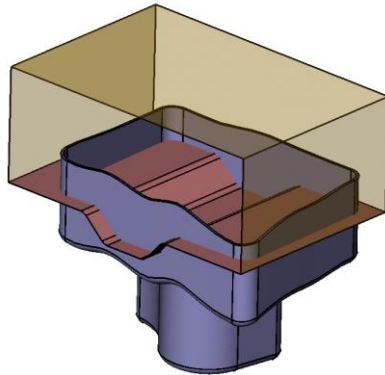
So **Combine > Remove** would give us the desired result.

Try **Combine**  > **Remove** .

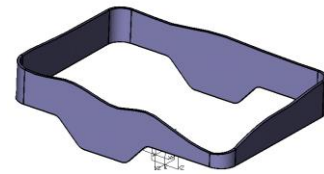
Select the main body as the **base shape**.

Select the extruded shape as the **operator**.

Now **Undo** it, we are going to try something a little different. 



Remove



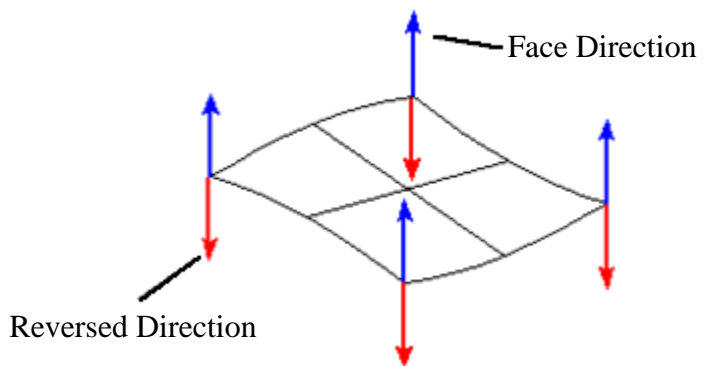
Intersect

**Reverse Face Direction** command from the **Free Form** Tab.



Use '**Shape**' Pick-Filter.

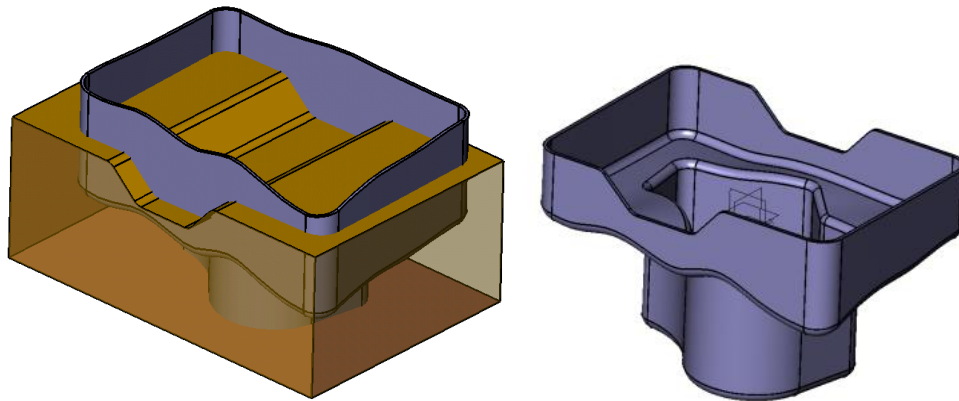
Use this command to reverse the normal direction of a face or shape. The user can decide on whether to reverse selected faces or the entire shape by selecting the relevant option in the pick filter.



Create the wall top using **Combine**  **> Intersect**  .

We will create the wall top shape again, but now with the blue side facing up, we will use the **Combine > Intersect** command.

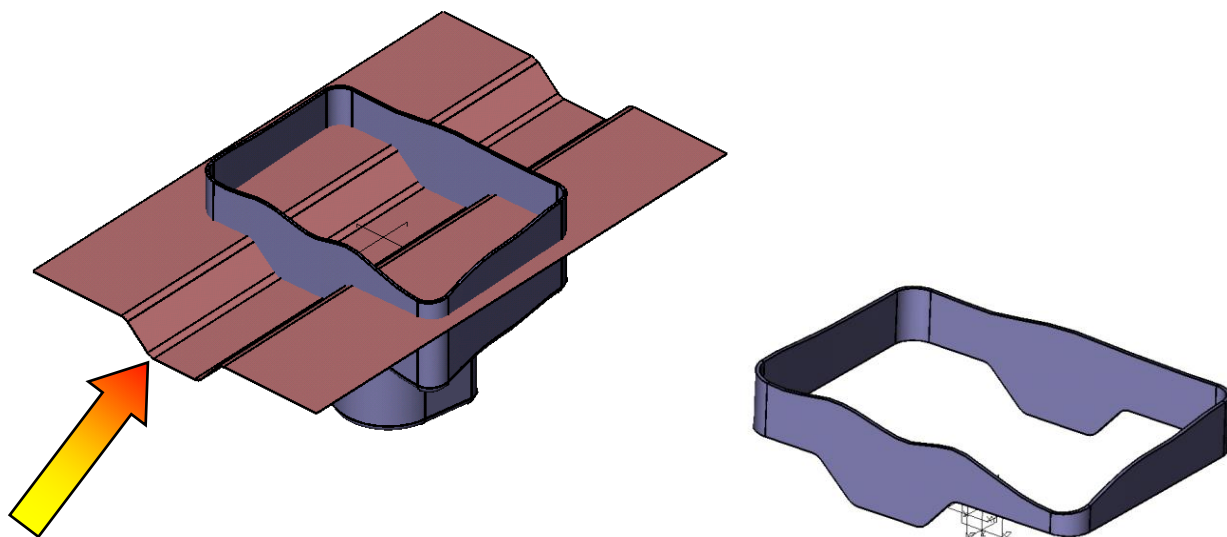
So you can predict the result when you are on your own, observe that the inside (or pink side) of the surface is facing down so a solid would envelop the lower part of our shape. Since this is the side we want, you would use the intersect option.



The image on the screen should look like this

Test your understanding of what we just did.

What command gave the results in the image below on the right?



Pink side up

Answer: Combine > Intersect

### Hybrid modeling, and non-native data.

But first let's review the iges input and tolerance setting, as described below.

**Auto sew geometry.** Closed surface volumes will be sewed (joined) at their mating edges to form a solid once they have been imported into ZW3D. This option requires

that the imported edges fall within tolerance of each other. (i.e., no free edges with gaps greater than the geometry tolerance).

If importing directly into an existing object or one that is currently active, the default tolerance displayed in the **part setting form** is used.

If importing into a new object (created during the import), the default system tolerance (**0.01mm**) is used.

**Auto activate part.** The imported file will be automatically activated for editing (i.e., it will become the active part). The name of the part will be taken from the name of the iges file.

If a part is already active for editing (open) the iges file will be merged into that part.

**Rename layer collisions** If the IGES file is being merged into the active part or drawing sheet and it contains references to layer names that already exist, the IGES file layer name references are renamed. If this option is not checked, then the contents of the layers are merged with the existing layers.

**Remove duplicate surfaces** If there are duplicate trimmed surfaces, the duplicate is thrown away if this option is selected; otherwise, the duplicates go to a layer named 'DUPLICAT'.

**Break trim edges.** Trim loops will be evaluated for tangent discontinuities and broken if required.

**Reset dependencies.** It is not uncommon for third party IGES files to contain incorrect dependency flags causing entities not to import, or to import more than once. Use this option to flush the dependencies reported in the file, and recreate them based on the entity references from the parameter section of the IGES file.

**Drawings.** Embedded drawings will be in 3D space.

**Bounded planes.** Bounded (108) planes will be imported.

**Unbounded planes** Unbounded (108) planes will be imported.

**Part Settings** form is obtained from EDIT > Preferences.

**Tolerance.** This setting refers to the default part geometry tolerance measured in the default units. The geometry tolerance is feature dependent. Every new feature is assigned a geometry tolerance as defined in the **Part Settings Form** at the time the feature is created. If the geometry tolerance is modified, only newly created features are affected. Earlier features retain the geometry tolerance in affect at the time the feature was created.

You can adjust the geometry tolerance for individual features using the **Edit > History > Modify Tolerance** command.

**CAUTION:** Modifying the geometry tolerance may cause a history regen to fail. It is recommended that you use the Regenerate Current Object command after modifying the geometry tolerance. If the change in tolerance causes the regen to fail, you can elect not to keep the change. **A regen failure later may be more difficult to locate and correct.**

Try to settle on a tolerance value that is right for the size and complexity of the part before you begin the design. A tolerance value that is too small will cause increased processing times and increase the complexity and size of the model's database. A value that is too large may cause the system to ignore features that are very small relative to larger features and decrease the accuracy of intersections and curve/face approximations.

### **Handling Non-Native Data and Hybrid Modeling.**

Generally speaking if the users Cad system is solids based only, it is fair to say that any meaningful work on the model can only be carried out if the geometric entities form an enclosed volume (i.e. a solid). This may seem obvious enough, and in fact when creating geometry in a solid Modeler the geometry is clearly already a solid, so where's the problem?


We have stated previously our opinion on the advantages of hybrid (as opposed to just surface or solid) modeling. But the advantages are never more obvious when the user has to import data from other cad systems. The data, for a variety of reasons (it may not have been a solid in the first place for example) often comes into the solid Modeler with various problems, gaps, overlaps, bad trims etc. The solid Modeler user then has to spend a considerable amount of time trying various methods of editing, if at all possible, or replacing the geometry until the part eventually becomes a solid, and he can at last do some work. Often this takes an unbelievable amount of time to achieve, and is even sometimes impossible, due to perhaps the complexity of the geometry, bad data, and or, transfer, or simply the lack of the proper tools. The Surface Modeler has a clear advantage here, as he does not need a solid. But as we have seen, working with Surface Modelers, while they are certainly flexible, are also relatively slow. It could also be, for various reasons, that a solid is required. ZW3D hybrid capability has an obvious and clear advantage here for two reasons.

We do not have to work with a solid but are not hampered by the lack of speed traditionally associated with Surface Modelers. If a solid is required, there are many powerful tools within ZW3D that can accomplish this task.

The following exercises use a ZW3D IGES file that has deliberately been constructed with poor edge data. We will see how we can work with this data, in spite of the fact that

is not a solid. Then we will repair the entities using a variety of techniques such as surface joining, topology checking and healing, etc.

### *Hybrid modeling example 3*

**Exit part**  and create a new part object “03\_Hybrid3\_IGES\_Import”.

Once inside the part, click this triangle to expand the menus.



Select **Edit > Preferences** pull down menu form. The default tolerance should be **0.01mm**, if it is not, change it now.

If you first create a part and then edit the IGES file, you have the opportunity of changing the tolerance. If you do not first create a part and instead rely on the auto activate option, or drag & drop, the part will be read in with the current session default tolerance and at the current units.


Go to **File > Import**.

Select the file type **IGES File**

Select the IGES file Hybrid3.igs which is located on the “\training” directory and open it.

Enter the “**IGES File Input**” form. We need to make one change to the list of the default settings, **Uncheck: Auto sew Geometry** and hit the **OK** button to open file.

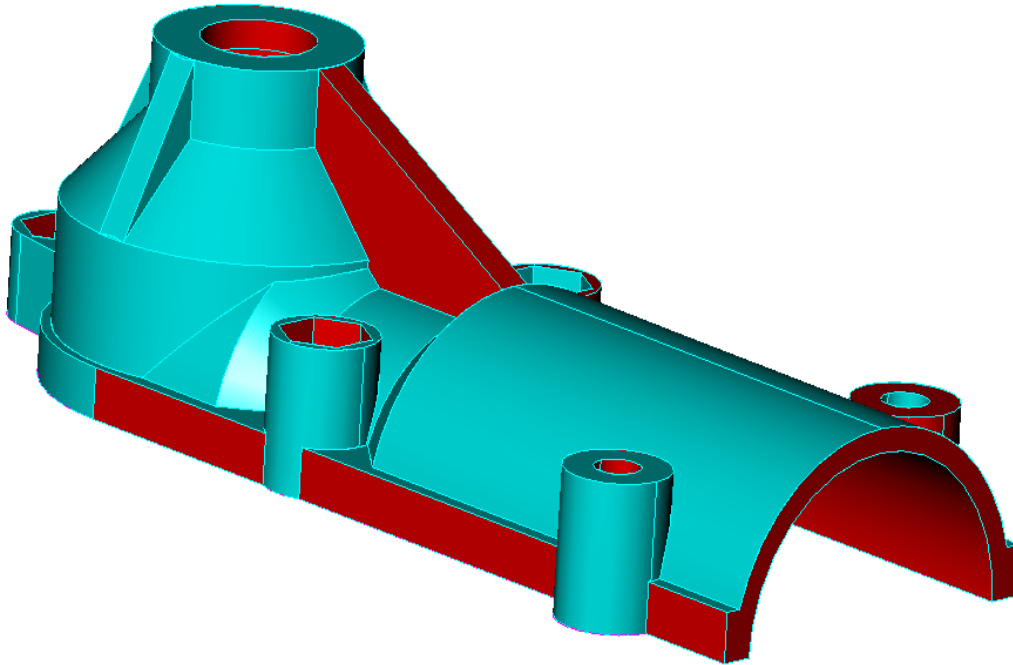
ZW3D has finished the import when the Translator Progress form reads “**Translation Complete**”.

From a **View > Standard Views > Front**, pick the **Erase** icon  and **&ltRight-click>** Window pick the entire lower Shape.

**&ltMiddle-click>** to finish.

If the display is set to wire frame, switch to shaded mode now.

The part should now be as shown below.



Notice that some of the faces are red colored. This indicates that the negative directions of the face-normals are pointing outwards. If the part were a solid the positive face-normals would instead be pointing outward. Shading the faces a different color indicates this. The color that the positive side of faces will be shaded is the default face color. If you wish to differentiate between the outside and inside of parts, then you should choose a “Face” color different from the “Back Face” color. The default back facing color is pink as we saw in the first exercise. An imported part takes on the colors at the time of exporting.

Note: open the **face** attribute option under **Attributes** pull-down menu, then set the color of the front and back from the pop-up dialog box.

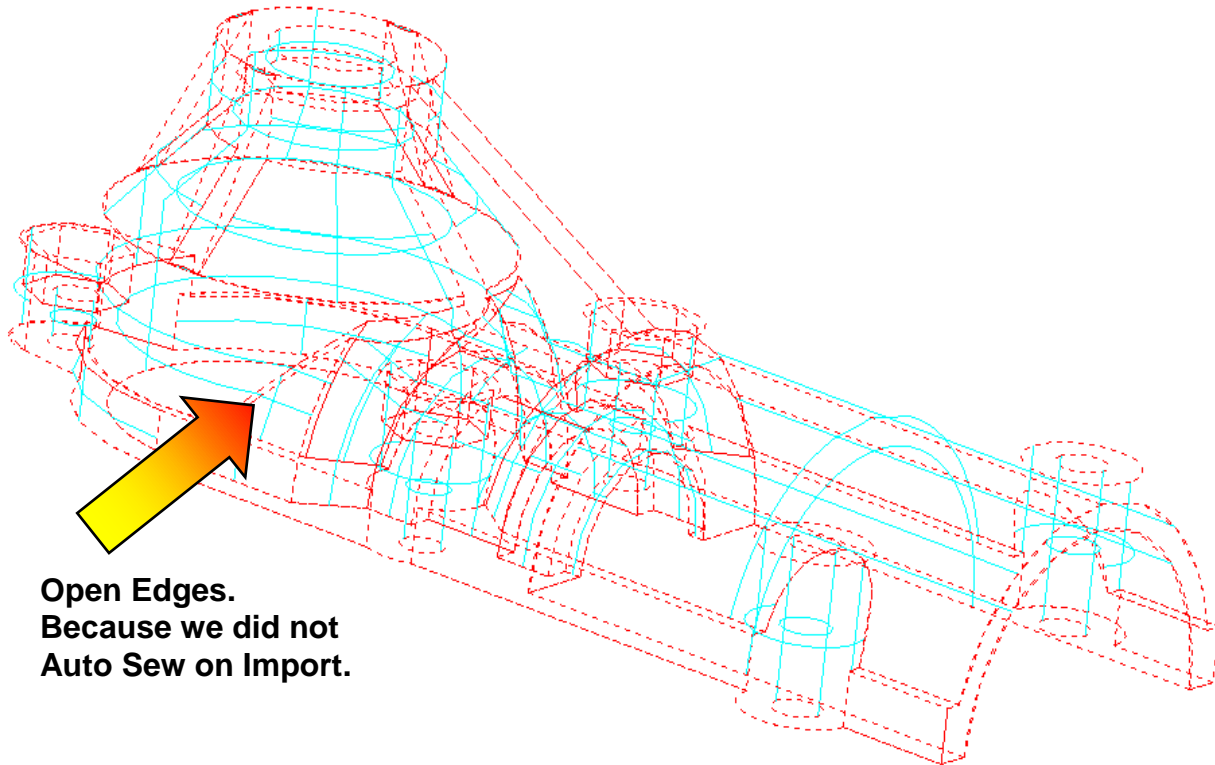
In our example we now know that the part is not a solid. WHY? Next, we will discuss a number of ways in which to establish whether or not the part is a solid.

Change the display to **View > Wireframe**.

Notice all of the edges appear **Blue and Dashed**. These are the open edges.


Note: The color of open edge can be set from the “Visual manager”





Dashed lines are **OPEN** edges in the surface model.

Another method of determining whether or not the part is a solid is by setting the Free Edges option to show. Select the **Edit > Preferences**, then pop up a “**Part Settings**” form, and is set to “**Alternate display of free edges**” by default. If checked, all free edges (openings in a shape) larger than the geometry tolerance are automatically displayed with specific attributes.

Pick the **Sew** command  from the Heal Tab , **<Middle-click>** .

Middle click for Faces to select all and middle click again to use the default tolerance (0.01). the **maximum error** (gap) is shown in the work area.


The sewing result should be “ Number of unmatched edges is [38]. Maximum gap between unmatched edges is [0.031700 mm].”

Repeat the **Sew** command. This time change the tolerance to **0.04mm**. Why 0.04? You might be wondering. We determined this value based on the maximum gap message displayed during the first sew command we performed. We rounded the error value up.

**0.0317**, the exact value will work also.

Note: The number of unmatched edges should become **[0]** and all of the dashed edges become solid.

In this next step we are going to verify the model is recognized as a solid model by ZW3D. With the **pick filter** set to **Shape**, highlight the model and **<Right-click>Entity Info**. The top line in the **Show entity information** form indicates our model is a **closed** Shape. We were successful. Your numbers may vary in this form. That's OK.

Congratulations, you have successfully imported and sewn your part into a solid. This part was easy to repair and make into a solid, but to demonstrate that we don't require this part to be a solid, let's **undo**  the last sew command and proceed with some modeling commands.

Reverse the direction of the face-normals using the **Reverse Face Direction** command .



Ensure the **pick filter** is set to shape. The part then changes color and the positive normals are now pointing outwards.

### **Determining whether or not the part is a solid**

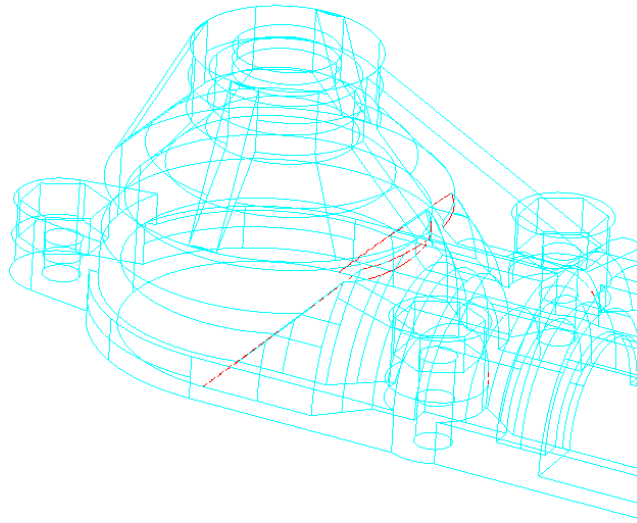
One method is to use the **Inquire** options.

Select **Inquire > Entity Info**.

Pick the part ensuring that the **pick filter** is set to **shape**.

The top line in the **Show entity information** form indicates our model is a **open** Shape.

Remember, open edges are indicated graphically by dashed edged that don't meet within tolerance.





The images shows clearly the problem areas on our part. Don't let the bottom edge of the part fool you. The shaded image shows the bottom face is a different color from the rest of the solid.

Sometimes the gap is so small it is not readily apparent where they are. In those cases use the command:

**Show open edges** from the HEAL Tab.



This menu indicates the number of open edges. You can click on the arrows   to cycle through all the edges for closer inspection. The length of the edge is given so you can determine if the edge is legitimate or possibly needs to be eliminated. You can also click on "Zoom To" in case the edge is so small you can't see the edges.

The part is still a surface model but we can nevertheless work on the part.

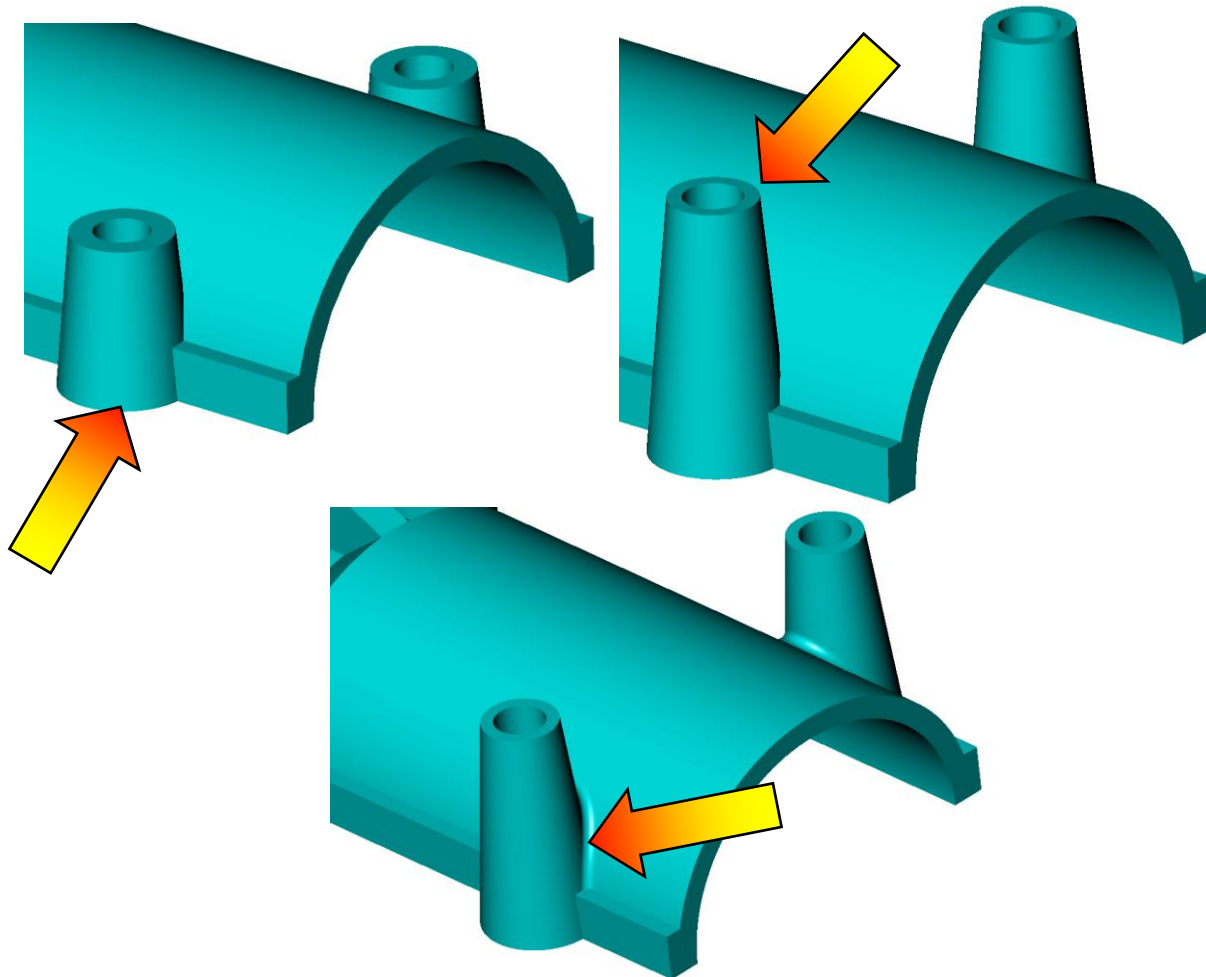
Let's prove this:

**Draft**  , **Face Offset**  , **Fillet**  .

Apply a draft angle of **5 degrees** to the two front bosses, picking the bottom **edge** of both cylinders as the entity to draft about. Refer to image below left. The reference plane is any plane perpendicular to the direction this part would be pulled from a mold or die. ( XY )

Next offset the top faces of the two bosses **+5mm** as shown below right. Note that the draft angle is maintained.

Finally apply a **0.5mm** constant radius fillet at the intersection of the two bosses and the cylindrical body. As shown in the bottom image.



Now let's **sew** the part at the **.04** tolerance and make this part a solid.

Next **Analyze**  from the HEAL Tab **Heal /analyze** command .

We have now joined our part together; all edges are now within our new tolerance of 0.04mm.

Let's check this.

Set the **Tolerance** option to **0.04mm**.

The Sampling Size option is explained in the HELP Manual.

The result is shown in **Report on part topology heal/ analyze** form that No error reported in vertices/ edges / loops.

Repeat the **Analyze** command with a point tolerance of **0.01mm**.

The result is shown in **Report on part topology heal/ analyze** form that there are gaps in edges.


The result of the analysis shows and identifies problem areas on the part geometry. The form together with the data on the screen informs the user of errors outside of the value given in the Point Tolerance option (in our example 0.01mm).

**Undo** the last command.

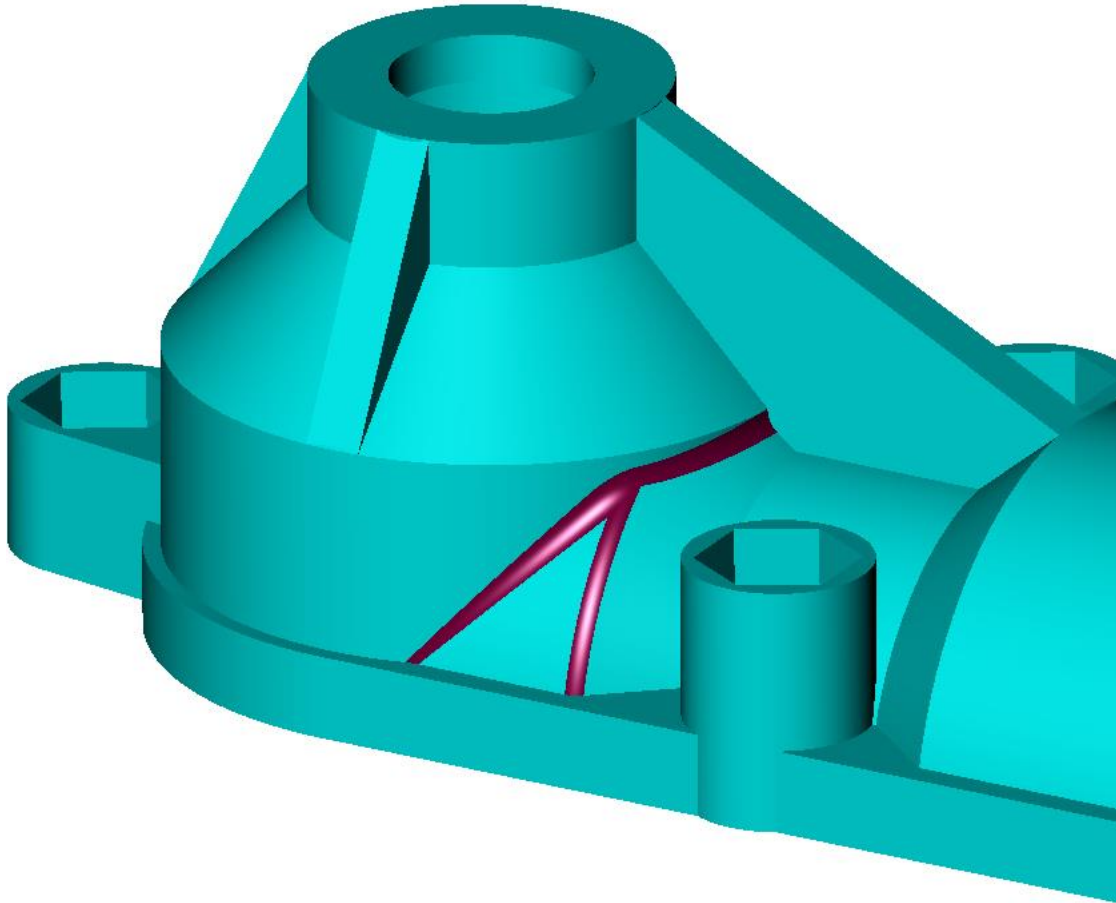
Select the **Heal** from the HEAL Tab **Heal /analyze** command.

**Heal** the geometry use a **Point tolerance** of **0.01mm**.  
Review the result with your instructor.

The gaps in the geometry have been eliminated and no errors are reported.


**Create fillets** on the edges shown .

The “bad” edges have now been repaired so let’s prove it by placing a 1mm fillet along the previous bad edges. Chain select the edges as below (press and hold the **shift** key, and select a adjacent edge), and place all the fillets in one shot as shown.



Did you notice how many edges were picked when you chain picked? This will happen frequently with imported files.

Undo your fillet command.

Use **Edit > Edge > Concatenate > OK**. Or use  icon from **Free Form** Tab (**Make this one of those commands that you do after every import.**)

Now add the fillets. There should be just 3 edges per side of the model.

## IGES Export Options Form

Use this form to map ZW3D geometry to IGES entity types.

**The options picked on this form have been found to work with most Importing systems.**

### *Geometry Form*

#### **Primitive**

Map to IGES primitive entities such as lines (110), circular arcs (100), surface of revolutions (120), and ruled surfaces (118), etc., when appropriate. Entities that cannot be mapped to primitives will be mapped to splines.

#### **Spline**

Map to IGES parametric spline curves and surfaces. These include rational B-spline curves and surfaces - Nurbs ( 112 & 114) and parametric spline curves and surfaces - Polynomials ( 126 & 128).

### *Solids*

The results of these selections will be governed by the **Geometry Form** and **Spline Type** selections.

#### **B-rep**

Map to IGES manifold solid B-rep objects (186) that contain vertices (502), edges (504), loops (508), faces (510), and shells (514).

#### **Free Surface**

Map surfaces independent of other surfaces without relationship information.

### *Other options*

#### **Group**

Pick the indicator to maintain group associations. Group associations include sketches, assembly components, and sheet geometry groups. Turn this option off for exporting to systems that have a known record of improperly importing IGES groups (308) and group instances (408).

#### **Extract Sub-Curves**

Pick the indicator to only extract sub-curves. Sub-curves include the arc portions of a circle or the trimmed portion of a curve. Turn this option on for exporting to systems that have a known record of improperly importing curve boundaries. Ideally, this option is left off so that the curve boundary (entire curve) and the trimmed portion are both exported.

#### **Split closed surfaces**

Pick the indicator to break closed surfaces into smaller non-closed surfaces to aid third party translators that have a difficult time with closed surfaces. This option is not supported for B-Rep solid export. Thus the ability to select 'Split closed surfaces' or 'B-rep' is disabled if the other is selected.

### **Extend surfaces**

Due to convex hull checking, other systems sometimes register ZW3D trim loops as falling off the surface in situations where the trim loop rides the surface edge. Use this option to extend the surface during the export.

### **Units**

Select the IGES export units from the list. All geometry will be scaled by the units selected.

Available units are Microns, Millimeters, Centimeters, Meters, Kilometers, Microinches, Mils, Inches, Feet, Miles.

## ***Hybrid Modeling example 4***

### **Non-Manifold Hybrid Modeling.**

In the next example we will see how solids and surfaces can be combined to create shut-offs in a Mold. Also how we can use assembly tools to merge in other parts, to be used in our current part, as “live” geometry, but still maintaining associativity with the original (parent) parts.

Edit the part “**Non\_man\_core**” .

With the right mouse button select **Insert Component**.

Check the **Anchor Component** option.

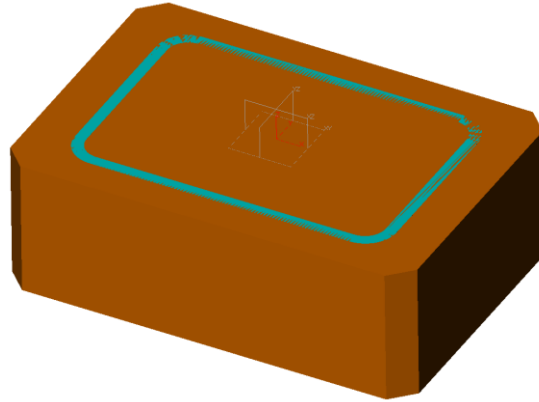
This eliminates having to align the component.

Pick the part **Non\_man\_part1**.

Type 0,0,0 for the location. Remember, you can type just one zero or type two commas.

The washed out effect is because the top face of both parts occupies the same space, and can be ignored.





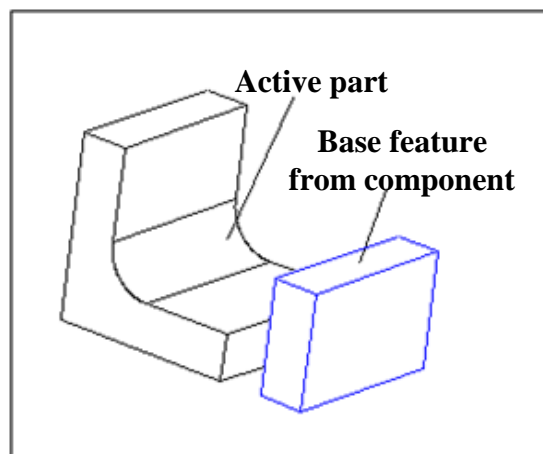
## Merging components into an active part.

Eventually we need to remove the component from the original part. At the moment this would be impossible as the component is an instance of the original part and is not usable as “live” geometry.

We can convert the component into live geometry in a number of different ways. Let’s review three options.

**Component merge** command from Shape Tab. 

**Component Merge (Base)** 

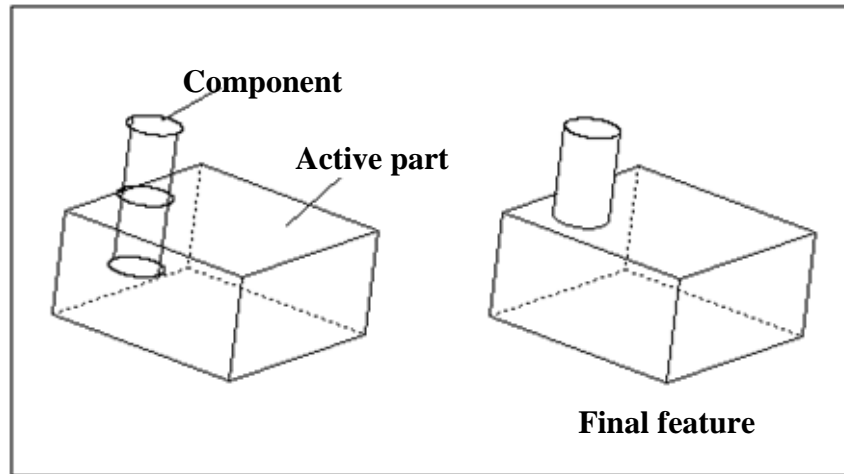


Use this command to create an **unattached base** feature by merging a copy of a component into the active part.

The **wire** command option determines if wireframe geometry (e.g., curves and points) existing in a component are to be merged.

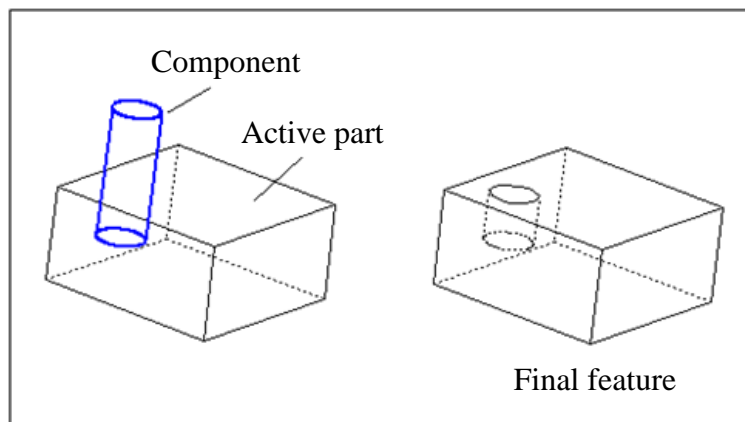
## Component Merge (Add)

Use this command to create a **boss** feature by merging a copy of a component and **joining** it to the active part.



Use this command to create a **boss** feature by merging a copy of a component and **joining** it to the active part.

## Component Merge (Remove)



Use this command to create a **cut** feature by merging a copy of a component and **cutting** it from the active part.

## Merging components as features on parts

The technique of merging components, and adding or removing them from an active part can be very convenient. Consider the following examples.

If you had a **family of parts** that shared the same features. It would be an advantage not to have to construct the same feature several times. On plastic bottles for example, it could be that different capacity bottles share the same neck shape. In this case the neck needs only to be modeled once and then attached to the different sized bottles.

Sometimes certain features are inherently complex, and can require several operations to construct. If another area of the part needs to be changed, a regen would be required, which would of course include the complex features. Why not model these separately and merge them into the active part. The regen on the active part would then not have to include re-creating the complex feature(s) only re-attaching them. Ensuring a quicker and more robust design change.

### **Remove the component from the base creating the core and cavity**

**Component Merge (Remove)** from SHAPES.

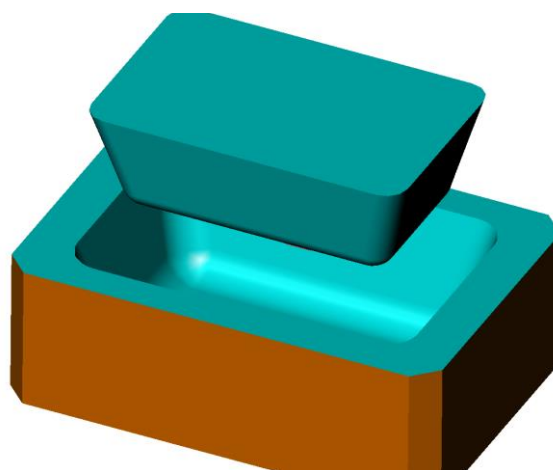
Select the **Component Merge** (Remove option) command.

Pick the **component**.

Accept all other defaults.

The shelled component is now removed from the base creating both halves of the Mold.

The parts in the image shown have been moved apart for clarification.



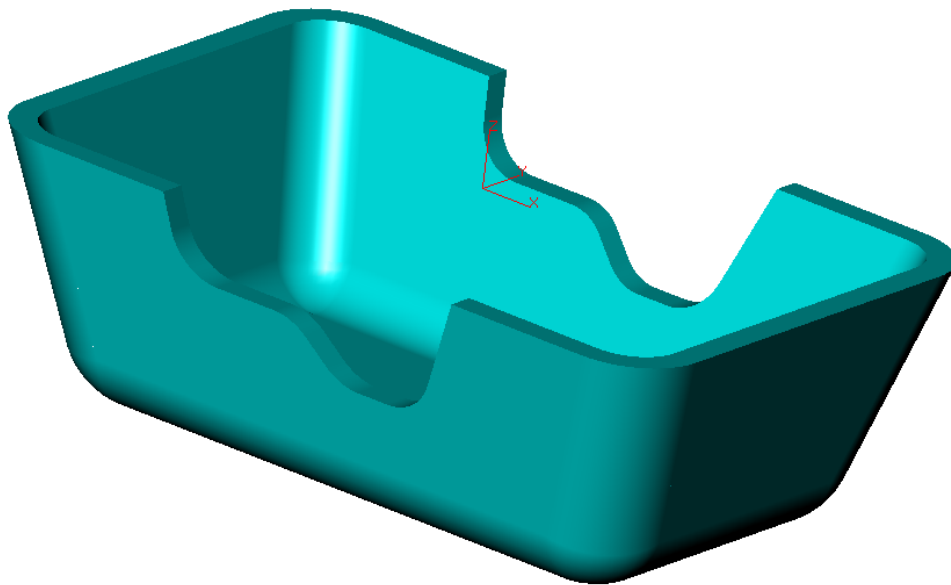
**Removing components that require shut-offs.**

The previous example showed a part that had a continuous parting face. Our next example describes a method that can be used when “shut-offs” are required.

**Undo** the **Merge** and **Insert Component** Commands.

Insert (right mouse button) the component “**Non\_man\_part2**”.

The image below shows the part with the base part blanked for clarity. Notice the cut outs on each side of the component.

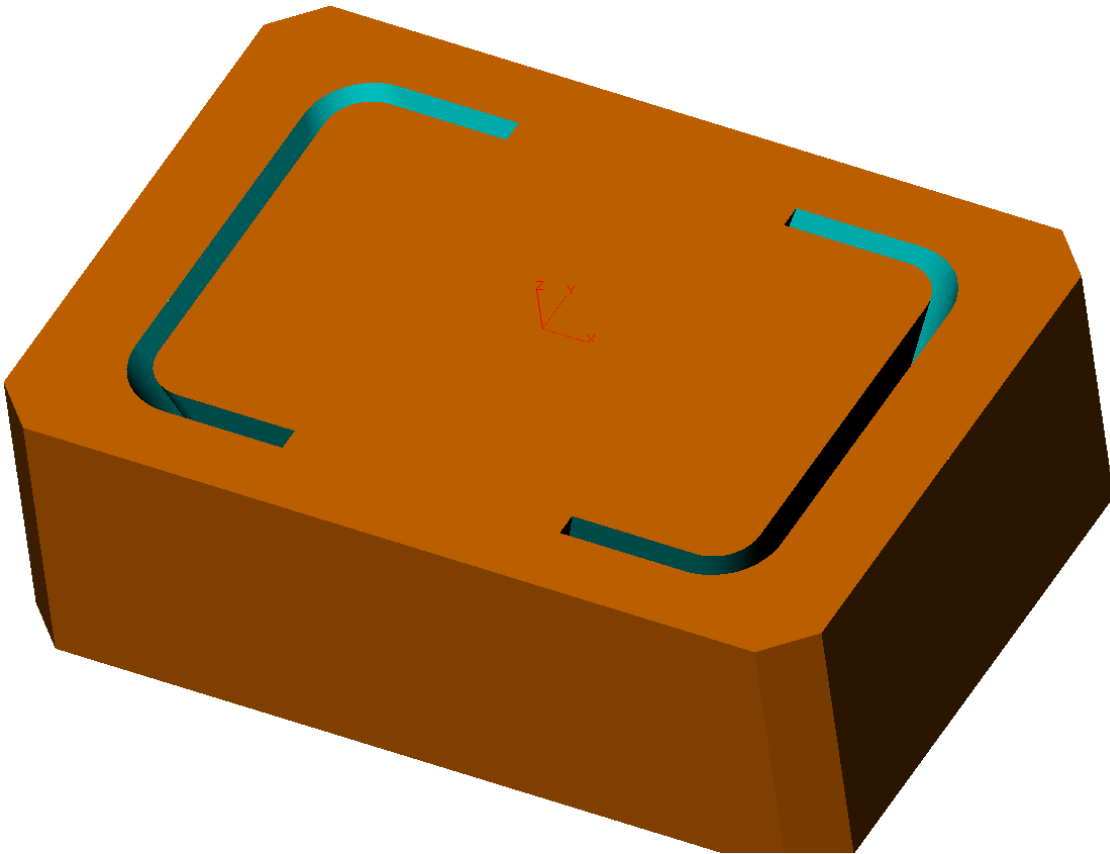


**Remove the component from the active part.**



Using the same method as previously described.

Remove the component from the active part.



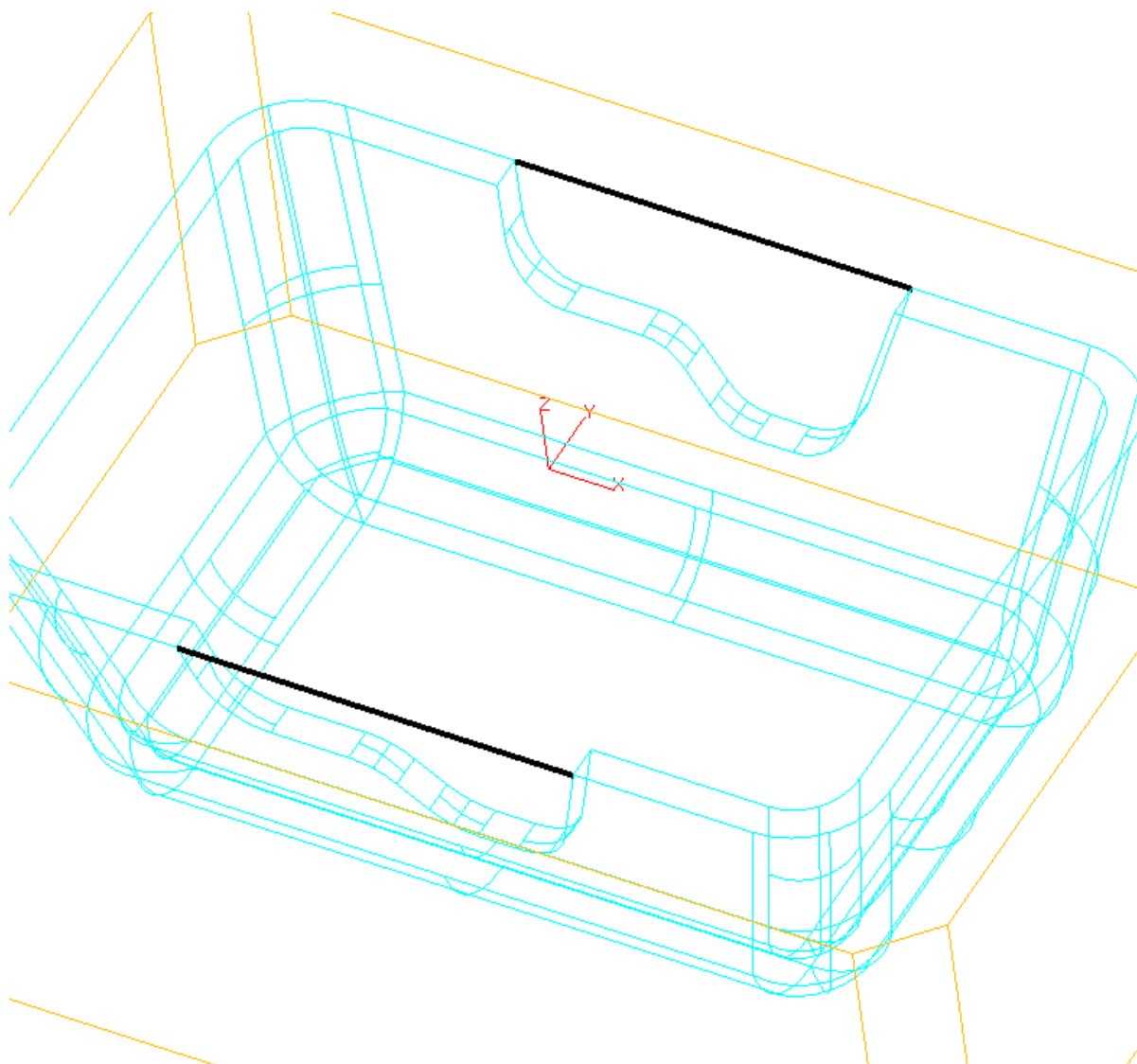
Notice that because of the cut outs in the part, the core and cavity of the Mold are still attached.

We need to create “shut offs”.

‘Shut-offs’ are faces that are used to separate the core from the cavity.

### Creating the shut off surfaces.

Create two, 2pt lines at the intersection between the top face and the “outside” face of the part as shown in black.



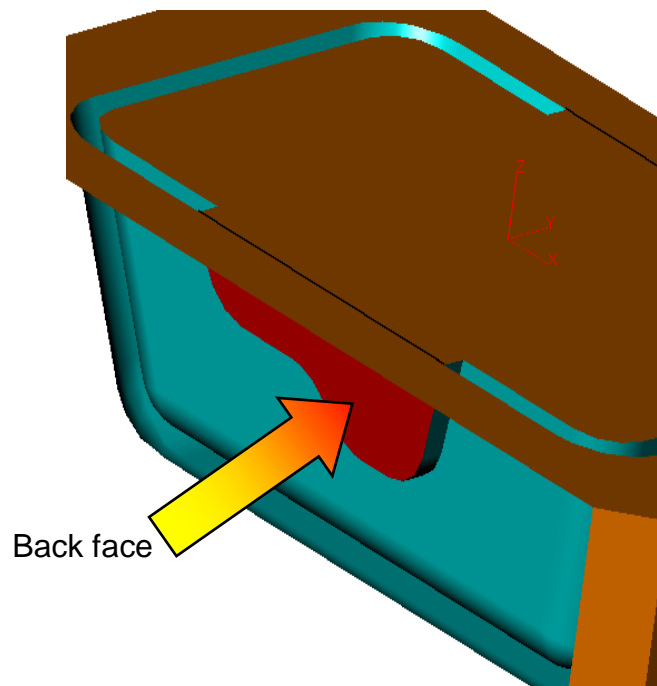
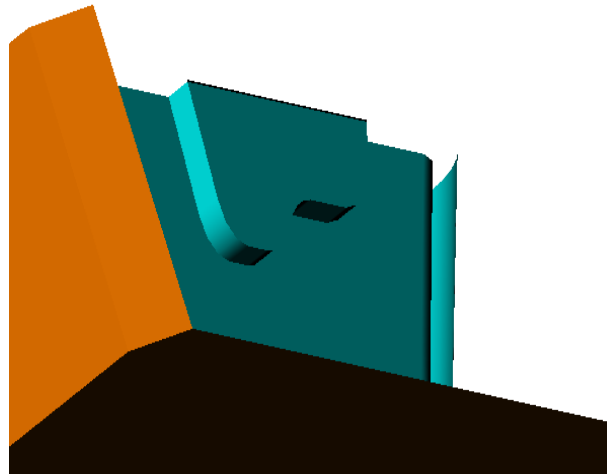
## Create two trimmed planar faces

Select the **trimmed plane** command from FREE FORM Tab.



Create a planar face on the “**outside**” of the part.

For each face, select the line you created in the last step and **chain** select the connected outside edges. Do each separately.



The two images above (displayed with some surfaces blanked for clarity) show the two trimmed surfaces.

Notice that the direction of the **positive** surface normals are in the same direction i.e. **positive Y**.

However, with respect to the **part** they are in fact pointing in **opposite** directions, that is, one surface is pointing **towards** material and the other surface is pointing **away** from material.

We **must** reverse the direction of one of the surfaces.

Which one we choose will affect the result of the **feature remove** command that we will shortly perform


The direction of the normals will either produce a “**core**” or a “**cavity**”.

Let’s try both, as described on the next page.

### Reverse the direction of ONE of the trimmed faces

This next instruction will make more sense if you **Explode** the outside face and **Blank** that new shape.

Use the **Explode faces command**  from the shape Tab to explode the outside face.

**Blank**  the exploded faces with **filter** on shape.

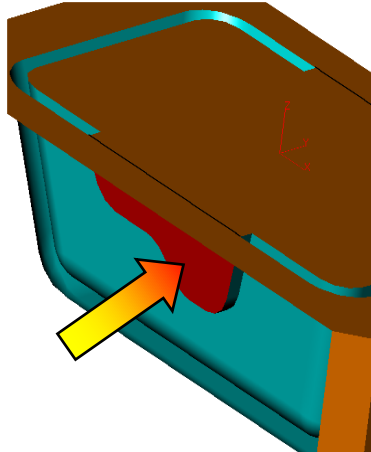
**Reverse** the direction of one of the trimmed surfaces.


Select the **reverse face direction** command.

Set the **pick filter** to “**face**”.

Pick the surface that lies in **negative Y** (shown at the right).





**Unblank**  the exploded faces and Sew (Heal Tab) them back.

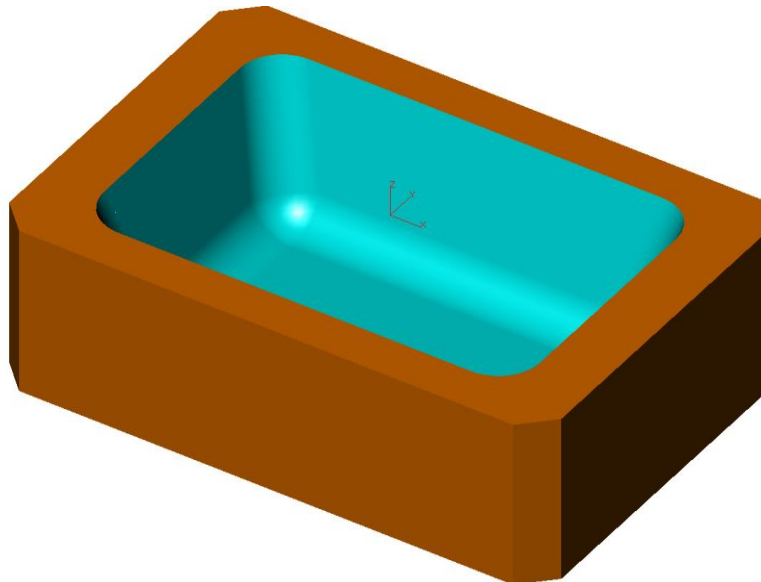
Remove the faces from the solid to create the mold cavity.

Use the **Combine**  - **Remove** command  .

Pick the **main** part as the **base shape**.

Window pick the **two trimmed surfaces** as the **Operator**.

The result is shown below.



## Creating the core.

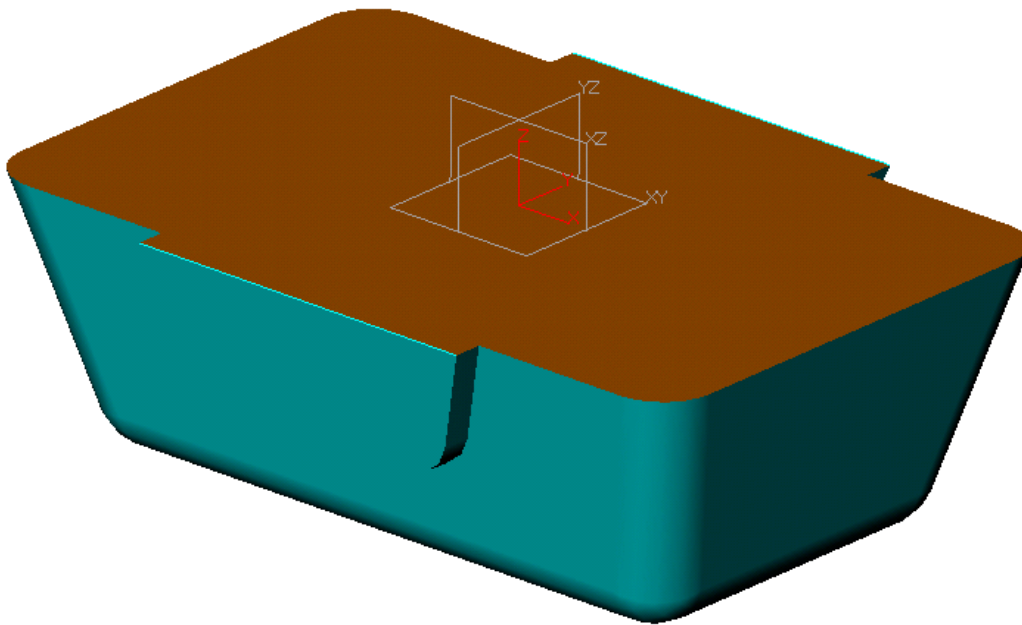
We used Non-manifold geometry to produce the mold cavity shape as described on the previous page.

Let's now create the mold core shape.

Undo the operations up to and including the Reverse Face Direction command.

Reverse the direction of the other (opposite) face. Remember that they have to face the same direction with respect to the part (inside or outside).

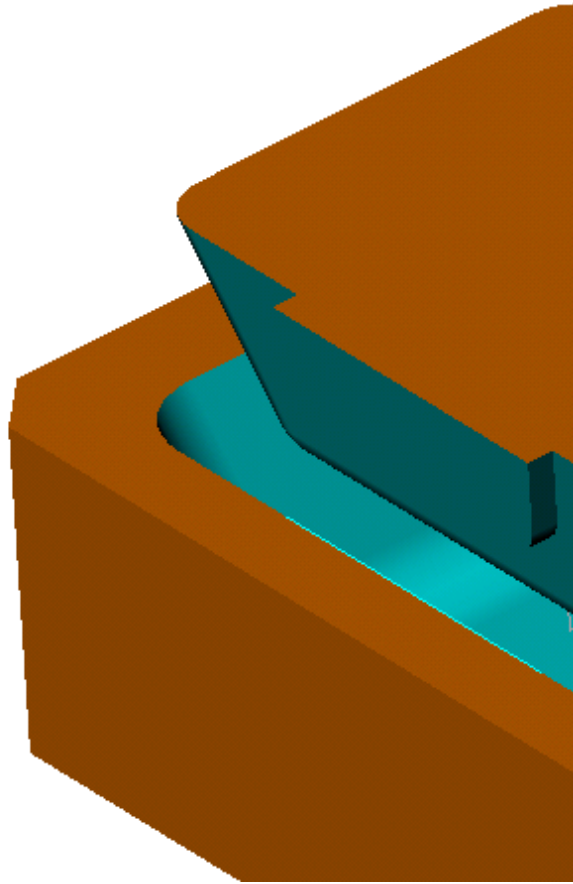
Repeat the **Combine – Remove** operation .



As you have seen, the difference between creating a core or cavity is usually editing one or two commands.

HINT: Once you have finished modeling the first side of the mold, Exit the part and COPY / PASTE your part. You will also want to RENAME the copy so you know what it is. Edit the new part, Step through the History and make the changes as described above.

Now you will have a separate part for the CORE and CAVITY.



## **Chapter 2    Face Editing**

We have seen how ZW3D can Sew surfaces together, and also the Heal / Analyze topology capabilities. These commands are indeed very powerful, but sometimes in themselves they are not enough. The user must be able to discreetly edit individual faces or Shapes to suit his requirements. ZW3D supports a multitude of surface editing commands that can be used to change the shape, or to repair existing geometry. In the following examples we will discuss a number of different face/shape editing options.

### Reduce points

Sometimes faces have an unnecessarily large amount of control points. This means the software has to work that much harder when executing any type of task that would require CPU intensive calculations, for example, surface to surface intersections or shading etc. Also if the faces are exported into a system that requires “meshing” (FEA for example), then problems can arise as surfaces with a large amount of control points are more difficult and time consuming to mesh.

There could be many reasons for copious amounts of control points on surfaces. Surfaces that are analytical in form such as planes and cylinders etc. can be defined arithmetically with the lowest possible amount of control points necessary to define the shape. However, when other surfaces are created from the analytical shape, such as offsets, fillets, and so on, some Cad systems fail to recognize the newly created surfaces as analytical shapes. They would create the correct shape using different techniques that inherently increase the number of control points that are required to define the “non analytical” form.

Some Cad systems use methods (other than the more modern Nurbs approach) to define surfaces. They must be created with a high “order” or “degree” to maintain the desired shape. The higher the order the more control points needed. If the user has to import data from such a system, he will quickly discover the surfaces seem to have an incredible amount of control points on them. Surface calculations will then become unnecessarily complex and more difficult to solve.

The reduce points command in ZW3D can be used to filter out, or weed, not needed points, still maintaining the correct shape (within a given tolerance).

Another use would be if the user wanted to smooth or fair the surface. Deliberately specifying a large tolerance drastically reduces the number of control points, and smoothes the surface. The user should be aware however that the surface shape then obviously changes.

### Reduce points example

Make sure that the system tolerance is set to 0.01mm.

Edit the Part “**Reduce\_data**”.



**Shade** the part. **Zoom** in closely, and **refine facets**. (View pulldown)

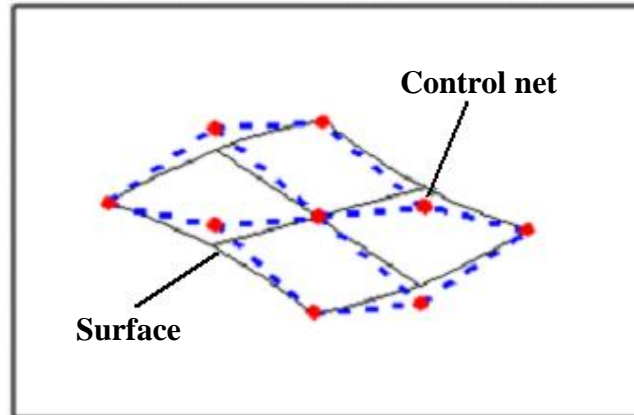
Notice it seems to take a relatively long time. Zoom back out.

There are more control points on some of the faces than necessary.

Before we perform a point reduction, let’s review two inquire options.

**Inquire – Face – Control Polygon**, and **Inquire – Face - Continuity Error**. As described on the next page.

**Inquire** pull down menus– **Face – Control Polygon (or Control Polygon**  **from the Inquire Tab)**



The user can display the control net of selected faces, by using the **control polygon** command.

Select the command.

Position the mouse pointer over the desired face(s) and the control points will be displayed.

**Inquire** pull down menus – **Face - Continuity Error( or Display continuity error**  **from the Inquire Tab)**

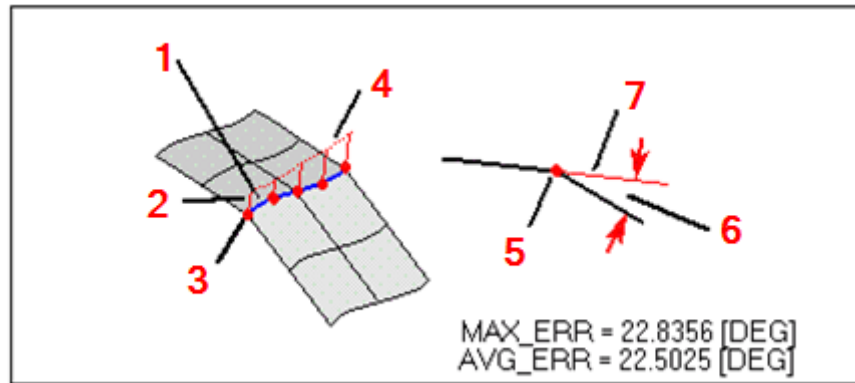
.

#### **Continuity Options-Position/Tangent.**

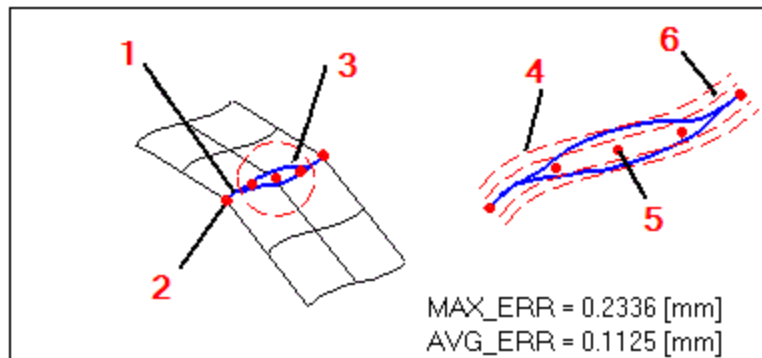
**Tangent** - Use this option to analyze the continuity across two mating faces at a selected edge.

The edge must be shared by both faces. (sewn) The results displayed are the maximum and average error measured in degrees.

**Position** - Displays the C0 continuity error. If gaps are present, the maximum and average gap distance between the two faces along the selected edge is calculated and displayed.



1: Shared Face 2: Discontinuity At Sample Point 3: Sample Points 4: Rate of Discontinuity Across Sample Points 5: Sample Point 6: Degree of Continuity Error 7: Zero Continuity Error



### C0 Discontinuity Error

1: Shared Face Edge 2: Sample Points 3: C0 Discontinuity 4: MAX\_ERR  
5: Sample Point 7: Sample Point

### Check the control points on our imported Iges data

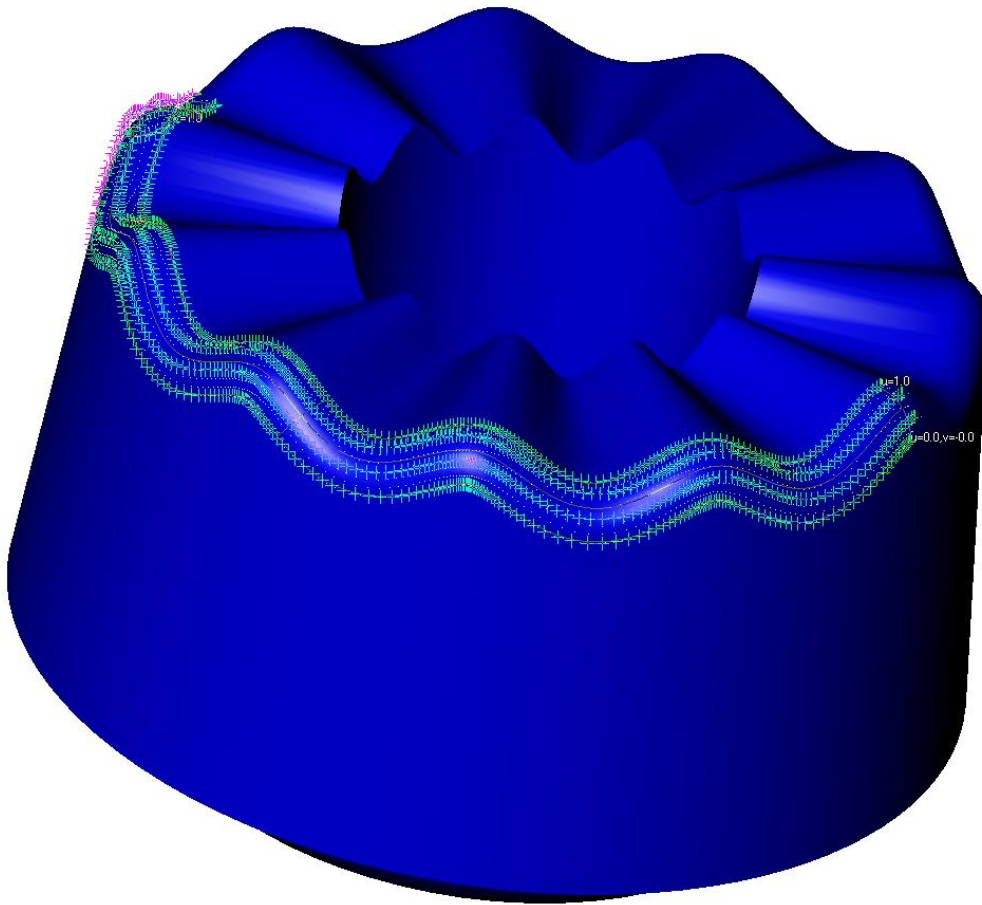
Use the **Inquire** pull down menus – **Face – Control Polygon** command.

Position the mouse pointer over the faces of the part.

Notice the control net is displayed on each face as you move the mouse.

Most of the faces on our part are fine, but check the fillets on the top faces.

These have far too many control points. (see below)



Use the **Inquire** pull down menus – **Face - Continuity Error** command.

Set the “ContLevel” to be **tangent**.

Place the mouse pointer over several edges of the part.

Notice the display graph on faces that are not tangent. (wireframe view may be easier)  
Pay particular attention to the edges of the fillet faces that have a dense control net (as shown above).

**Reduce the point data ( Edit pull down menus>Face>Modify NURBS>Reduce data)**

Reduce the point data on the part.

You can select individual faces or multiple faces with window pick for example.

For our part pick the top front fillet face only.

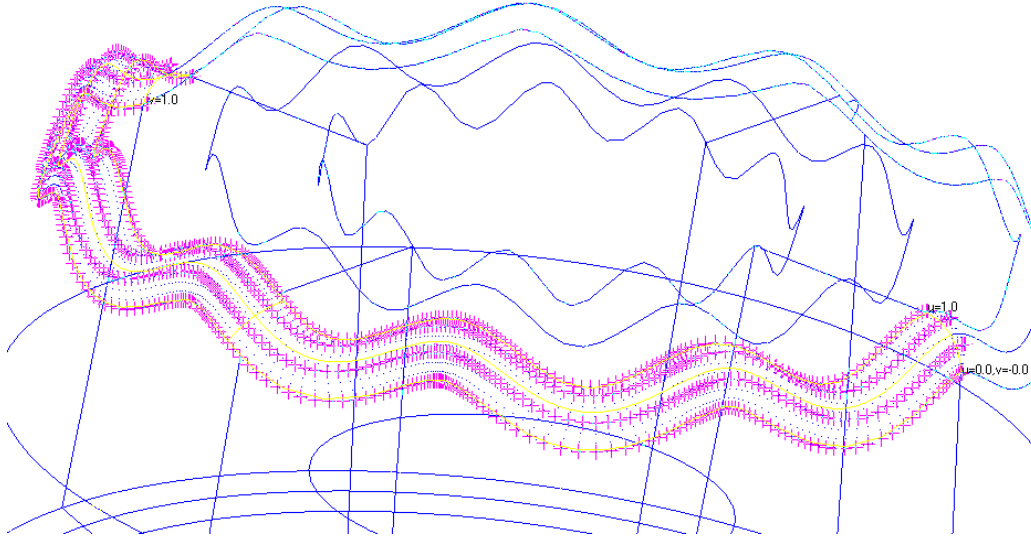
As shown below.

Input the tolerance for point reduction of **0.01mm**.

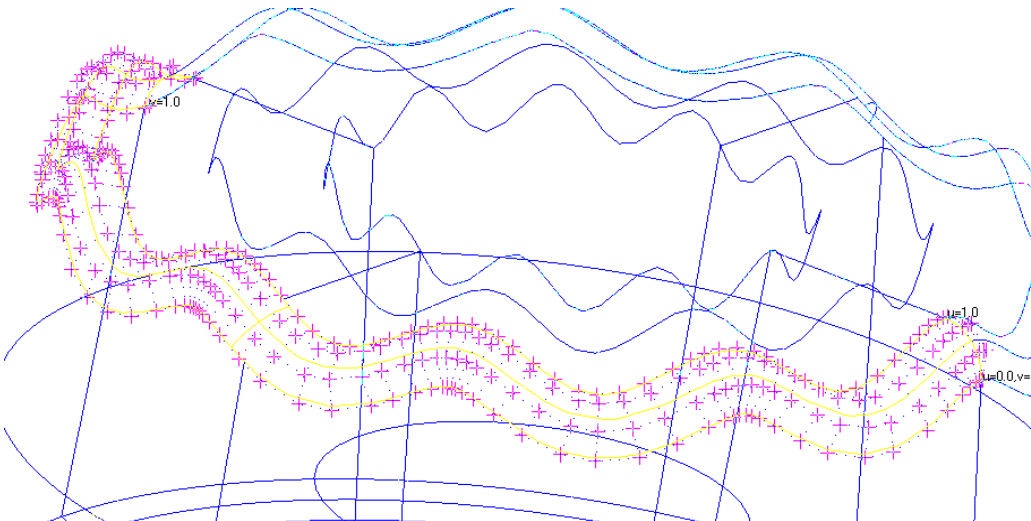


Then compare and review with your instructor, the point densities and continuity error of this face and the other “identical” fillet face.

Before point reduction.



After point reduction to **0.01mm**. Control points reduced from 912 to 320. 65% reduction.



## Extending edges of faces( from Free Form Tab)

Use this command to extend the edge of a face.

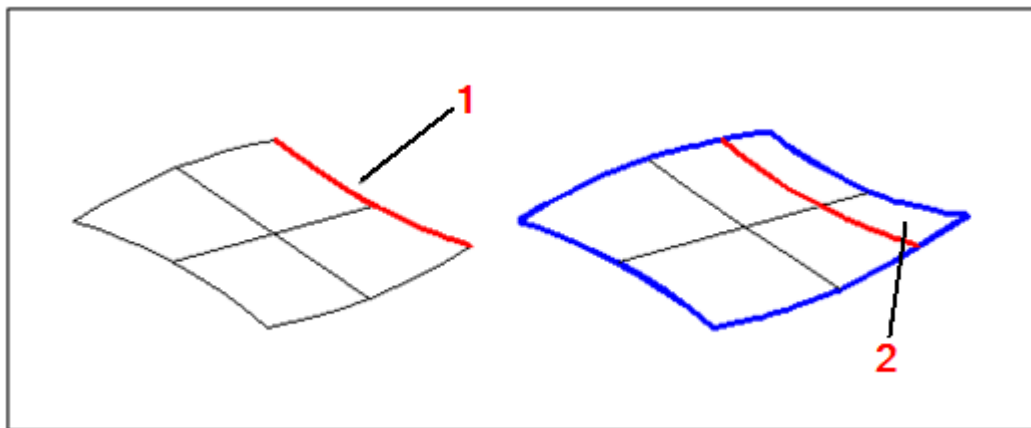
Pick the **face** to extend.

Pick the **edge** where the face is to extend.

Choose the desired options.

Enter the amount to extend the face.

Note a **plus** value **extends** the face, and a **minus** value **trims** the face.



1:Edge to extend 2:Extension

## Extension

Use this command option to control the path of extended curves and faces.

Available options are Linear, Circular, Reflect and Curvature diminishing.

See the HELP Manual for examples.

## Delete Loop ( from Free Form Tab)

Trim loops define the trim shape of faces. The outermost trim shape is controlled by the outer trim loop(s) and the innermost (holes) are controlled by the inner trim loop(s). All Faces have trim loops. The most basic case would be a face that is not trimmed, where the only trim loop is an outer one, and is exactly the same as the outer periphery of the untrimmed surface. If selected trim loops are deleted, the face is restored to the exact same shape as it was originally, before the trimming took place. This is not the same as, for example, patching a hole in a doubly curved face. Although ZW3D has many tools that will enable the user to maintain tangency (and even curvature), the new face would not be identical to the original. However the user could remove the trim loop and the “repaired” face is absolutely the same as the original. In fact it is the original face but with the selected trim loop information removed. The user can also delete outer trim loops. As with the “hole” example, this is not the same as extending the surface. Often the outer shape can be changed by first deleting the outer trim loop, and then re-trimming the face. However the new trim cannot extend beyond the original outer shape of the untrimmed face. If this is required then the user must find an alternative method (such as extending the face).

In ZW3D the user can decide which trim loop(s) to delete.

**Inner** - Delete all inner loops, Outer loop stays the same.

**Outer** - Delete the current outer loop. The outer edges of the untrimmed surface become the new outer boundary.

**All** - All loops are removed. The outer edges of the untrimmed surface become new outer boundary.

**Selected** - Only selected loops are deleted. If one of those selected is the outer loop, then the outer edges of the untrimmed surface become new outer boundary.

The ability to selectively remove trim loops from faces is a very powerful tool. Consider imported data that has bad trims for example. These faces can be repaired, knowing that the face topology will not be affected.

We have created an iges file with exaggerated trim loop errors.

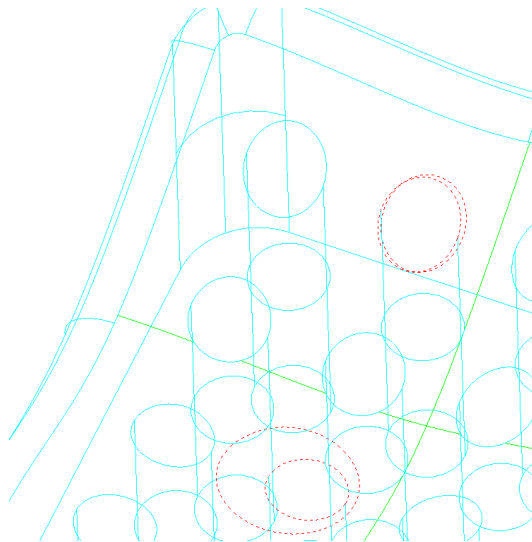
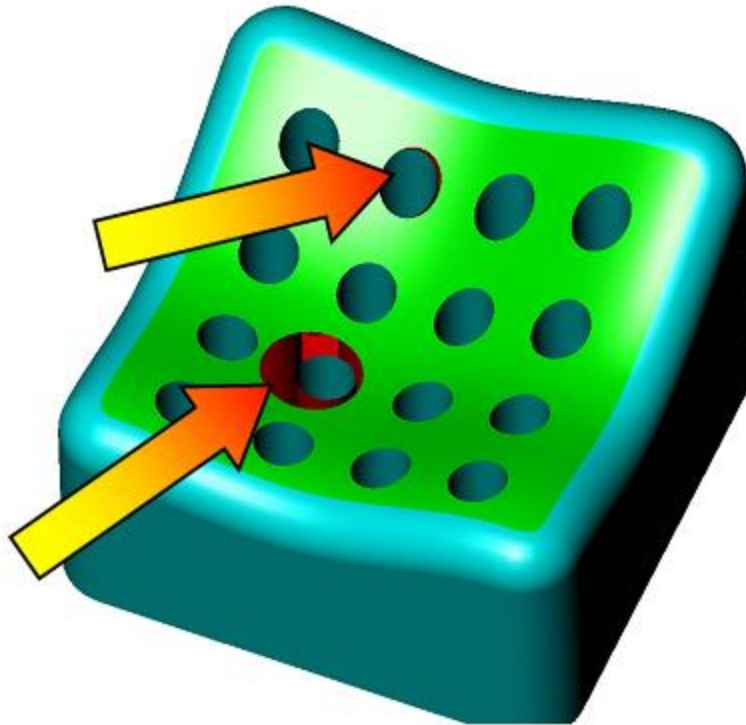
Clear the current session.


Open a new part.

Open part named “Bad\_Trim\_IGES\_Import”.

Set the Iges import filter to read surfaces only.

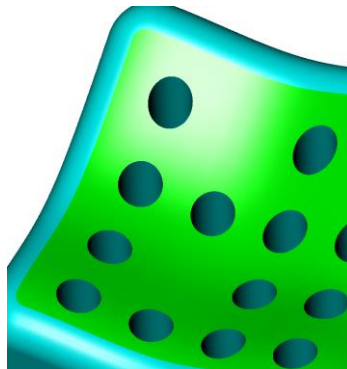
The shaded image below and the wire frame image below left, clearly show the deliberate bad trims on two holes on the **green** face.




Using the Delete loop  command , delete the bad trims on the **green** face, using the **'Selected' Loop** option.

Make sure you pick the trim loop to be removed from the **green** face **not** on the hole cylinder faces.

The result is as below.



The green face can now be trimmed to the hole cylinders using **Split with curves >**

**Trim to curves**  (Free Form Tab). Make sure the **pick filter** is set to **face** and not **shape**.

Use the **Sew** command (**Heal Tab**) to sew, stitch the faces together.

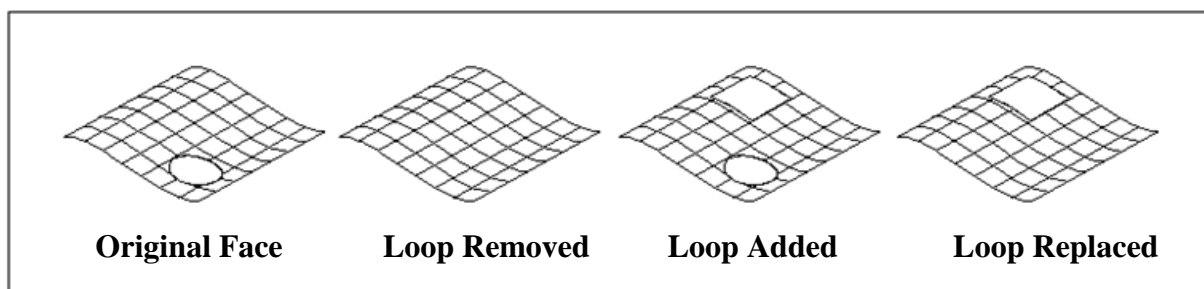
**Replace Loop**  from Free Form Tab)

Use this command to **replace** trim loops on a face. The curves selected for new trim loops will be projected normal to the face. Middle-clicking at steps 2 or 3 below will produce varying results. Trim loops can be added, deleted, or replaced.

Select the face to modify.

Select an edge on the loops to be removed or middle-click if no loops are to be removed.

Select curves for the new trim loops or middle-click if no loops are to be added.

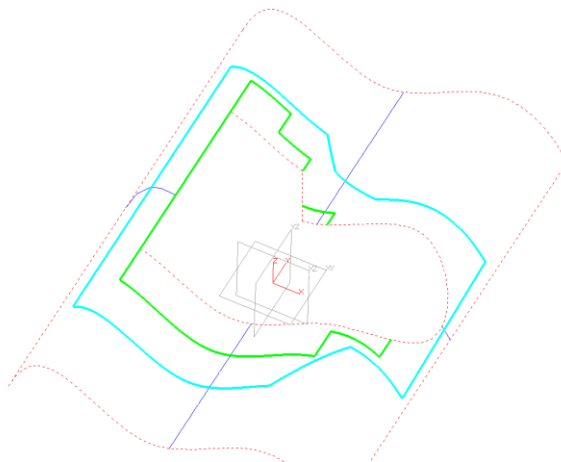


The previous example showed how the user could delete a selected trim loop and re-trim the face. This command allows the user to do the same task with fewer steps.

With help from your instructor, undo the previous example (or clear session and re-import the iges file “Bad\_trim.igs”), and repair the model with the replace loops command. Make sure you first pick the loops in the green surface, and then pick the edges of the cylinders as the loops to replace. Sew the model together.

This command can be used to totally replace trim loops with a completely different shape. The new curves must form a closed loop on the existing face. Consider the following example.

“Exit part” and edit the part “**Replace\_loops**”.



The trimmed face shown opposite has two sketch curves projected in +Z direction onto the face.

The **green** colored curves cannot be used with the replace loop command, as they do not form a **closed loop** on the trimmed face.

The **blue** colored curves do form a **closed loop** and can be used.

If you wish to use the green curves, **delete** the trim loop **before** the sketch is projected.

Invert Loops (  from Free Form Tab)

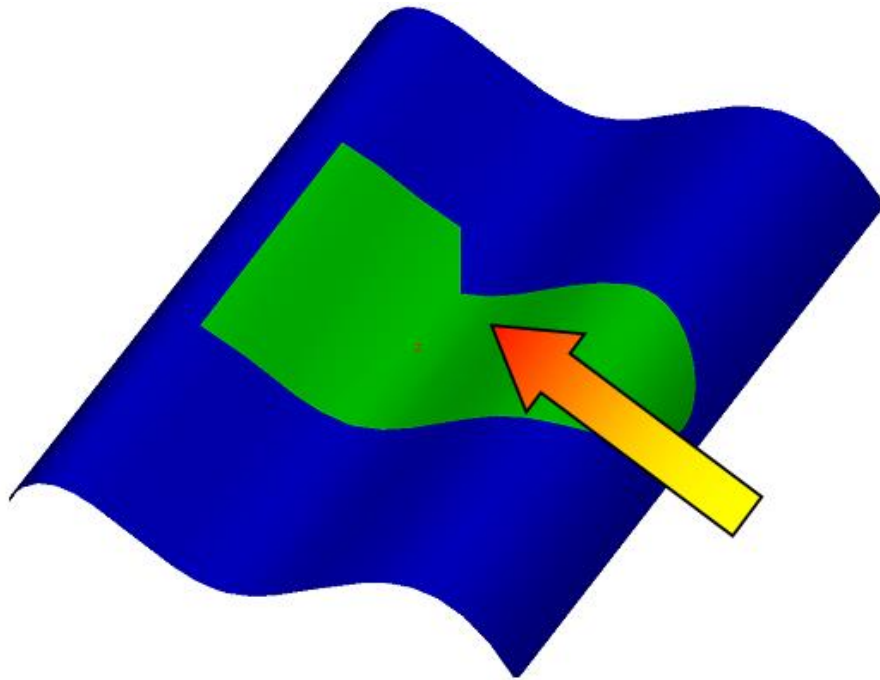
Found under the Delete loop icon on the FREE FORM Tab.

Use this command to create a face by inverting the trim loops of a face. The selected face remains intact and a new face is created. This allows you to create a face where there was once a trimmed hole in a face. The new face will match the mathematical characteristics of the selected face. This command can be used to create parting faces in Mold design if the part model has holes in its topology (consider buttons in a cellular phone) and there are trimmed faces that can be directly converted to parting faces by inverting their trim loops.

“Exit part” and edit the part “Invert\_loops”.

Use the **Invert loops** command. Pick the surface and one edge on the trim loop.

A new face is created as shown below in green.



**Bounds** Option. Erase the blue surface.

Use the same command again.

Use the Bounds option if the face has an outer trim loop. (i.e. the green face)

**Untrimmed Surface** - The outer boundaries of the untrimmed face are used.

**Outmost Loop** - Only the outer most trim loop is used.

**Try both of these options on the green surface.**

## Match Edges ( from Free Form Tab)

Use this command to match or replace the **untrimmed** edge of a **face** with a **curve**.

to the object level and edit the part “**Match\_edge**”.

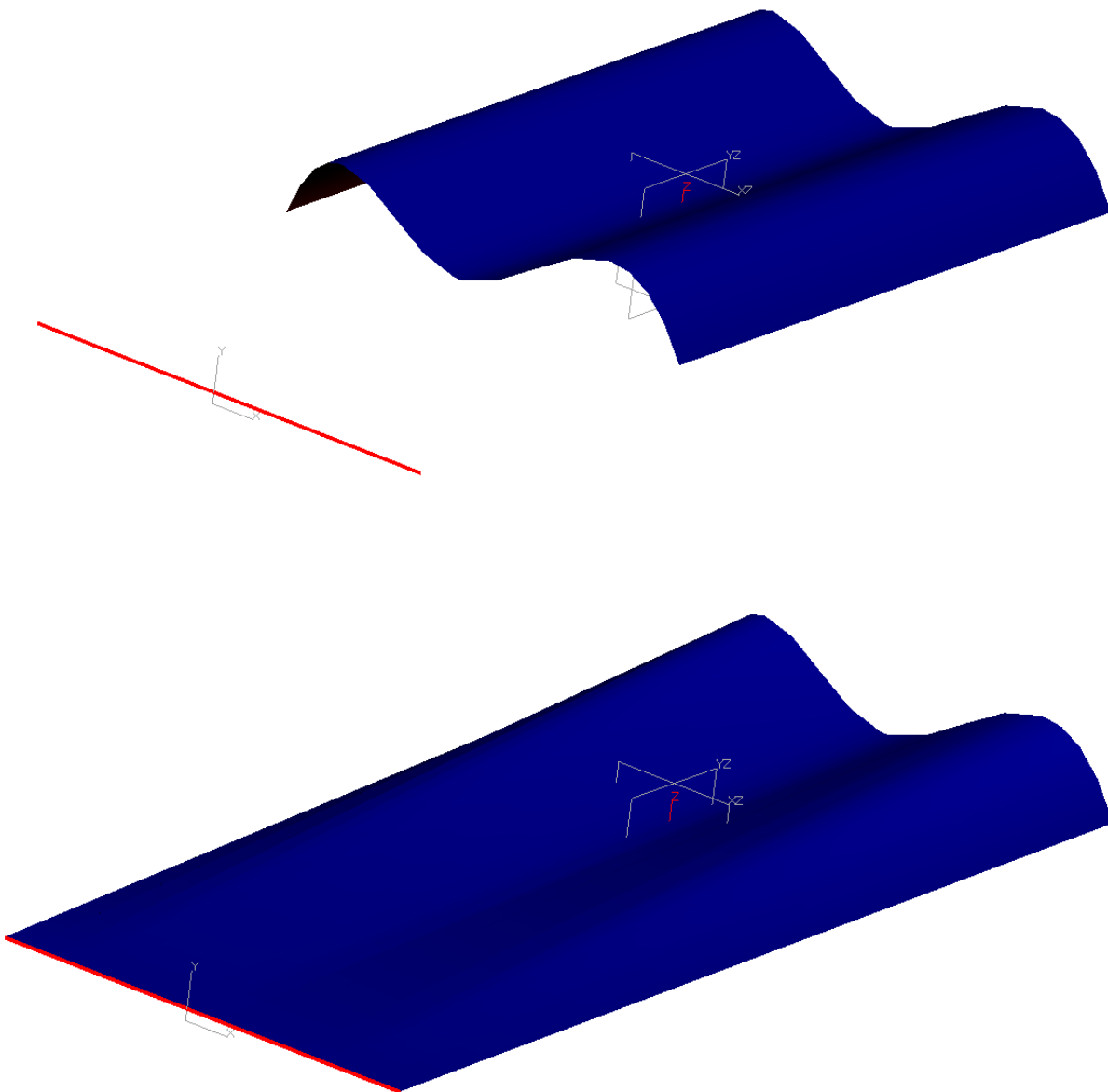
Select the command **match edges**.

Pick the face.

Pick the edge to replace.

Pick the new edge (the red curve).

As shown below.



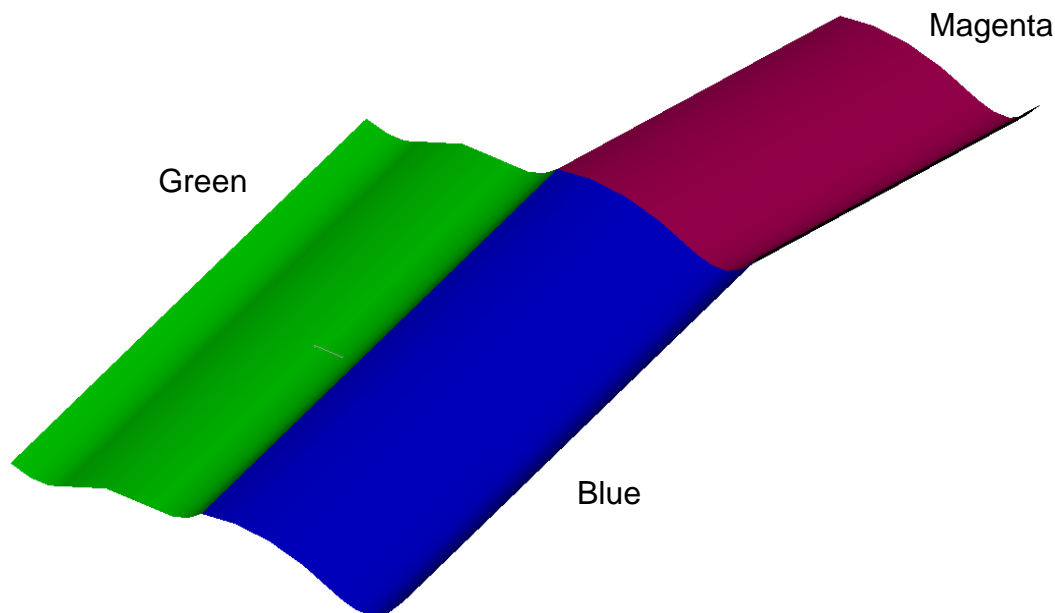


## Match Tangency (Make Faces Tangent Continuous, from Free Form Tab)

Use this command to modify two adjacent faces so that they are tangent continuous along a shared edge. Master faces are those that will remain unmodified while slave faces are those that are forced to match them. Slave faces do not match each other. An exception to this is any master face that has a seam and meets no slave faces. The command will attempt to make its seam tangent.

Previously we have seen how too dense a control net on faces can be a disadvantage. Faces with a minimum number of control points are for the most part, definitely desirable. However faces with a minimum number of control points are not so flexible in terms of changing their shape.

Clear the current session and edit the part “**Tangent\_continous**”.



The part contains three faces. The edges of the faces are not tangent continuous. We will use the Match Tangency command shortly, to make the green and magenta slave faces, tangent to the blue master face.

First, use the **control polygon** command to show the control points on the faces. ( from **Inquire** Tab or **Inquire** pull down menus)

All three faces were created with the extruded face command, and are therefore curved in one direction only. Each face has five control points in the “U” direction, and only two control points in the “V” direction. This means that although we could change the shape of the faces in “U” we cannot change the shape in “V”, as there are not enough “V” points. Let’s prove this.

Use the **Match tangency** command.

Pick the **blue** face as the **master** face and the **magenta** face as the **faces to be reshaped**.

Notice the error that Surfaces or curves may require more DOF.

The error states that the face may require more DOF (degrees of freedom), which in this case means that the magenta face does not have enough control points in the “V” direction.

-We can increase the amount of control points, by using the **Refit Surface** command under the **EDIT** pull down menu>**Face>Modify NURBS**.

-Pick the **face**.

-Select the **V-Dir** option.

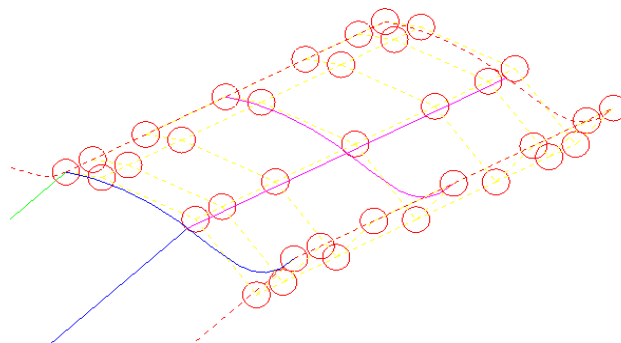
-Specify the number of points (**FIVE**).

-Select **degree 3**.

The resulting increase in control points is shown below.

How many control points you enter will affect the way in which the face will change shape.

The more control points that you apply to a face the more **localized** the change.



## Make all the faces tangent continuous

Once more, using the **match tangency** command.

Pick the **blue face** as the **master** face and pick **both** the **magenta** and **green** faces as the **faces to be reshaped**.

All of the faces are now tangent continuous.

Question, we increased the amount of control points on the **magenta** face, but not the **green** face, **Why?**

Do not clear session; we will use the same part in the next topic.

## Merge

Use this command to merge faces into one. This is not the same as the sew command as this will convert several separate faces into one shape. Be aware that the merged face is not absolutely identical to the original faces. The edge boundaries of the two faces must coincide within the geometry tolerance and be tangent continuous.

Select the command.

Pick the faces to merge together (pick all three faces).

Select the method to enforce face **normals** along the boundary edges. (None)

**None** - No surface normal constraints are enforced along boundary edges.

**Inner** - Surface normal vectors to constrain the resulting face at its boundary edges are extracted from the in-bound faces attached to boundary edges and that belong to the set of faces to merge.

**Outer** - Surface normal vectors to constrain the resulting face at its boundary edges are extracted from the out-bound faces that are also attached to boundary edges, but do not belong to the set of faces to merge.

**Averaged** - Surface normal vectors along boundary edges of the resulting face are computed by averaging the surface normal vectors of in-bound and out-bound faces attached to the boundary edges of result face.

## Sample

Use this command option to control the method used to collect sampling points (Edge).

**Bound** - Collect sampling points from exterior boundary edges only.

**Edge** - Collect sampling points from all edges both interior and exterior.

**Face** - Collect sampling points from all edges and merged faces.


**U-Direction** Use this option to select a preferred U-Direction for the faces (**Default**).



**Angle tol** Use this option to set the angle tolerance used to check for tangent continuity along the mating edges (**Default**).

## Splitting shapes to curves or other shapes

What is the difference between dividing a shape and splitting a shape?

Divide and split commands have two separate functions.

The **divide shapes** command  (Shape Tab) allows the user to **divide** the face(s) into **distinctly separate** shapes.

The **Split with faces**  and also the **Split with curves** commands  (Free Form Tab), split the **face(s)**, but the **split face(s)** remain **joined** as a **faceset (shape)**.

It would be similar to **dividing** the face(s) and then **joining** the divided faces back together.

**Split** faces cannot be moved apart unless they are first **exploded** from each other.

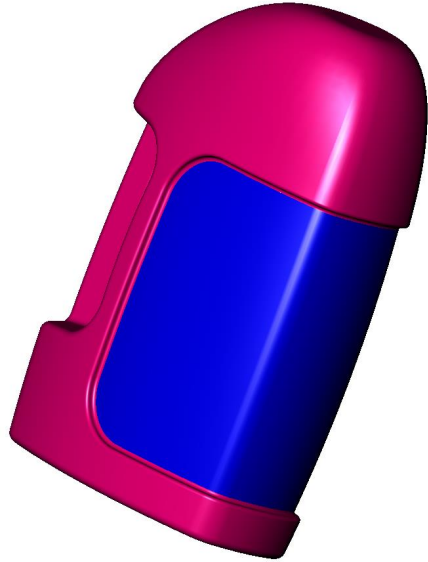
The ability to split faces is a powerful tool to ZW3D modeling capability.

“Exit part” and edit the part “**Split\_shape**”.



The bottle part design requires a stepped feature, based on the sketch shape.

We will create curves on the part, split the relevant faces and offset the new faces creating the step form in the bottle.



### Creating the stepped form

Using the **Extrude** command. (Shape Tab)

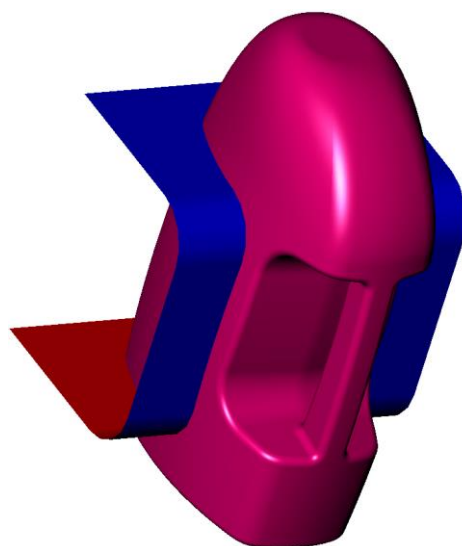
Extrude the sketch (**set the pick filter to sketch if necessary**).


Direction **Z**.

Start **-100mm**.

End **+100mm**.

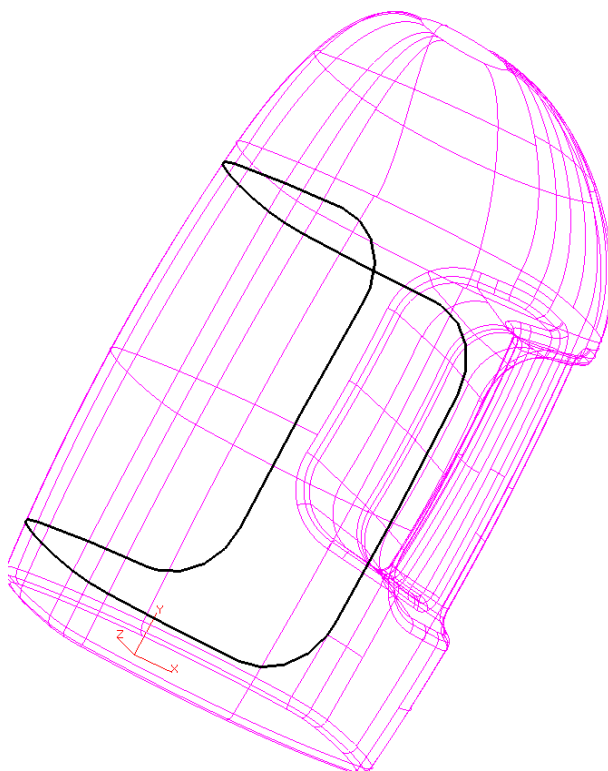
As shown below.



Set the **pick filter to shape** and create **intersection curve**  (found in the WireFrame Tab) between the **Shapes**.

**Erase** the **extruded** shape.

As shown opposite.



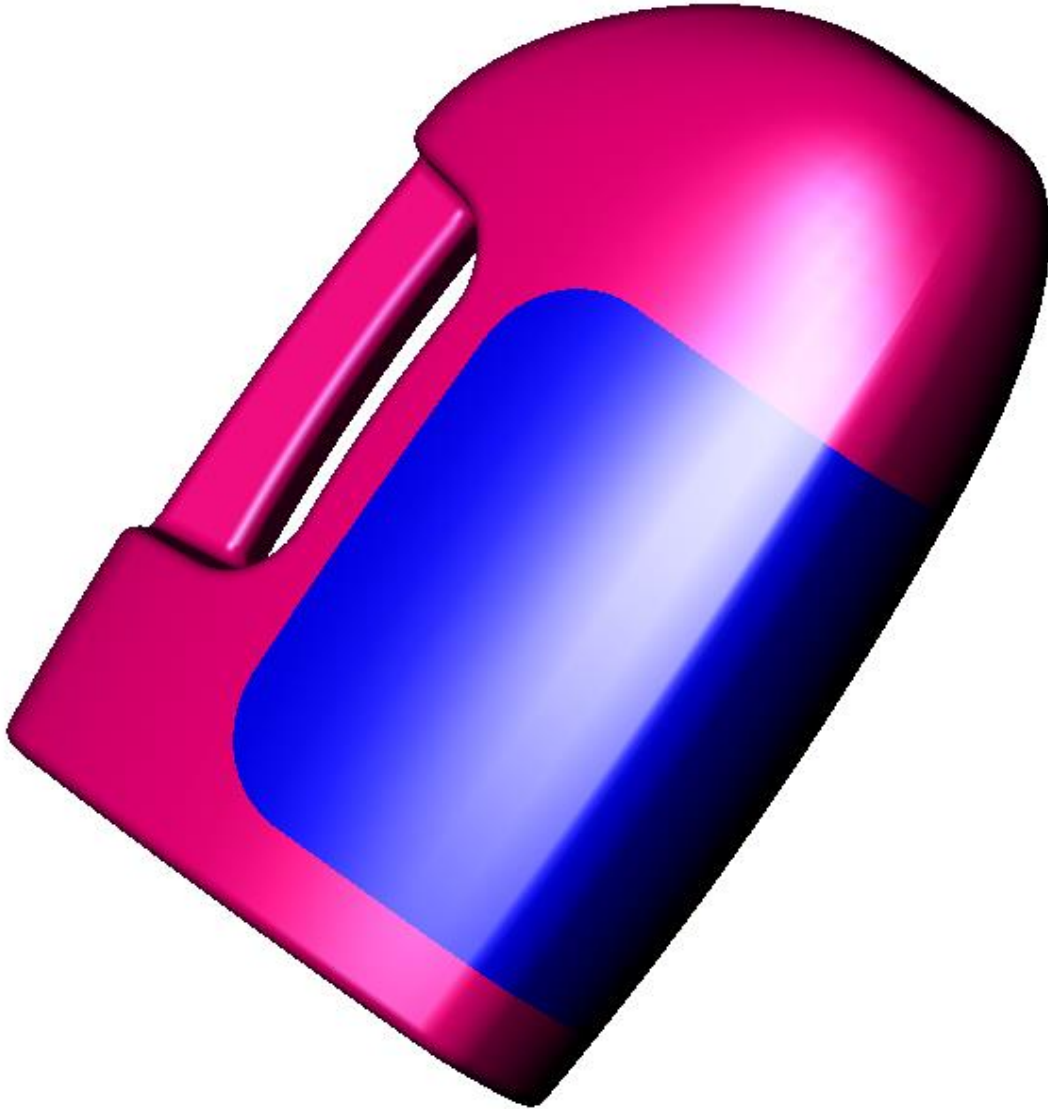
Using the **Split with curves** command. (**Free Form Tab**)

Pick the **bottle** (filter set to **shape**).

Chain select the **curves** (pick filter set to **curve**).

When the command is complete, **erase** the **curves**. (Set the pick filter to **curve**)

As shown below (the new faces have been shaded **blue** for clarity).



The blue faces shown on the previous page are split faces, but they are still part of the shape.


Use the **Face offset** command (Shape Tab) and offset the **five blue** faces **-5mm**.

Use the **fillet** command (Shape Tab), **chain** select the new edges and apply a **2mm** fillet to the inside and outside edges. As shown below.





**Modify the offset value.**

Display all dimensions. 

Double-click on **-5offset** to change to **+5offset**.

The part history will regen automatically.

As shown below.



## Unfold onto Plane (from Free Form)

Use this command from the FreeForm Tab to create a set of **2D planar curves** that are flattened from the edges of a **3D face**. You can position the curves on the plane using a **base** and **target** point or select additional curves on the face other than its edges using the **FlatCvs** command option. Only faces that are **linear** in either their **U** or **V** parameter space can be flattened with this command.

Select a **face**.

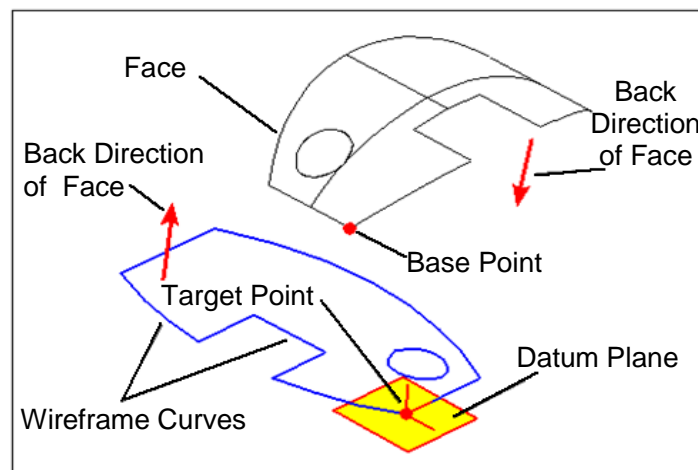
Select a **base** point for unfolding or middle-click to accept a default base point. You can position this base point on the datum plane that the flatten curves will be located on.

Select a **plane** to locate the curves on or middle-click for the default **XY Plane**.

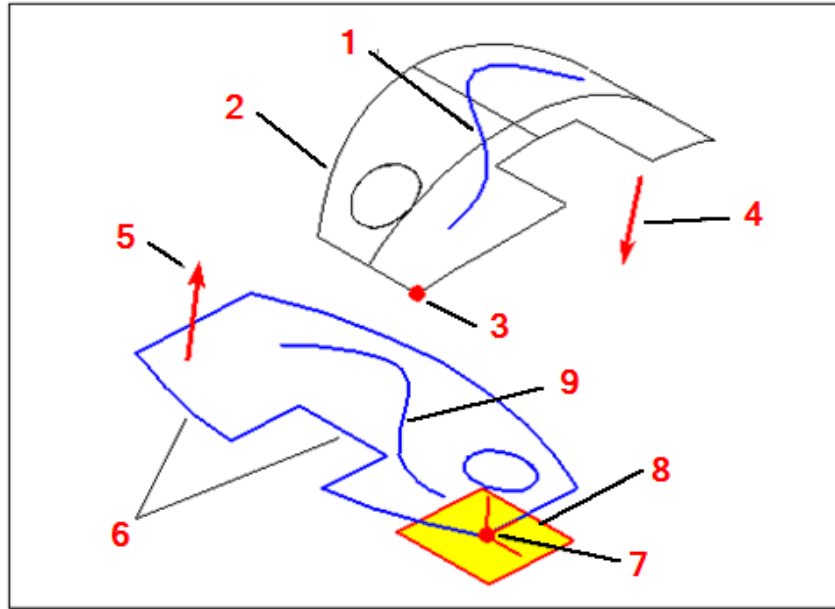
Select a **target** point on the plane for the new **origin** or middle-click to accept the default 0,0 origin of the plane. The base point in step 2 will be positioned at the target point.

Select the **curves** command option to select additional curves that lie on the face.

The selected curves will be flattened along with the face edges.



Flatten Curves from Face



1: Additional Curves on Face 2: Face 3: Base Point 4: Back Direction of Face 5: Back Direction of Face 6: Wireframe Curves 7: Target Point 8: Datum Plane 9: Additional Curve

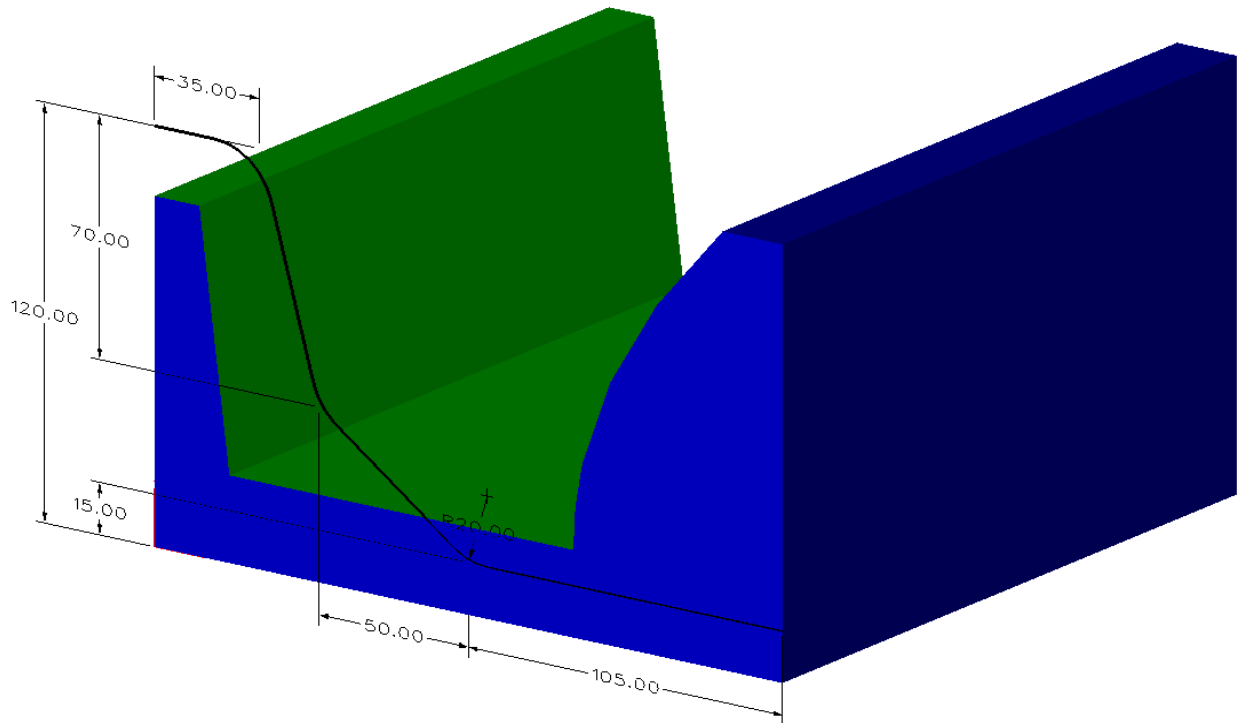
Clear the session.

Edit the “**Flatten\_faces**” part and create flattened curves as shown above.

## Replace Faces (Shape Tab)

Use this command to replace faces of a solid with a single face or an open shape.  
(Shape Tab under the last icon on the right)

“Exit part” and edit the part “**Replace\_face**”.



Notice the sketch (shown in **black** above).

We will **extrude** an open shape based on the sketch, and use this shape to replace the **3 green** faces shown.

The Replace face command can replace several faces. It will add material where necessary and subtract material where necessary.

**Extrude** the faces.

**Extrude** the faces from the **sketch**.

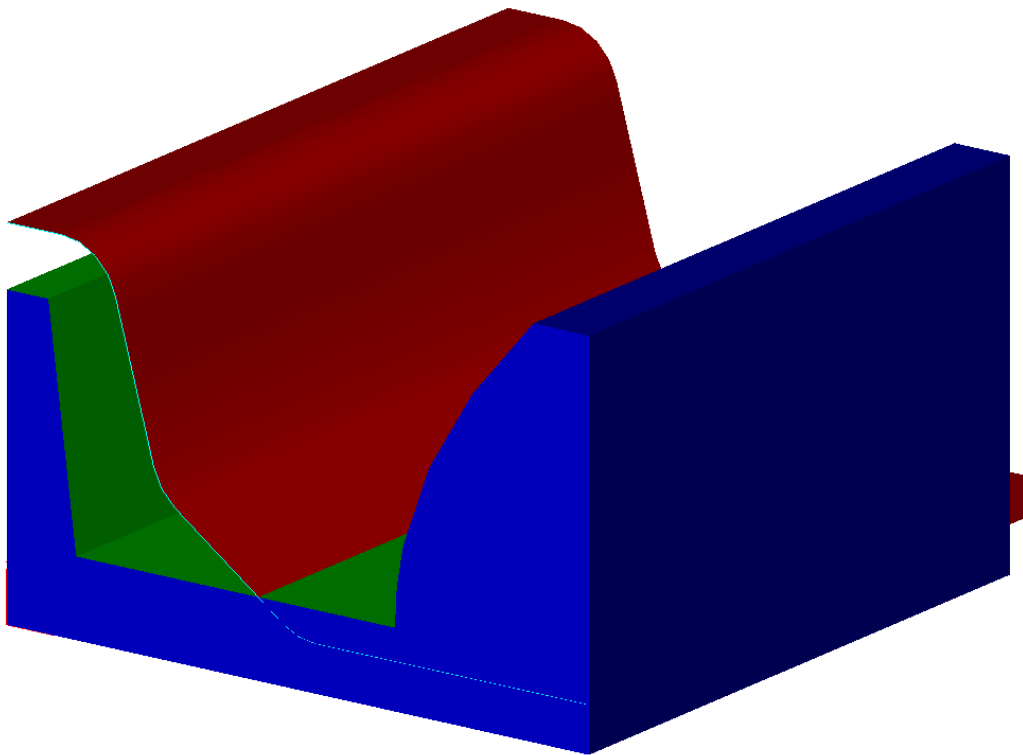
Direction **Y-Axis**.

Start **-10**.

End **250**.

Pick **OK**.


As shown below.



## Replace the green faces

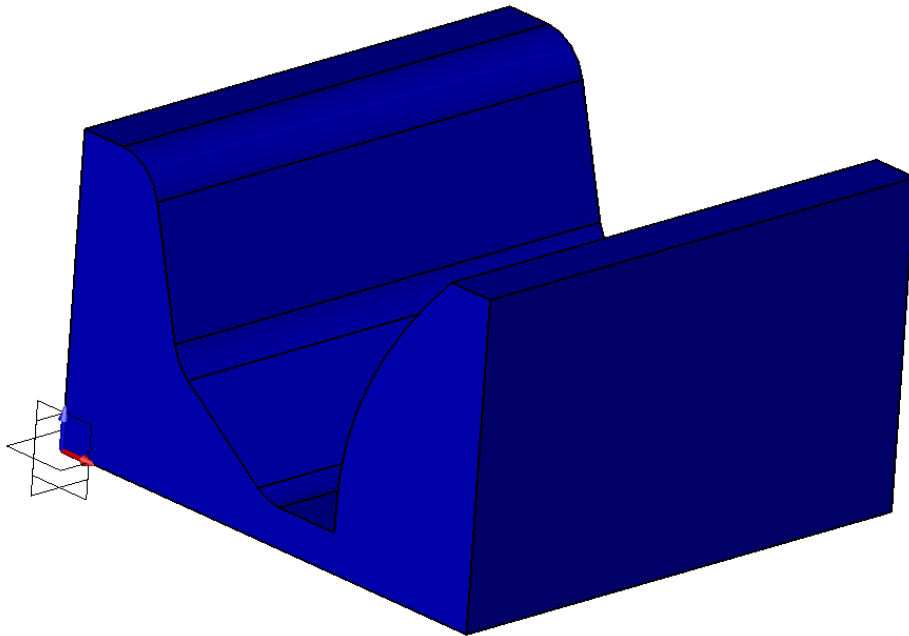
Select the **Replace** command in the Shape Tab.



Pick the **three green** faces as the **Base faces**.

Pick the **attribute filter** icon , select the color **green**, and window pick the three faces. Set the filter back to **ignore** before closing the form.

**Middle click** to continue.

The final result is shown below.



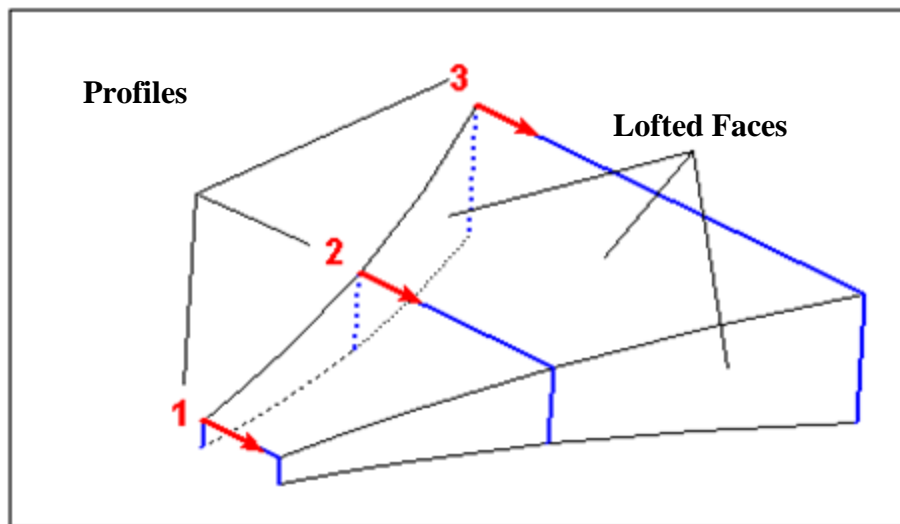
Do the **UNDO**  and **REDO**  commands a couple of times to see exactly what happened.

## Curve List and Lofts with Continuity

Lofts can be created as a lofted solid or as lofted faces. The same command is used; the only difference is an open or closed profile. For our next discussion we will concentrate on lofted faces.

### Lofte (Shape Tab)

This command creates a **shape** by lofting (blending) faces through a series of profiles. The profiles can be sketches, wireframe curves, curve lists, or face edges.



Create Lofted Faces

Select the **profiles** in the **order** you want to loft them making sure that the **arrows** are pointing in the **same** direction.

### Continuity

Use this command option to attach **continuity** constraints to mating edges during lofts.

This will enforce **continuity** at the start and end of the loft.

This option becomes active after the profiles have been selected.

### Continuity Level

Use **None** to enforce the position of the edge only.

Use **tangent** continuity to enforce the position of the edge and tangency to the face at the edge.

Use **curvature** continuity to enforce the position of the edge, tangency to the face at the edge, and the curvature across the face.

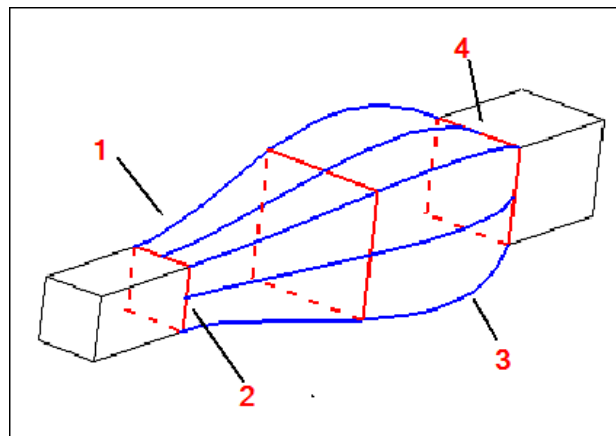
Light tends to reflect off the surfaces giving a more contiguous appearance.

## Continuity Weight

**Weight** - This determines the amount of the loft affected by the enforcement of tangency (i.e., how long before the loft returns to its normal curvature).

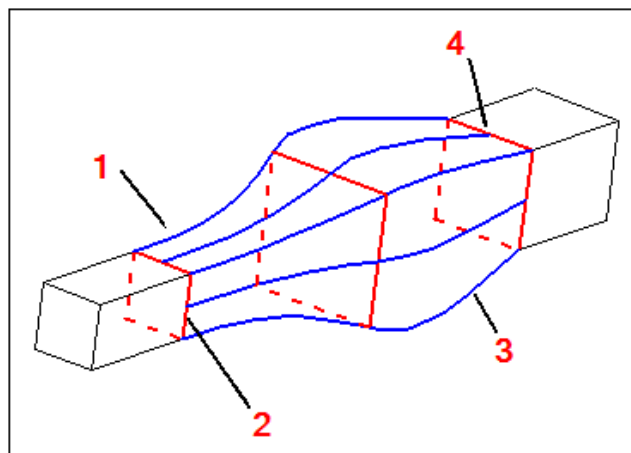
Move the slider bar between -1.0 and 1.0 to specify the continuity weight.

*Notice that as the slider bar moves the echo of the resulting loft is dynamically adjusted to the continuity magnitude.*



**Continuity with less Weight**

1: Less Weight 2: Start Continuity = Tangent 3: Loft Shape 4: End Continuity = None



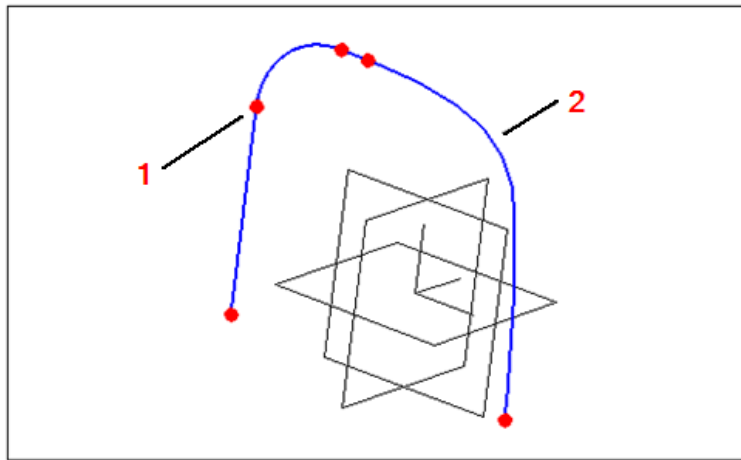
**Continuity with more Weight**

1: More Weight 2: Start Continuity = Tangent 3: Loft Shape 4: End Continuity = None



## Insert Curve List

Use this input option (right mouse button in the graphic area) to create a curve list from a selection of curves connected end to end. This allows multiple curves to be combined into a single selection for use during face features and base, boss, and cut solid feature commands. The curves are not physically combined (concatenated) or otherwise modified just grouped.

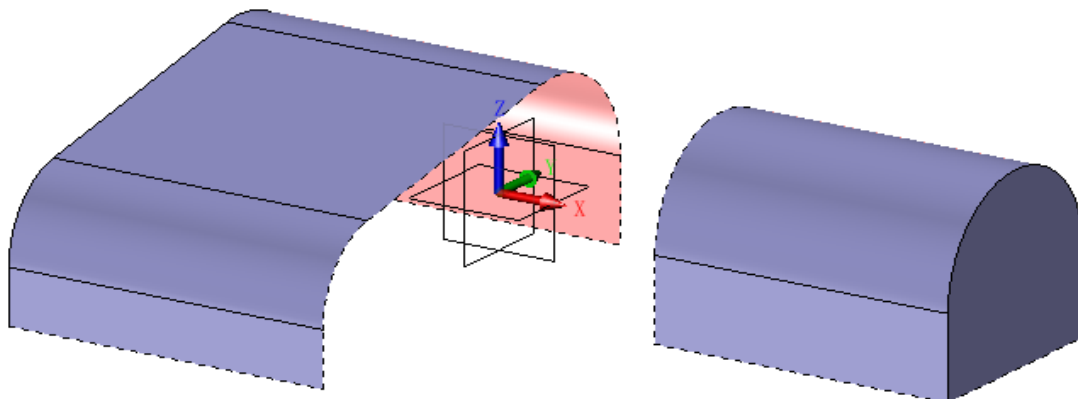


Make Curve List

1:Curves are connected End to End 2: Curve or Edge Segments

“Exit part” and Edit the part “**Curve\_list**”.

The part shown below consists of two open shape features. Step through the history to become familiar with the construction technique. We wish to create a **tangent loft** between them. We can create a loft feature between the two existing shapes, using the **Insert Curve List** option.



## Loft

Select the **loft**, command.

Select the **Add** option .

With the right mouse button, select the **insert curve list** option.

Pick the edges of the left hand shape.

The picked edges are shown with **blue** crosses on the bottom left image.

Press the middle mouse button to terminate selection of the **first** curve list.

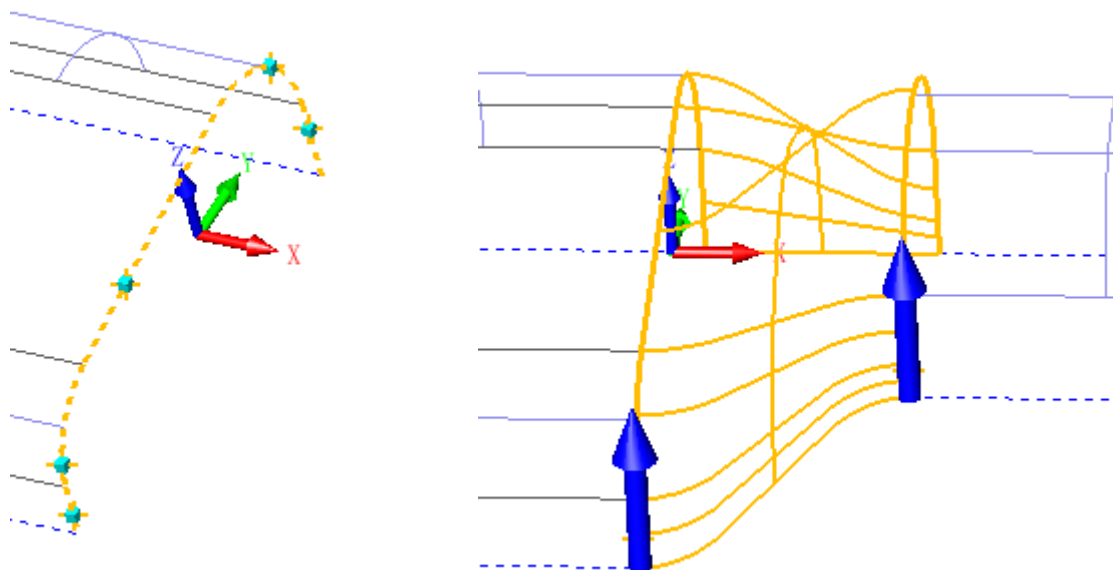
**Repeat** the process for the second curve list on the right hand shape.

Press the middle mouse button to terminate edge selection.

The result is shown below in the right image. Notice the arrows, this is the start of the loft and must be in the same direction on both curve lists.

The default continuity is **Tangent at Both Ends**.

**Do not complete the loft at this stage.**



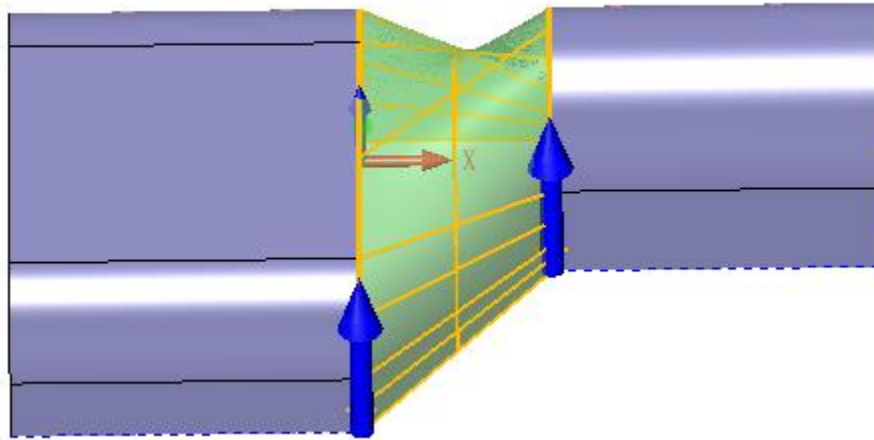
If the loft need not be tangent to the existing faces; we can apply **continuity**.

Select Continuity **None**.

**Make the Loft not tangent.**

The image below shows the graphical preview of the loft.

Notice that the loft is not **tangent** to the edges of the existing shapes.

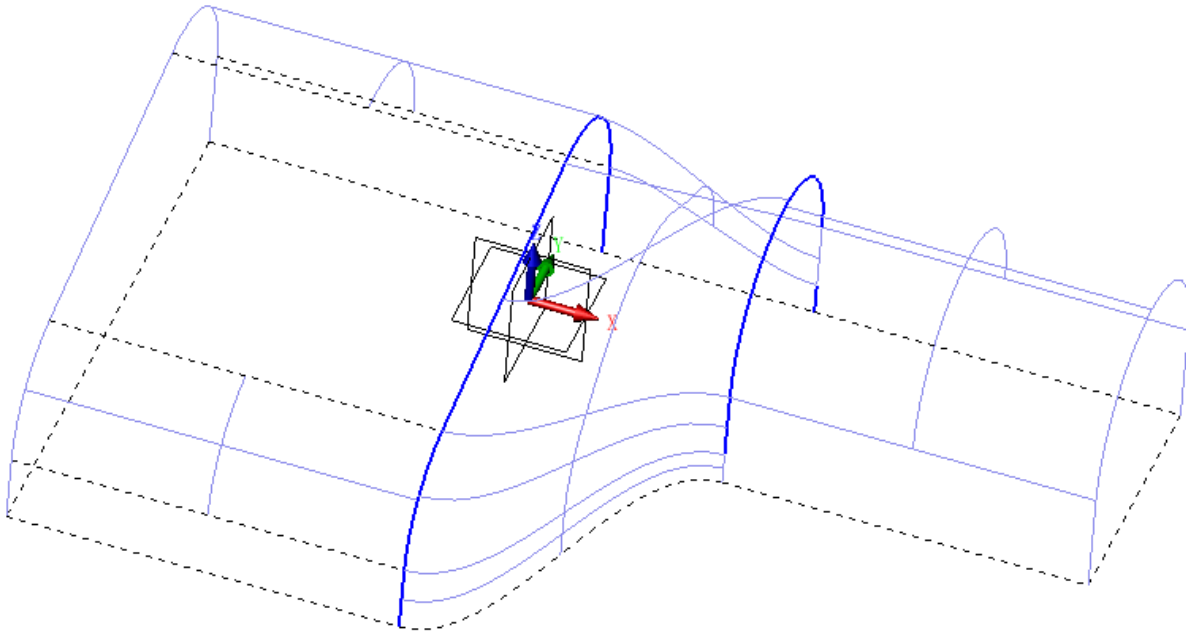


From the option form, select **continuity** at both ends **tangent**.

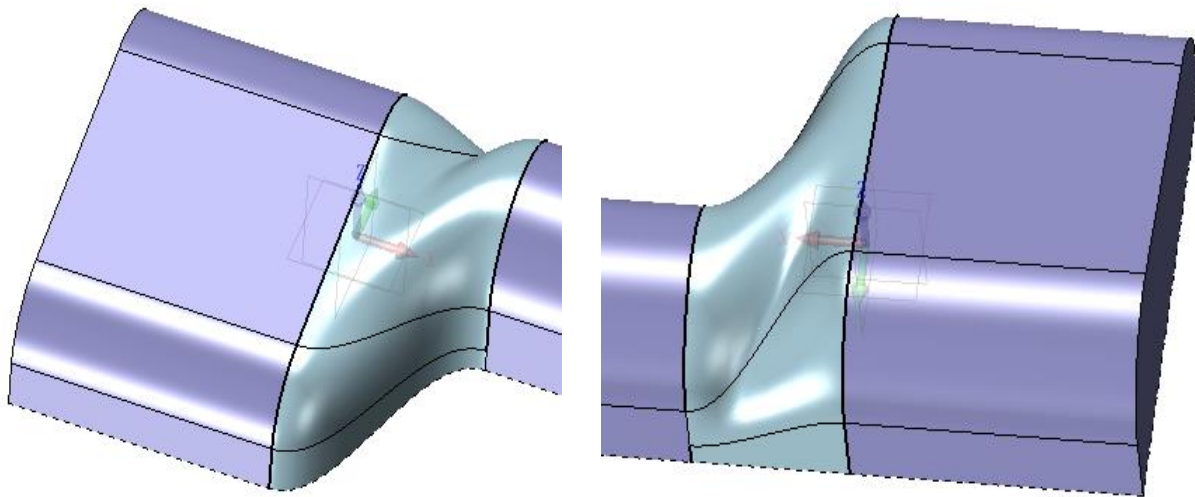
The preview will update.

The completed loft is shown on the next page.

## Make the Loft tangent

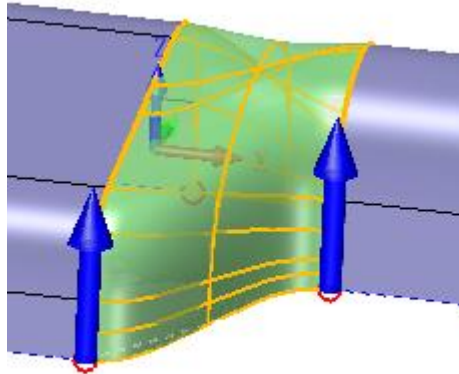


If you look closely at the wireframe view, we have 5 edges on the left matching with only 3 edges on the right. We have let ZW3D decide how to match the two profiles. This may or may not be what is really wanted. See the shaded view and rotate the image to the other side. We do indeed have a tangent condition at the edges but because of the uneven number of edges, twisting in the surface is unacceptable.



Right-click and select the **Redefine Last** command.

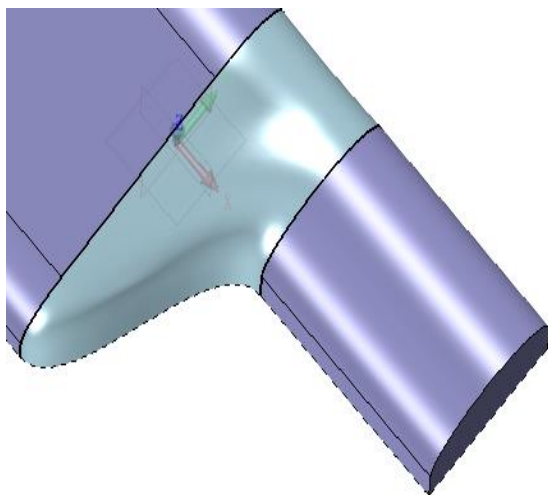
According to the default settings, the **Use Tangent vertices** option is not checked. Click on the **Auto** option under the **Shape control** tab. This will reset the display of the surface without using the tangent vertices. You now have a more even transition between the 5 edges on the left and the three edges on the right.



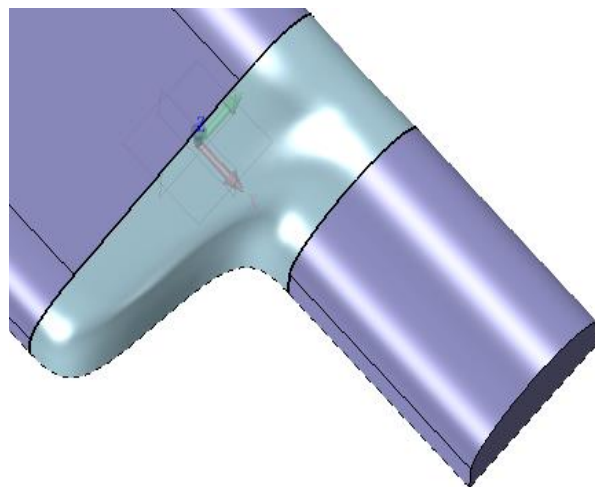
This looks better. There are now only two **connection lines**; one at either end of the profile.

Below the image shows the difference between selecting **Tangent** and **Curvature** continuity.

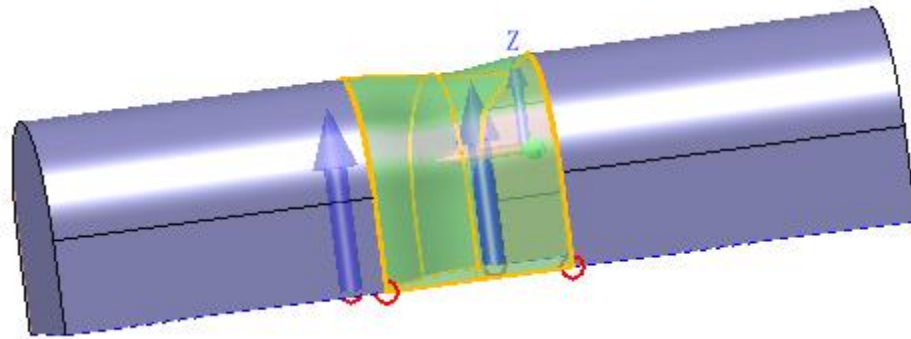
You decide which looks better.



Continuity=Tangent



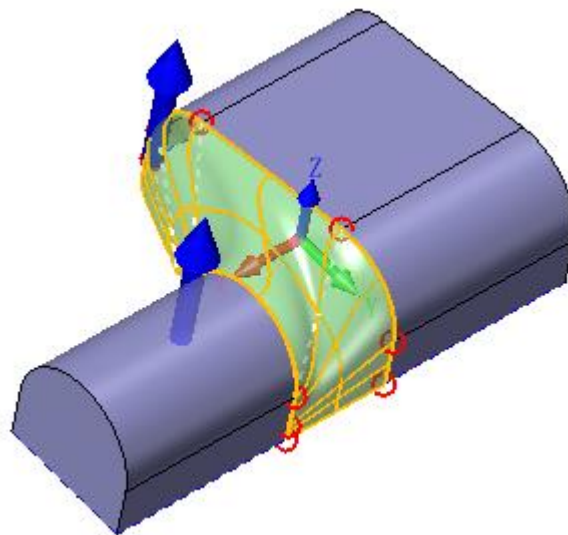
Continuity=Curvature



If you look closely from an angle similar to the one above you will still see a dip in the surface which may not be what you want.

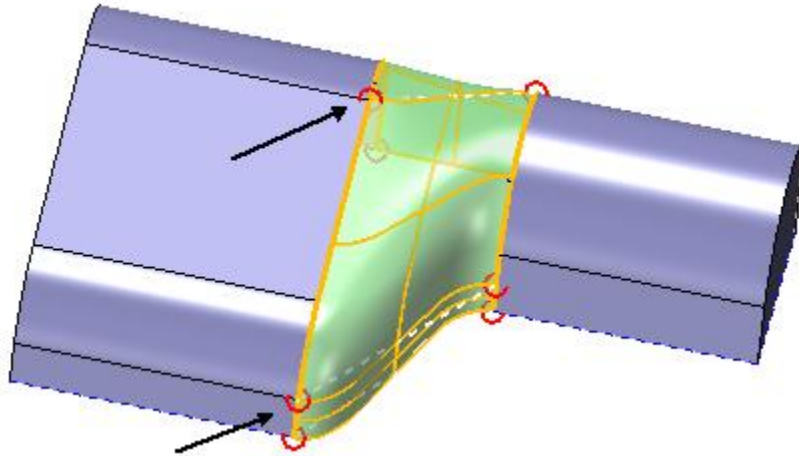
The following shows another technique that may be helpful.

Turn the **tangent vertices on** and click **Auto( Advanced tab)**.



Now **Delete** the middle 4 lines.

**Add** two more connection lines as shown right.

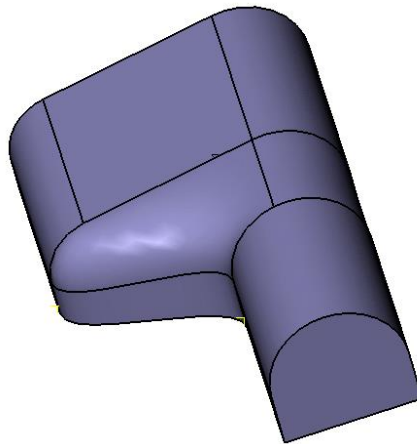


The first connection line keeps the transition between the vertical walls vertical.

Now **modify** the second one as shown.

The transition is now localized and the part is smoother.

Practice adding, deleting and modifying connection lines.

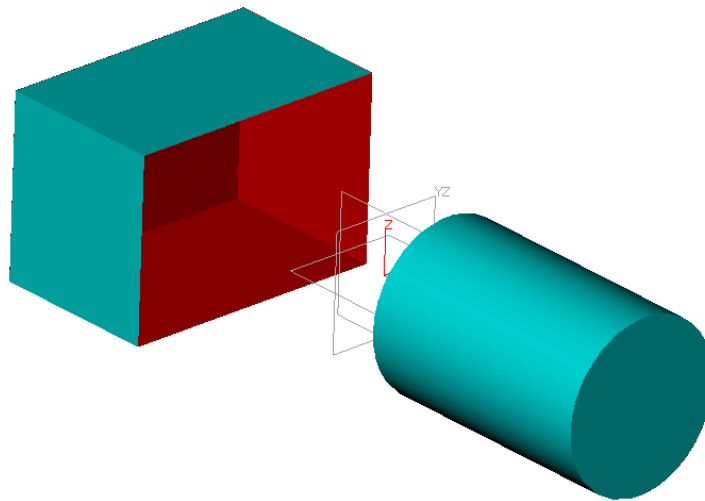


### ***Classic example of editing the loft connection lines***

Let's look at another example of creating a lofted surface; this time on a closed boundary where user interaction will give a more desirable result.

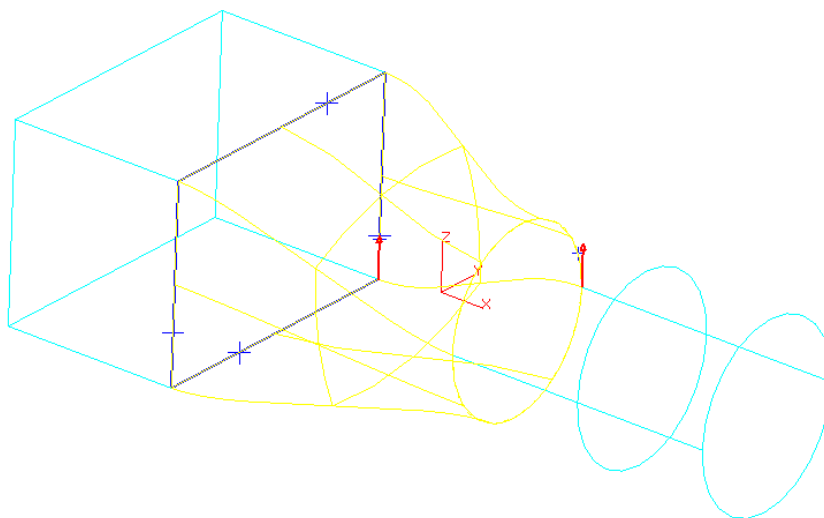
#### **Create Lofted Faces**

"Exit part" and Edit the part "**Loft\_with\_Connection\_lines**".



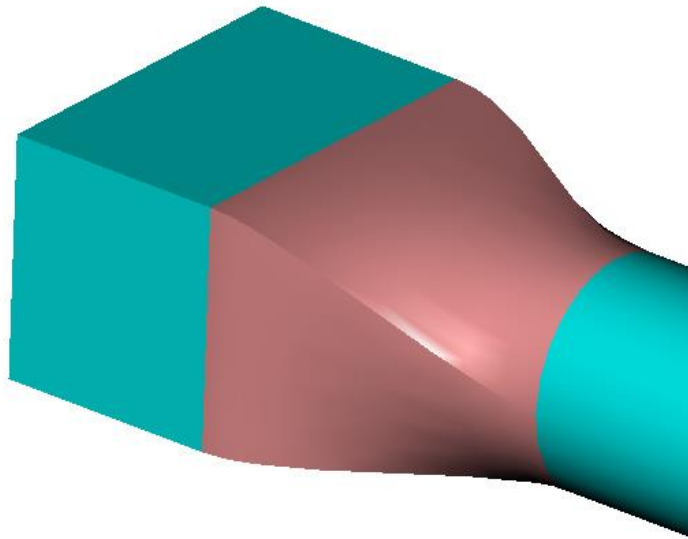
Create a **Loft** using the **Insert curve list** option. You only need one curve list this time. See the image below for the **start point**.

Pick the circle so that the arrow is pointing in the same direction as the rectangle.





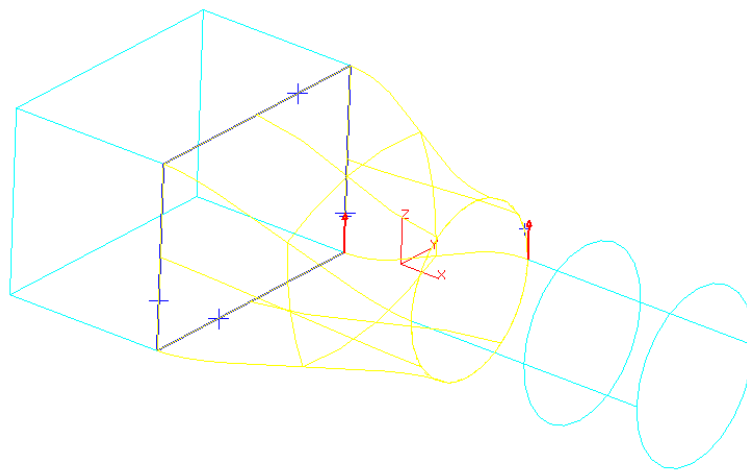
Just like the previous example, the loft has a twist that is caused by the difference in entity count from sketch1 to sketch2.



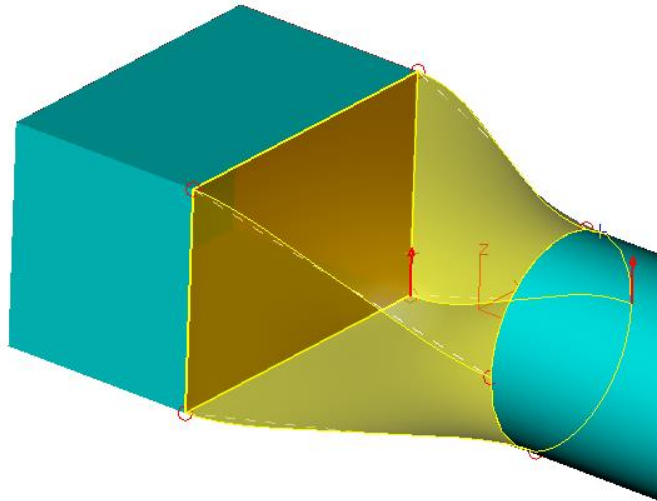
### Redefine last.

This part has no tangent edges so it won't matter if the option box is checked or not.

Select the **Auto** button to display the connection lines.



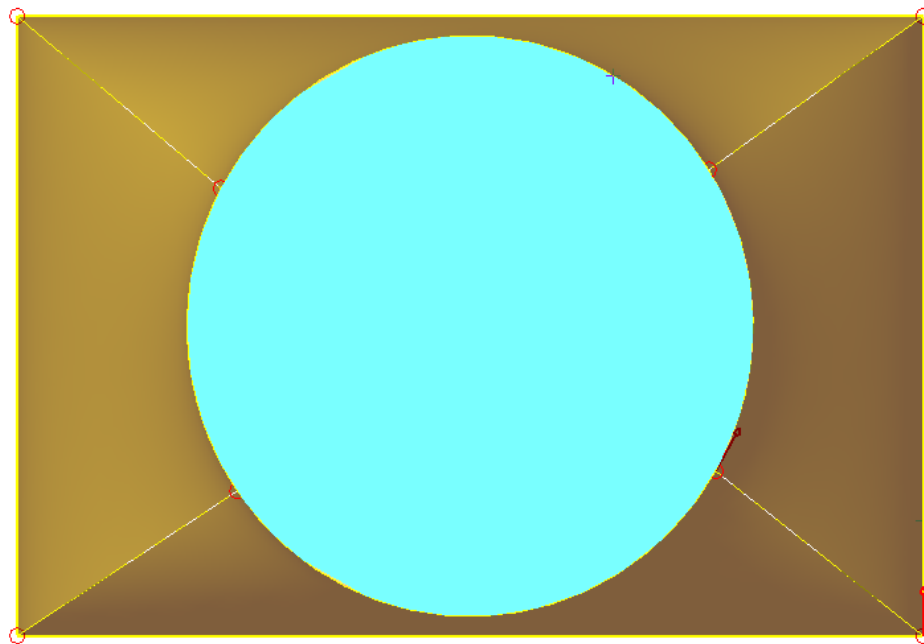
Use **Dynamic Preview** and rotate the part to see the connections.



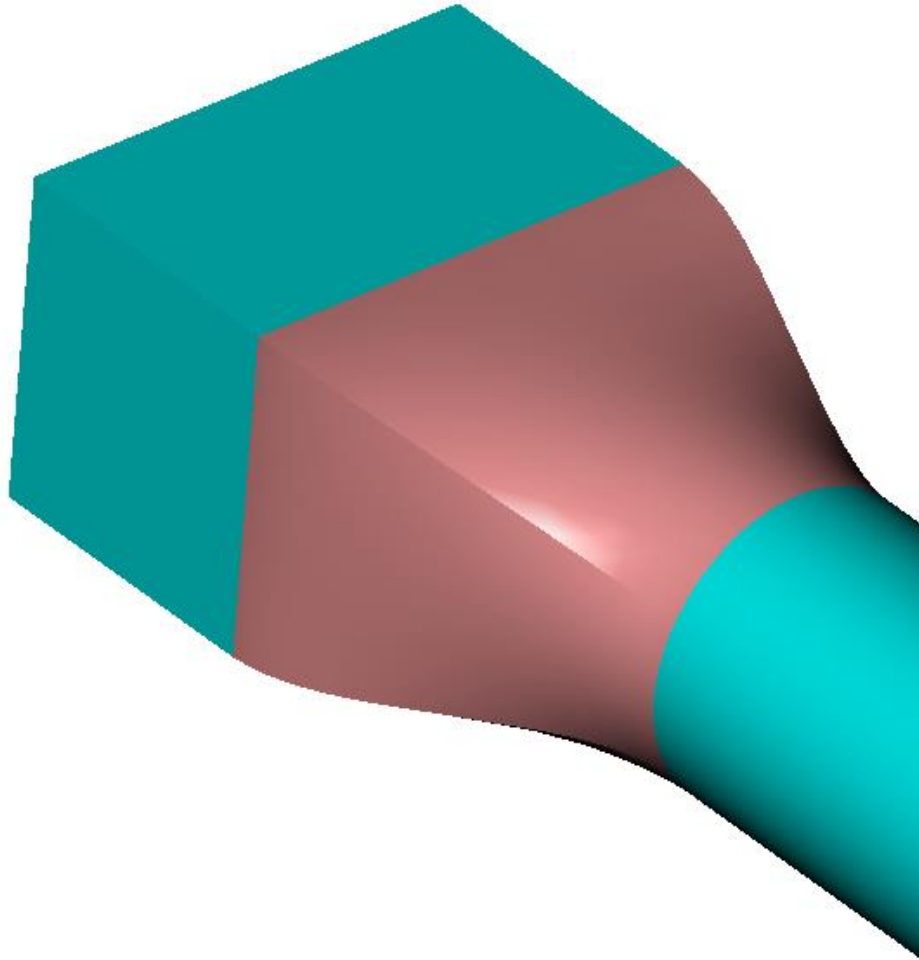
Select the **Modify** button.

Pick a connection line and relocate the end attached to the circle like the image on the right.

**Repeat** for the other three.



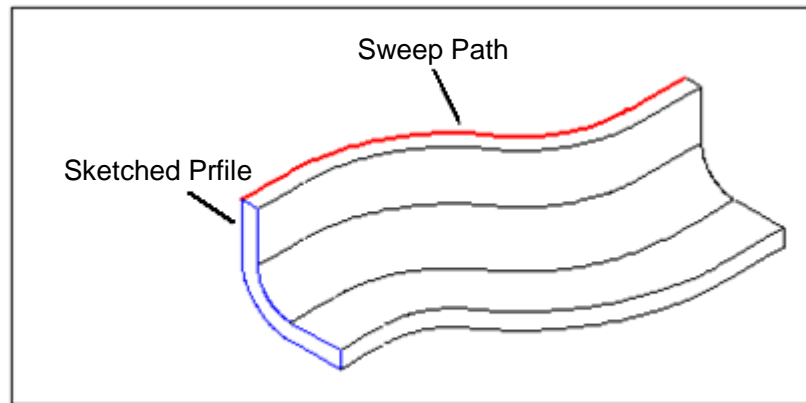
Result :



## **Chapter 3    Sweeps**

The following refers to **Solid** sweeps  (Shape Tab). Surface sweeps behave in a similar manner.

Use this command to create a solid base feature by sweeping an **open** or **closed** sketched profile along a path. **Open** profiles require the use of the **Offset** command option. The sweep path can be a curve or chain of curves.



Create Swept Solid

## COMMAND OPTIONS

The sweep command can be controlled by a combination of several command options.

The current command options available are:

**Boolean types:** **Base, Add, Remove, Intersect**

**Offset**

**Endcaps**

**Frame**

**Z-Spine**

**X-Orient**

**Scale**

**Twist**

### Offset

Use this command option to specify an offset method and distance(s) to be applied to curves, curve lists, or open or closed sketched profiles. This option adds thickness to the feature automatically.

### Endcaps

**Both** Connected perimeter faces are created and endcaps are placed at both ends of the shape feature.

**Start** An endcap is placed at the start of the shape.

- End** An endcap is placed at the end of the shape.  
**None** The resultant shape is left open at both ends.

## Z-Spine

Use this command option to control the Z axis direction of a sketch or objects during sweep and move commands.

**Natural** - The Z-axis remains tangent to the sweep curve. No additional steps are required.

**Spine** - The Z-axis remains tangent to a spine curve. The spine curve can be any existing curve. If it is the sweep curve, the result will be the same as the **Natural** option.

**Parallel** - The Z-axis remains perpendicular to an existing plane. This forces the profile plane to remain parallel with the existing plane.

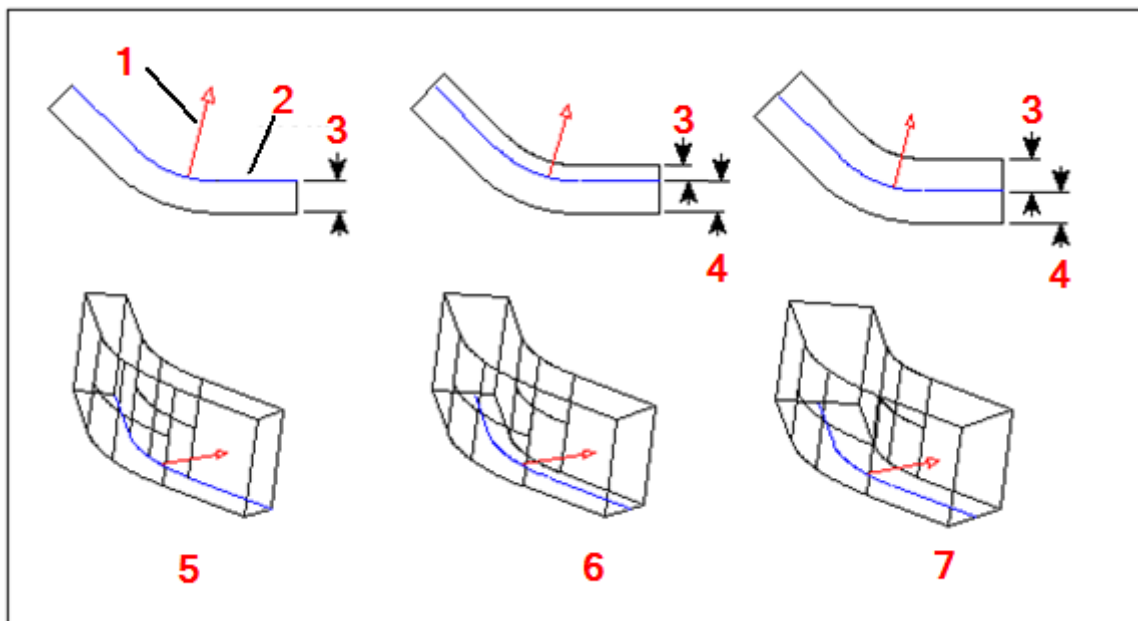
## X-Orient

Use this command option to control the orientation of the X-axis of a sketch or objects during sweep and move commands.

**Natural** - The X-axis is constrained to minimum rotation.

**Guide Plane** - The X-axis stays "in" an existing plane. It is perpendicular to the plane normal.

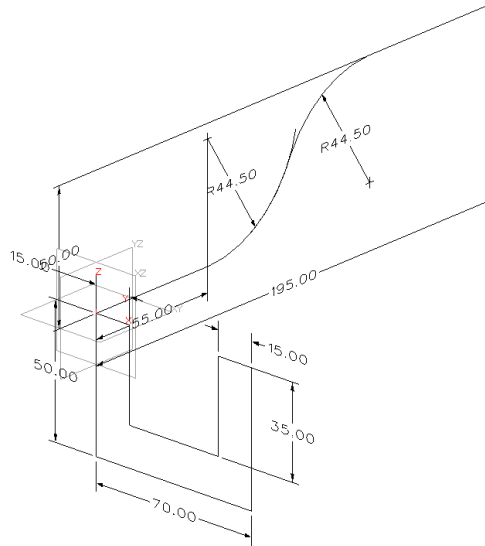
**X-axis Curve** - The X-axis is projected and follows the intersection of an existing curve.



1:Inside 2:Open Profile 3: Dist 1 4: Dist 2 5: Shrink/Expand 6: Thicken 7: Thicken Same

## Command option examples 1.....Z-Spine

Clear the current session and edit the part “**Z-Spine**”.  
We will sweep the **U shape** profile along the curved sketch.



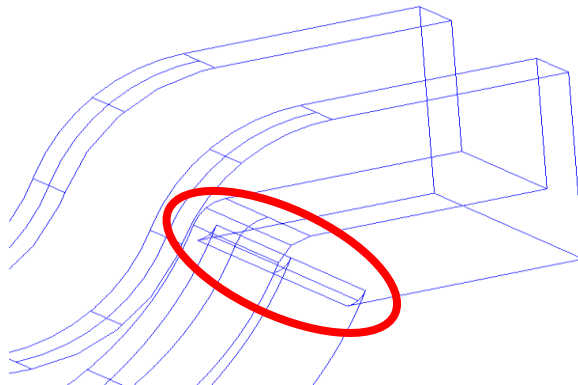
Pick the standard sweep command.

Pick the **U shape** sketch.

Pick the **curved** sketch as the **sweep path**.

Accept the **defaults** for the rest of the prompts.

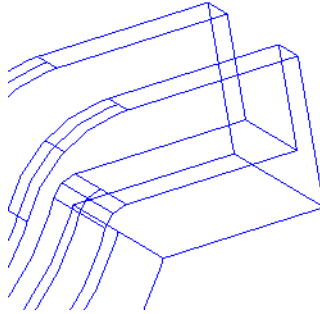
The profile is swept **perpendicular** to the sweep path, and self-intersection problems occur as shown opposite. The reason is the height of the profile is greater than the radius.



Pick the **Remove self intersections** command  from the Shape Tab.

Pick the part and select **OK**.

**UNDO** the last two operations and let's look at another solution.



Using the Z-spine option can prevent the problem shown in the last example.

Pick the **Sweep** command as before.

Pick the **U shape** sketch.

Pick the **curved** sketch as the **sweep path**.

Since the profile sketch is built in the correct position we don't have to change the **FRAME** from the default **At Profile**.

Pick **Z-Spine – Parallel**.

With the right mouse button select **Y-axis** for the normal of parallel plane.

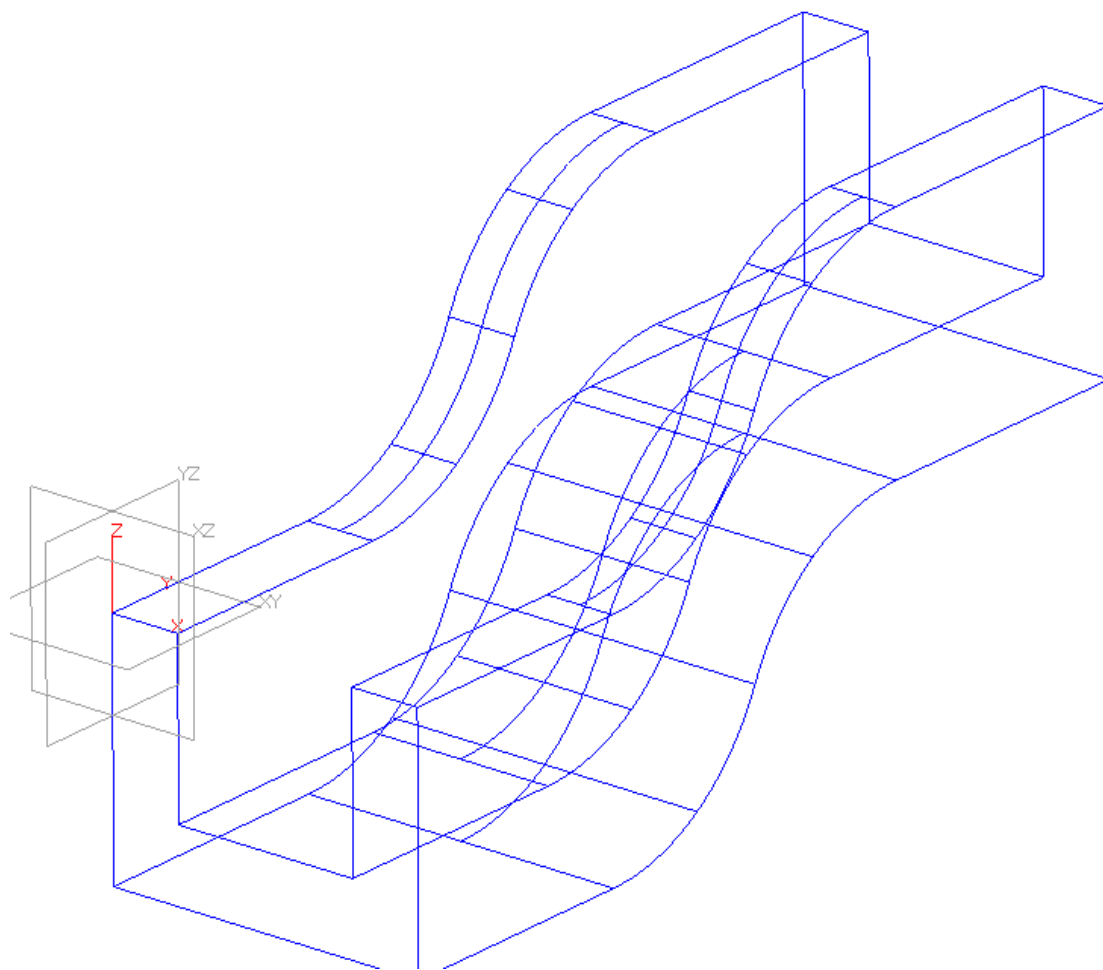
Accept the defaults for the rest of the command.

The resulting sweep without the self-intersection problem is shown below.

Any section through the part created perpendicular to the Y axis will create identical cross sections.

Both solutions are correct. It depends on your design intent.

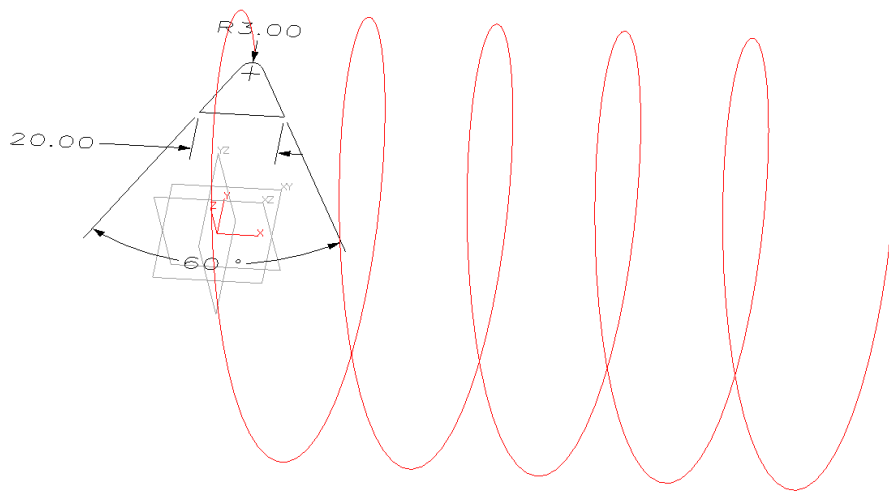




## Command option examples 2..... X-Orient

“Exit part” and edit the part “**X\_orient**” as shown below.

As may happen from time to time, the background color may be the same as the sketch or dimensions. If you can’t see the triangular sketch, change the background color in your configuration so you can see everything clearly. Try gray.



Pick the **sweep** command.

Pick the **triangular** shaped sketch.

Pick the **spiral** curve near the triangular sketch as the sweep path.

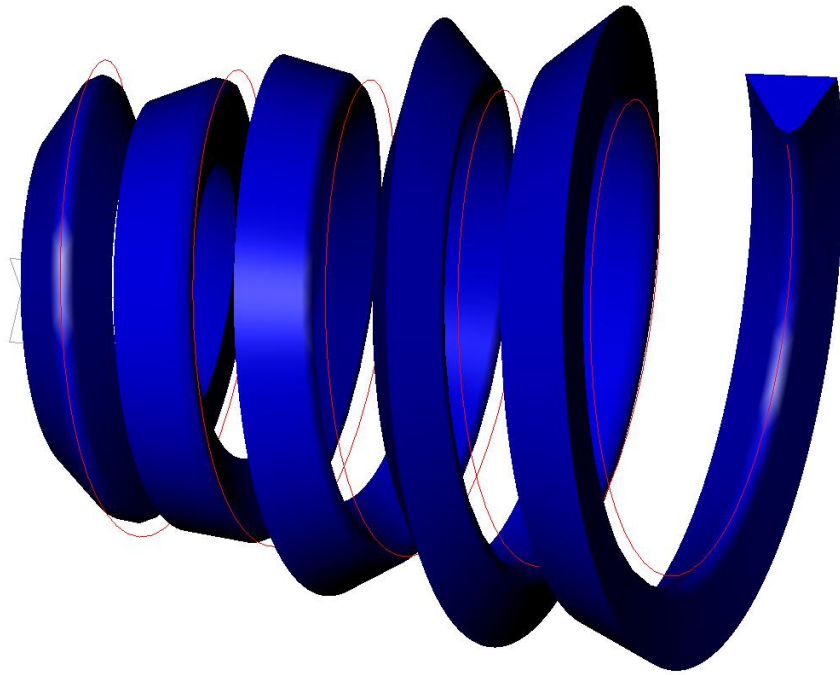
Pick **At Path** as the sweep **frame** method.

Accept the defaults for the rest of the command.

The triangular profile is swept **normal** to the drive curve.

The resulting solid twists as it goes along the curve producing the shape shown opposite.

**UNDO** the last operation.



Using the X-Orient option can prevent the problem shown in the last example.

Pick the **Sweep** command as before.

Pick the **triangular** sketch.

Pick the **spiral** curve as the sweep path.

Pick **At Profile** as the sweep **frame** method.

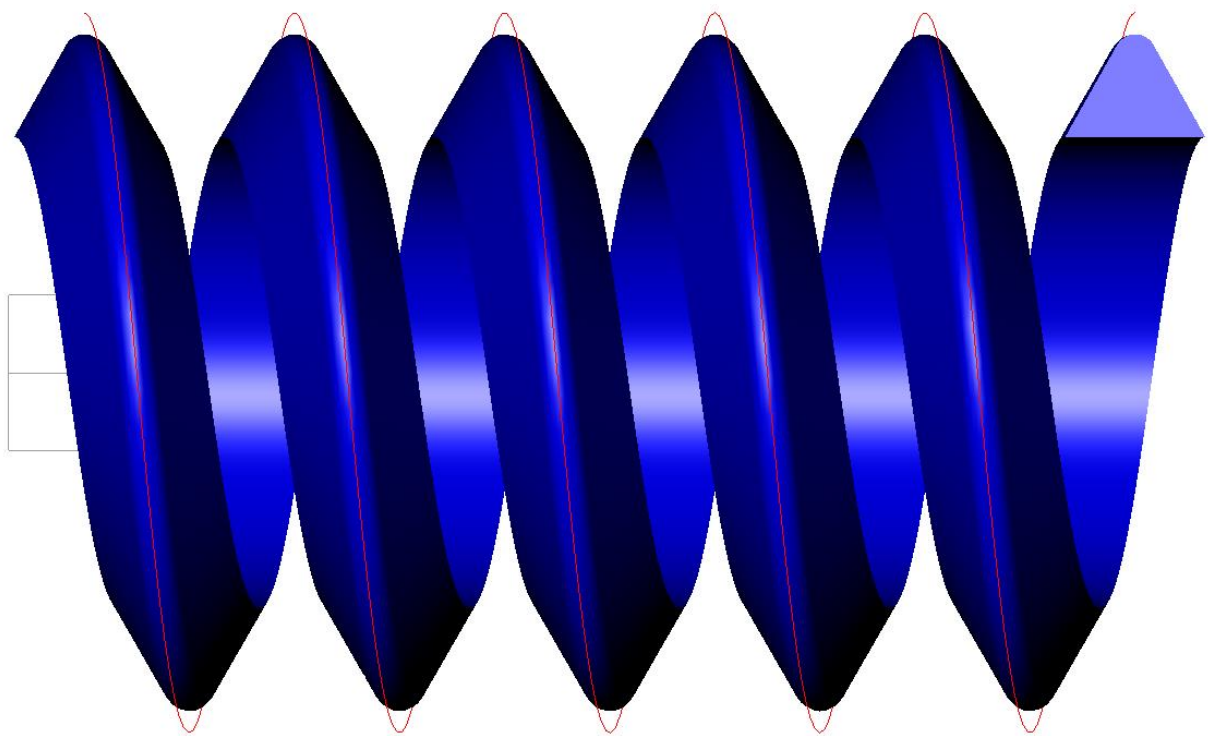
Pick the **X-Orient** - guide plane option button.

With the right mouse button select **X-axis** as the guide plane normal.

Accept the defaults for the rest of the command.

The **X-axis** of the sketch is now kept parallel to the guide plane axis.

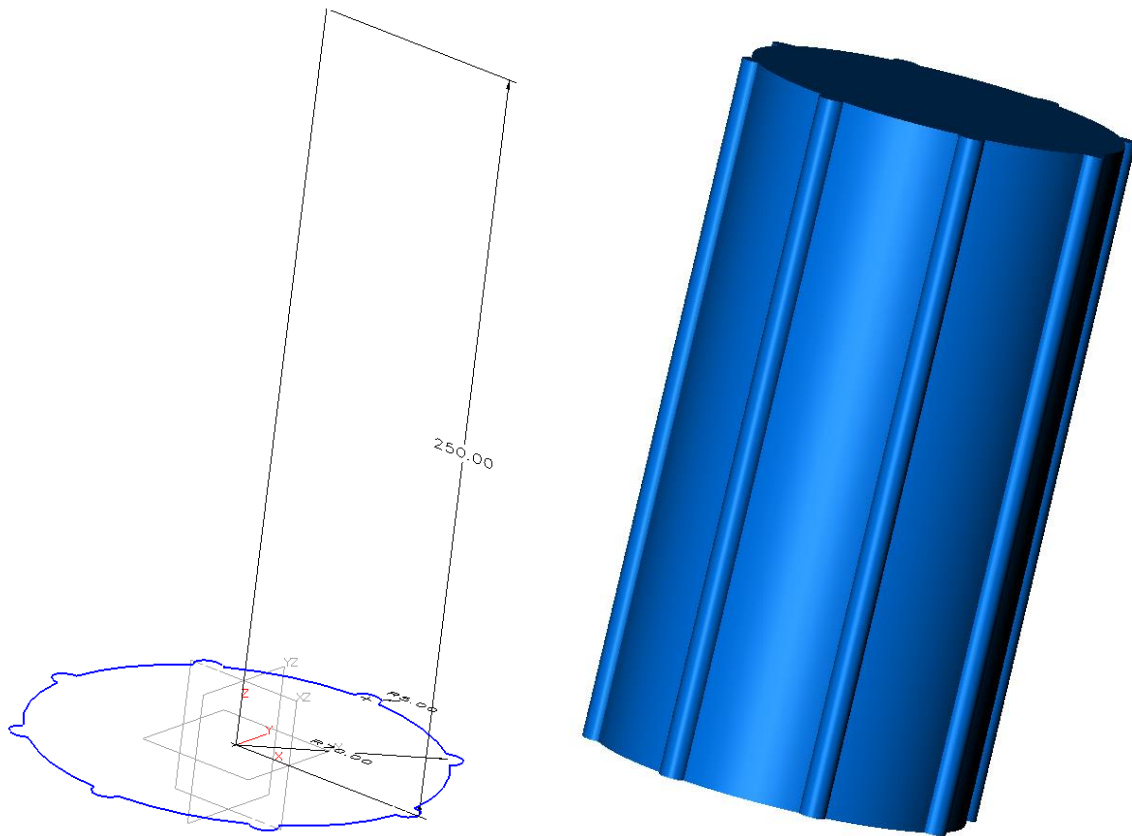
The sketch profile does not twist as it follows the drive curve as shown below.



## Variable Sweep

Clear the current session and edit the part “**Variable\_sweep\_1**”.

We will sweep the blue trace profile sketch along the vertical curve. The sweep path can also be curved if desired, but for the examples that follow we will use a straight curve.



Pick the **Sweep** command.

Pick the blue trace profile sketch.

Pick the straight line as the sweep path.

Accept the defaults for the rest of the command.

The part is as shown above.

**UNDO** the last operation.

## Applying linear scale and linear twist along the sweep

Select the **Sweep** command.

Pick the **blue** trace profile.

Pick the **Straight** line as the sweep path.

Use the default **At Profile** as the sweep **frame** method.

Select **Scale – linear**.

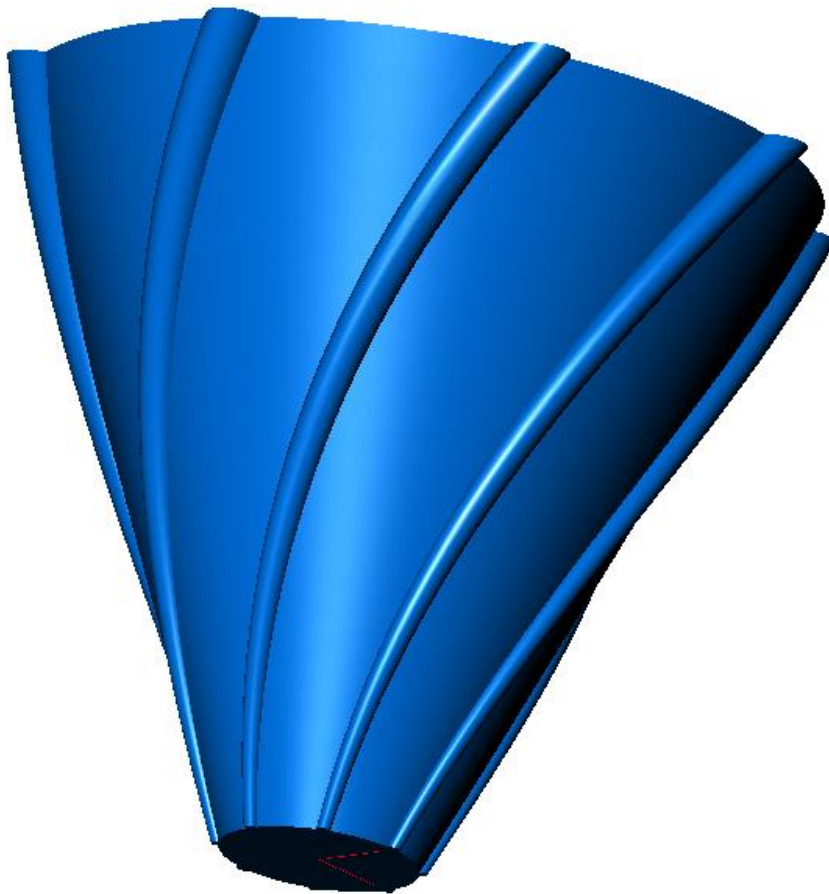
Start **0.5** end **2.0**.

Select **twist – linear**.

Start **0.0** end **60.0**.

The part is shown below.

**UNDO** the last operation.



## Applying variable scale, linear twist and using offset thicken

Select the **sweep** command again.

Pick the **blue** trace profile.

Pick the **Straight** line as the sweep path.

Select **Offset - Thicken**.

Enter **-3** for the value.

Select **Scale – Variable**.

Pick the **add** button.

With the right mouse button choose **critical**.

Pick the start of the sweep curve (**at 0,0,0**).

Enter **1.0** as the scale.

Pick the opposite end of the sweep curve.

Enter **1.2** as the scale.

With the right mouse button choose **on entity**.

Pick a point on the sweep path about  $\frac{1}{4}$  up from the bottom.

Enter **0.8** for the scale.

Pick a point about **half way** up the sweep path.

Enter **1.6** for the scale.

Pick a point about  $\frac{3}{4}$  up the sweep path.

Enter **0.8** as the scale.

Press the middle mouse button to accept.

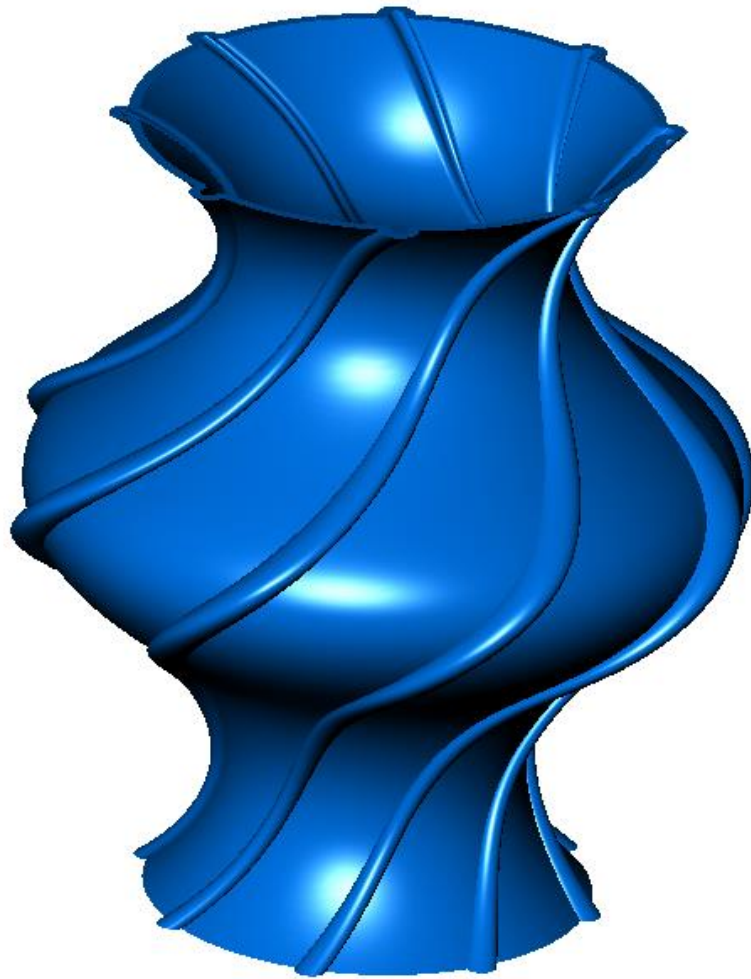
We forgot, no, you forgot to add twist.

**Right click** Redefine Last.

Select **Twist – Linear**.

Start **0.0** end **150**.

See the result below:





## Applying variable scale and the Flat option

**UNDO** the last operation.

Continue to use the **UNDO** command until all of the scale attributes have been removed from the part.

As before sweep the **blue** profile along the **straight** line.

Use the **variable** scale option and **Add** a scale factor of **1.0** at the start of the line.

Apply a scale factor of **0.5** at the **middle** the line.

Finally apply a scale of **0.5** at the **end** of the line. **Middle click** twice.

Notice the result (shown below).

The scale of **0.5** does not stay constant (it gets smaller between the two .5 scale attributes).

We can control this **locally**, using the **Flat** option.



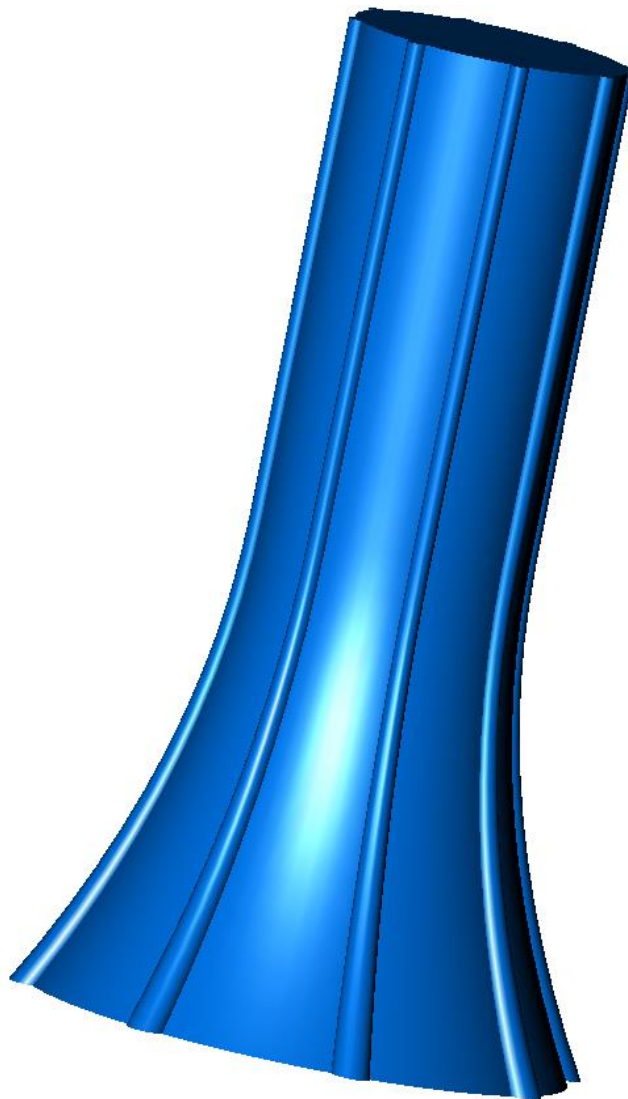
Apply the flat option to keep the scale constant

**UNDO** the last operation.

Continue to use the **UNDO** until all 3 scale attribute have been removed from the history. Sweep again using the **variable scale** option and apply a scale factor of 1 at the start. Next check the “**make scale locally flat**” box. Apply a scale factor of **0.5** at the center of line.

<Middle-Click> for the defaults for the rest of the command.

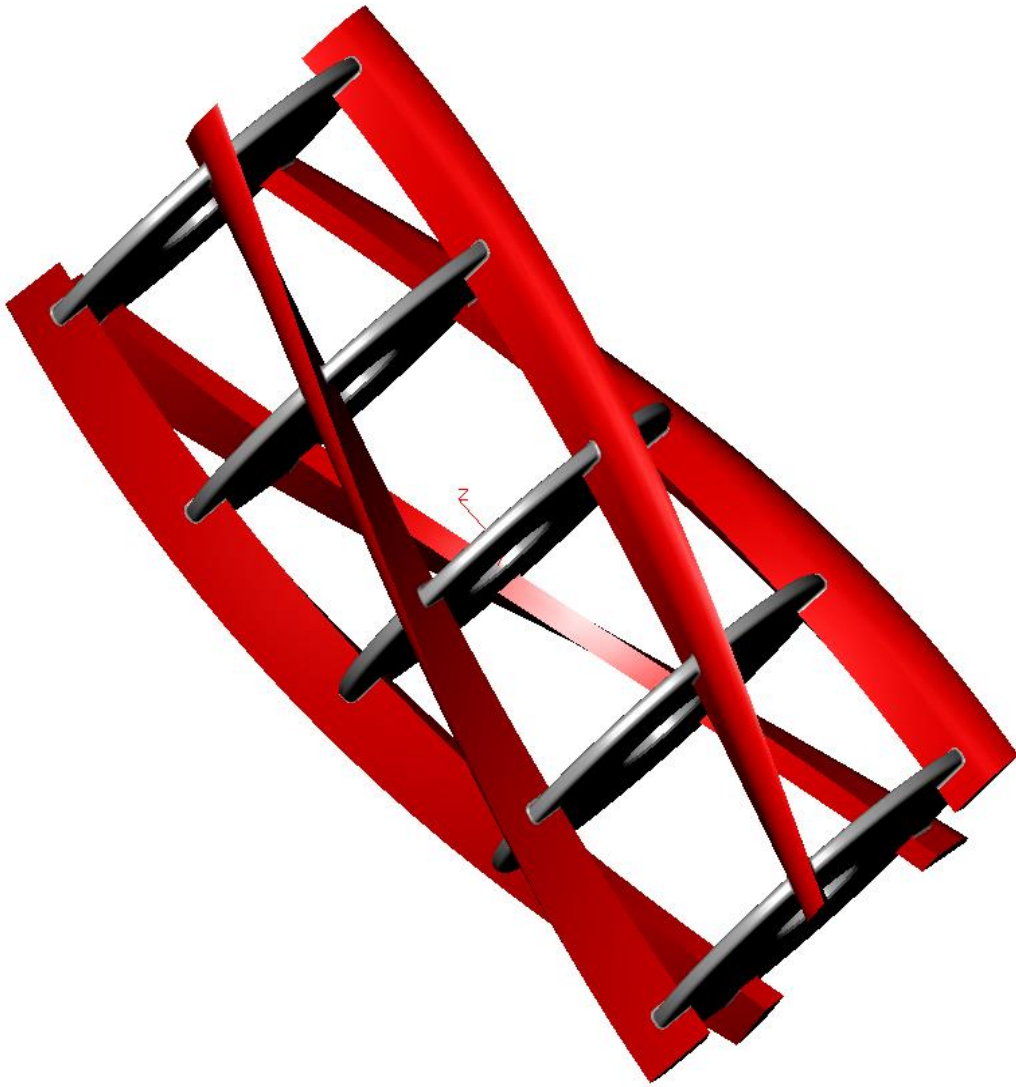
The **0.5** scale now stays “**locally flat**” along the curve. (or, until the next attribute is applied)



Another example of the sweep command with twist.

Clear the current session and edit the part “**Blades\_comp**”.

Step through the history to see how the part below was constructed.



## Variational Sweep

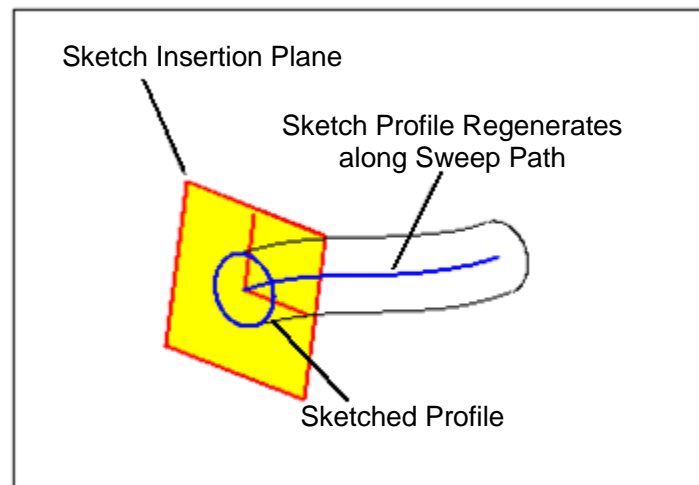


Use this command to create a base shape feature by sweeping a sketched profile along a path. The sweep path can be a curve or chain of curves. It is called a base shape feature because it can be used as the first feature in a new part or component. With a **variational** sweep, the profile will be **regenerated** as it is **swept** along the path. This will give the same results as the **Sweep** command **unless** the profile contains an **external dependency**. Currently, there are two ways to set up the external dependency.

Reference the variable **[~VxPathParam]** in an equation that drives a sketch dimension. This variable will be **0.0** at the **beginning** of the sweep and **1.0** at the **end** of the sweep.

Reference external geometry that will change during the sweep (e.g. an intersection with a curve) and constrain the sketch to the external geometry. Or we can use a combination of both.

The **sweep frame** method is forced to always be **in place**. **Scale** and **twist** command options are **not** supported since they can be in conflict with the variational solve.

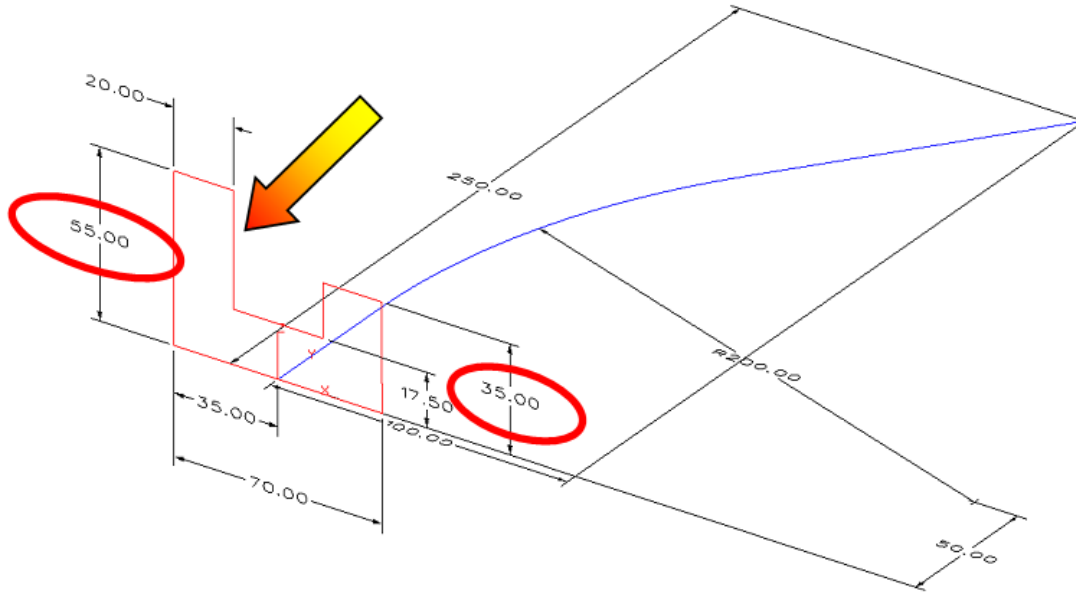


Create Variational Swept Solid

Create a **Variational Sweep** using the external dependency **[~VxPathParam]**.

Clear the current session and edit the part **"Variational\_sweep1"**.

The part shown below contains **two** sketches.



Notice the **55mm** and **35mm** vertical dimensions in the red sketch.

If the red sketch was to be swept along the blue sketch the 55 and 35mm dimensions would clearly not change.

What if we need the **55mm** dimension to get **smaller** and the **35mm** dimension to get **bigger** as they move along the sweep path? Let's say we want the **55mm** dimension to start at **55** and end at **35mm** and the **35mm** dimension to start at **35mm** and finish at **55mm**.

We can do this by using the variable [**~VxPathParam**].

## Edit the red sketch.

Double click on the **red** sketch. This will enter the sketcher.

We will create two variables to control the **55 and 35mm** dimensions.

Sketches must be fully constrained, remember they will change as they move along the sweep path and need to continually update. The example sketch is already fully constrained for you.

Select **Edit** pull down **Edit > Edit equation**, to enter the **equation** editor.

Create two variables, one called **leftside**, and one called **rightside**.

The **leftside** variable equation is as follows.

$$-20*[\sim VxPathParam]+55$$

Click “**Submit Equation Input**” and ‘**APPLY**’.

The **VxPathParam** system variable must have the **tilde “~”** prefix, and is also **case sensitive**.

The **20** value is the **difference** between the start and end values, **(55-35)**.

So at the start of the sweep, **-20\*0+55** equals **55**.

The end of the sweep will be **-20\*1+55**, which equals **35**.

Repeat the above method and create the **rightside** variable.

$$20*[\sim VxPathParam]+35$$

The start will be **20\*0+35** equals **35**.

The end will be **20\*1+35** equals **55**.

**Assign the variables to the 55 and 35mm dimensions**

**Edit > Dimension > Link to Variable**, or **Right Click** over each dimension and select **Link to Variable**.

Link the **leftside** variable to the **55mm** dimension.

Link the **rightside** dimension to the **35mm** dimension.



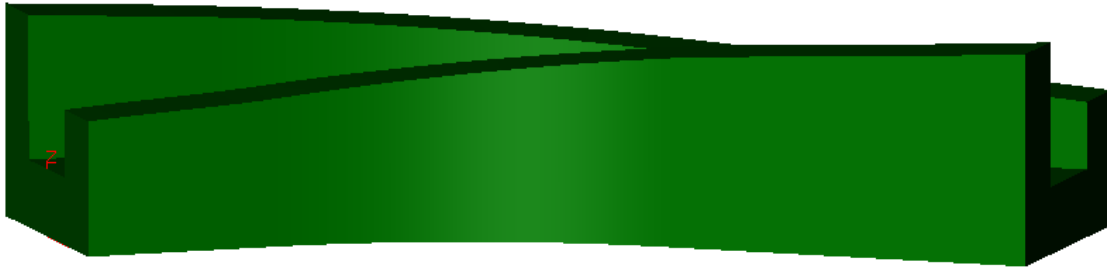
Click ‘**Solve Current Sketch Automatically**’.

Exit the sketcher.

Use **Variational Sweep**.

Sweep the **red** sketch along the **blue** sketch.

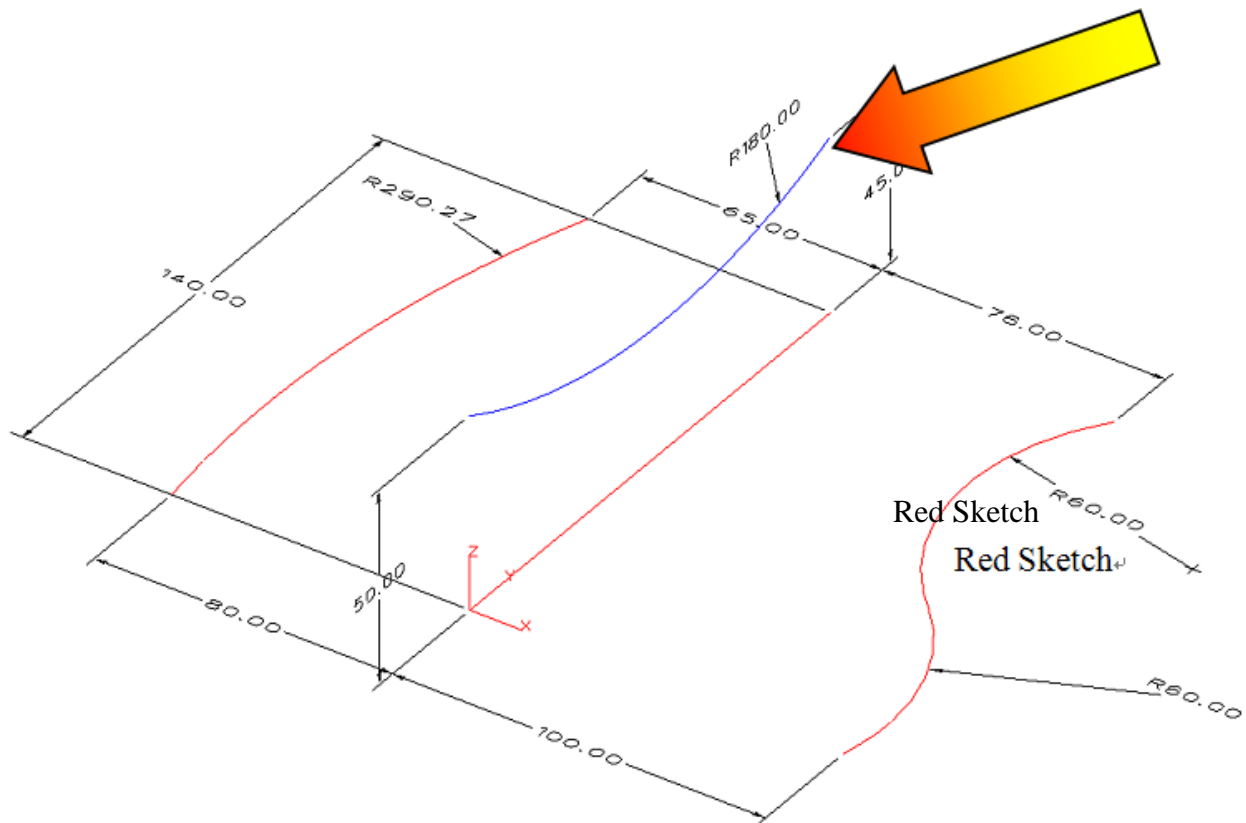
The result is shown below.



## VARIATIONAL SWEEP EXTERNAL DEPENDENCY, GEOMETRY

Create a **Variational Sweep** referencing geometry that will change during the sweep.

Clear the current session and edit the part “**Variational\_sweep2**”.



The part contains **two** sketches, we will create a **third** sketch on the **XZ-plane**. This new sketch will be the **sweep profile**; the **sweep path** will be the **straight line** in the **red** sketch. The new sketch will also be fully constrained with dimensions and with reference to the existing sketches (curves).

First we have to create a **concatenated** curve. Notice the **red** sketch, the right hand shape is constructed from **two 60mm radii**. For reasons that will later become clear, we

need to merge the two radii together to make one curve. We could have done this in the sketcher, but then we would lose the possibility of changing the **60mm** dimensions, as the **two arcs** would be replaced by a concatenated **curve** (in the sketcher). Instead we will create the concatenated curve in the **Modeler**. This will create an **additional** curve; the sketch arcs still exist. The advantage here is that the user can edit the arc dimensions, which of course edits the sketch, and because the modeler's concatenated curve is associative, a regen will also update the concatenated curve.



### Concatenate curves (in Wireframe Tab).

Pick the two **60mm arcs** shown above.

### Creating the third sketch.

With the **right mouse button** select **insert sketch**, pick the **XZ-plane** as the sketch plane.

Accept the rest of the defaults to enter the sketcher.

The system is probably set up to flip the sketch plane parallel to the screen.

Press **Ctrl\_U** to activate an auxiliary view.

We need to control the sketch such that when it moves along the sweep path, the existing curves will influence the shape.

In order to do this we first have to create reference geometry.

We will create three reference points by intersecting the sketch plane with three of the existing curves.



### REF WRONG.

Remember that the sketch changes as it move along the sweep path, influenced by existing geometry.

Therefore it would be **incorrect** to use the **project reference point** command.



### REF RIGHT.

**Instead** use the **curve intersection** command.(from the **Reference Geomertry** command)

This will intersect the picked curves with the sketch plane, and will continuously update the geometry as the sweep profile moves along the path. path.

Select the command and pick the **left red curve** at the **start**. (Pick Filter on **Curve**)

Repeat this for the **middle blue** curve.

Next we need to pick the **concatenated right curve**. Here we have a problem, if you place the mouse pointer over the concatenated curve, the **60mm arc** highlights, we do not want to pick the **arc**, as we want to control the sweep profile over the length of **both**

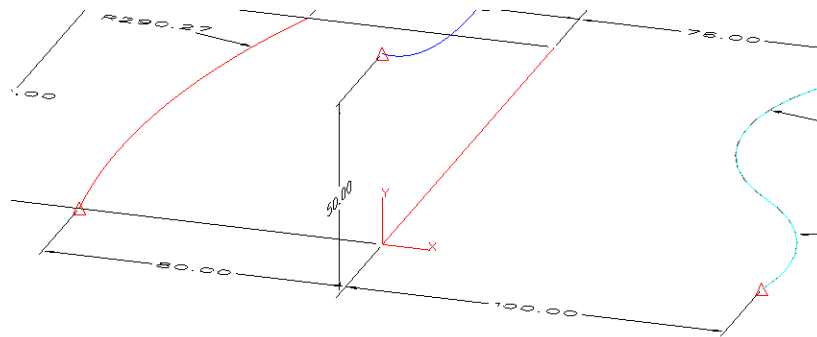


**arcs.** This is why we first created a **concatenated curve**. We must select the **curve** and not the arc.

To do this, move the mouse pointer away from the curve, then **press and hold down the Alt. Key**. Next move the pointer over the **arc** and the **concatenated curve** will highlight.

Pick the **highlighted concatenated curve**.

The three reference points are shown below as **three red triangles**.



Change the view to a **plane view**.

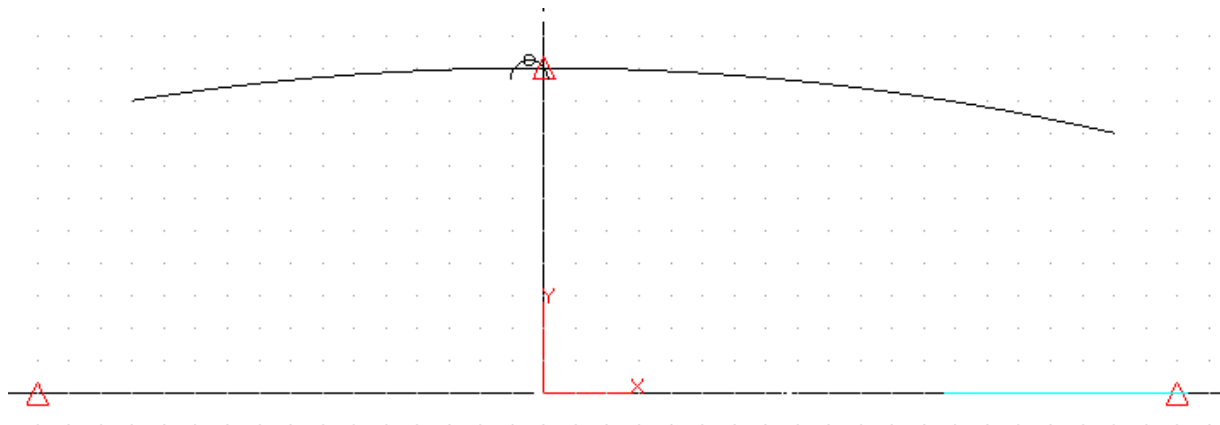
Using the right mouse button, select **Draw**.

Create the **arc** shown below.

Start point **X90** and **Y40**.

End point **X-65** and **Y45**.

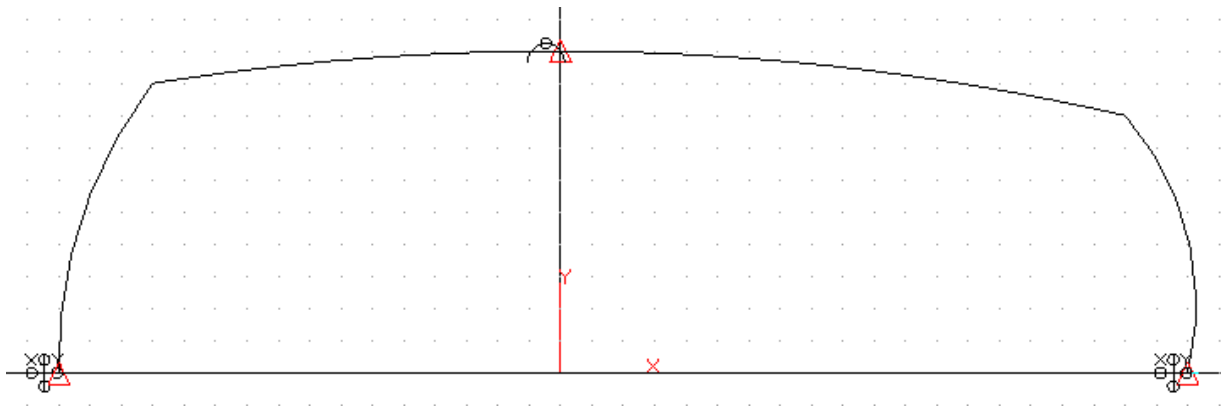
The middle point of the arc must go through the **middle reference point**, ensure you have the **constrain point to curve** constraint attached as shown below.



Next create a **line** between the **two lower** reference points.

Then using quick **draw**, create **two arcs**, between the **lower reference points**, and the **end points** of the large **arc**. As shown below.

**Note please use fairly large radii values or the subsequent center point constraints will not produce the required results. The next page contains a more detailed explanation of the potential problem.**



Next we will constrain the **center points** of the **two small arcs** to be aligned in **Y** with **one** of the **lower reference points**. If the small arc values are **not** large enough the center points will be located on the horizontal line correctly but the arc(s) will “**flip**” to the wrong side. Ask your instructor for a more detailed explanation if you are unsure what we mean.

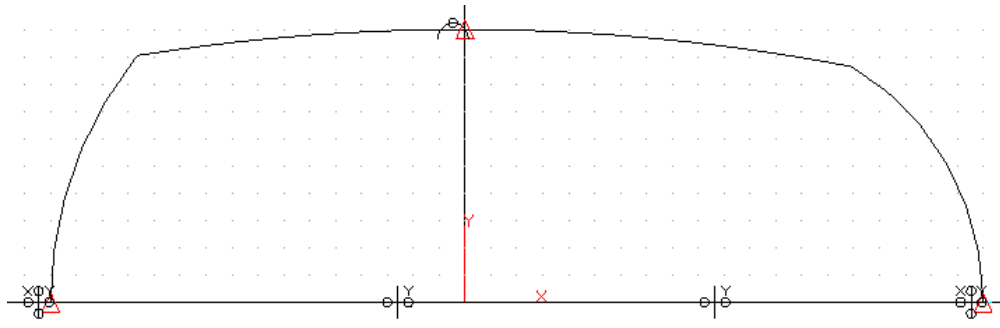
Pick **Points horizontal**. 

Pick the **lower right hand reference** point as the base point for aligning **Y**.

Place the mouse pointer over **both small arcs in turn**, picking the **center** points of **both** arcs, as the points to **align in Y** with the **right lower reference** point.

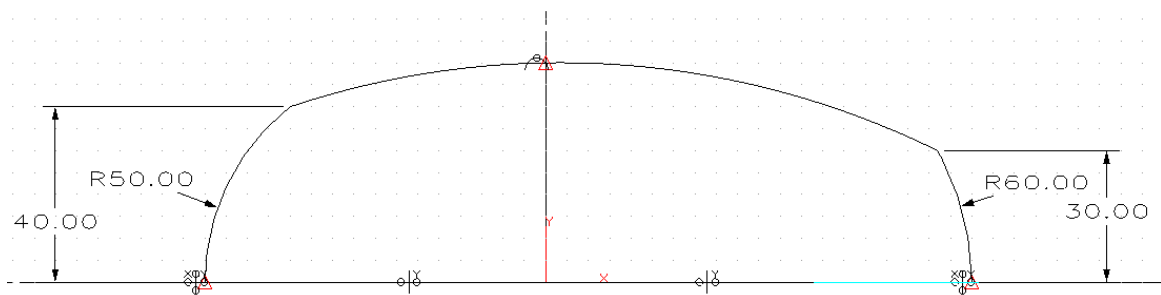
Press the **constraint solve button**, and the center points of **both arcs** will snap to the horizontal position aligned with the lower reference point, as shown below.

This would ensure that if the final solid were mirrored about the **XY-plane**, the resulting surfaces made up from the small arcs, would be perfectly tangent to each other.



Using Quick **dimensions** attach **two radii** dimensions and **two vertical** dimensions. This will fully constrain the sketch.

**Modify** the values as shown below. You may have to use “**Delay Solve**” until all the dimensions have been updated to the values below.



**Exit** the sketch.

**The sketch we have just created contains a deliberate problem**

Create **Variational sweep**.

Select the above command.

Pick the **newly created sketch** as the sketch profile.

Pick the **red straight line** as the **sweep path** (make sure the **pick filter** is set to **curve** not sketch).

Notice that the preview of the solid is completely corrupted.

**Do not** complete the command. Especially if you have shading switched on.

**Cancel the current command.**

What's gone wrong?

Let's check the sketch as it moves along the sweep path to see where the problem lies.

Pick the **move entities along curve** command . ( from **Move** command  on the **Shape** Tab)

Pick the **newly created sketch** as the entity to move. Press the middle mouse button to terminate selection.

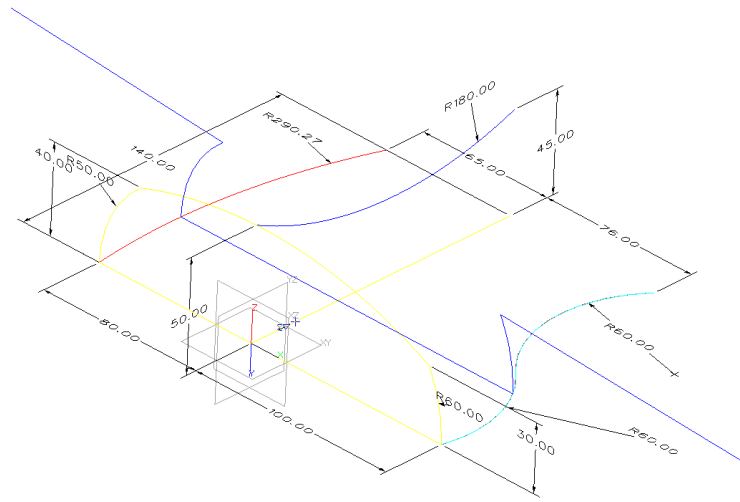
With the pick filter set to **curve**, pick the straight line.

Right click and pick **On Entity**.

When you move the mouse pointer the sketch moves along the **sweep path** continually updating as it moves.

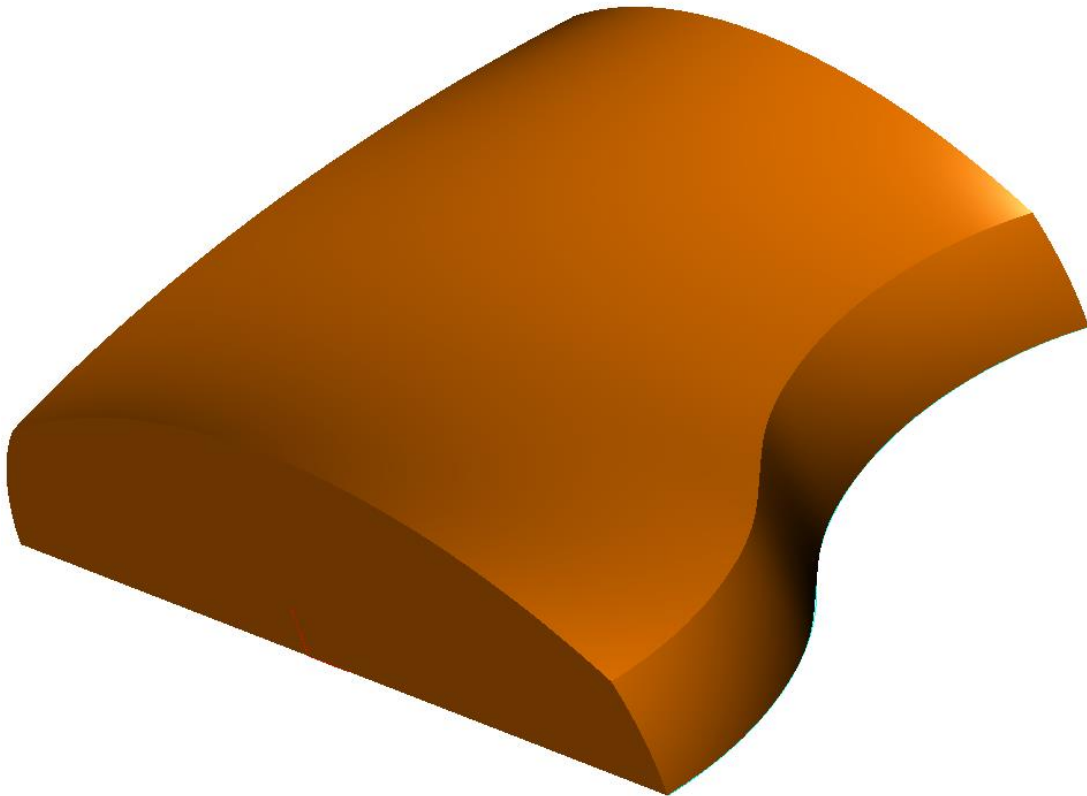
The large radius gets bigger and bigger until it at some point becomes a straight line. ZW3D will not create an arc of zero radii, so the arc flips, as shown below.

Escape out of the current command, and we will edit the sketch profile to suit.



Let's edit the sketch and change the **40** dimension to **30**, and the **30** dimension to **25**.

Repeat the **Variational Sweep** command and the part will be as shown below.

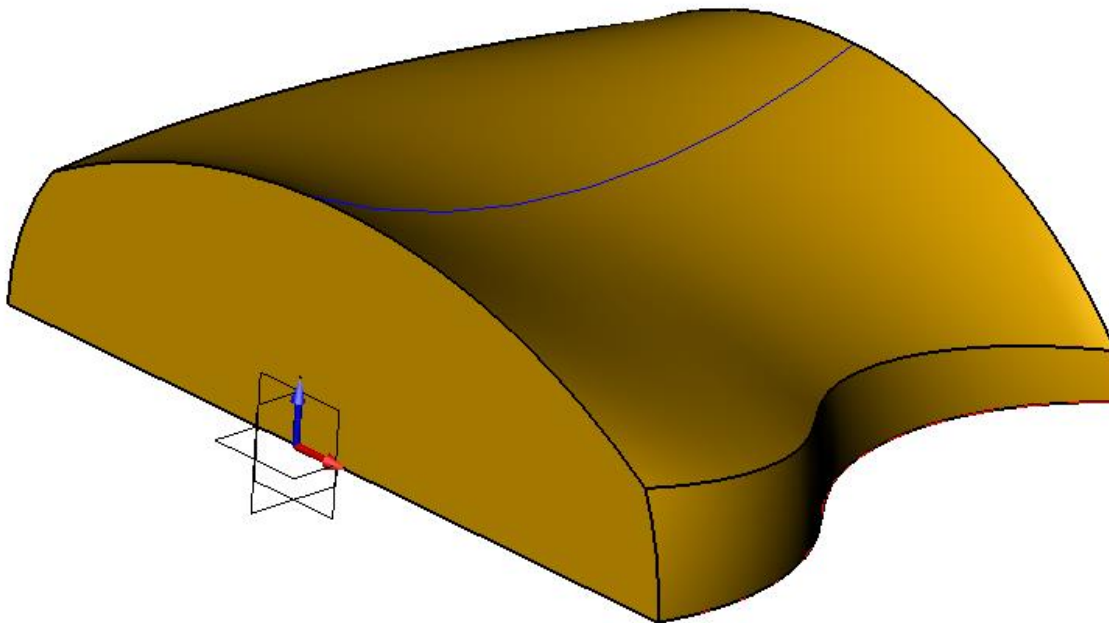


The above sweep was controlled by **Referencing geometry**.

The previous example used the variable **~VxPathParam**. We can use **both** together to control the sweep.

Let's say we want the **right hand 25mm** dimension to start at **25** and finish at **10mm**. With help from your instructor edit the sketch you created and create the **link a variable** to the **right hand 25mm** dimension.

Exit sketch, regen the part and the result will be as shown below.




## Variational sweep maintaining tangency with existing surfaces

We have seen how we can control variational sweeps using the **~VxPathParam** variable and also by using reference curves. A third option is also available, which keeps sketch entities parallel to surfaces as the sketch profile moves along the sweep path.

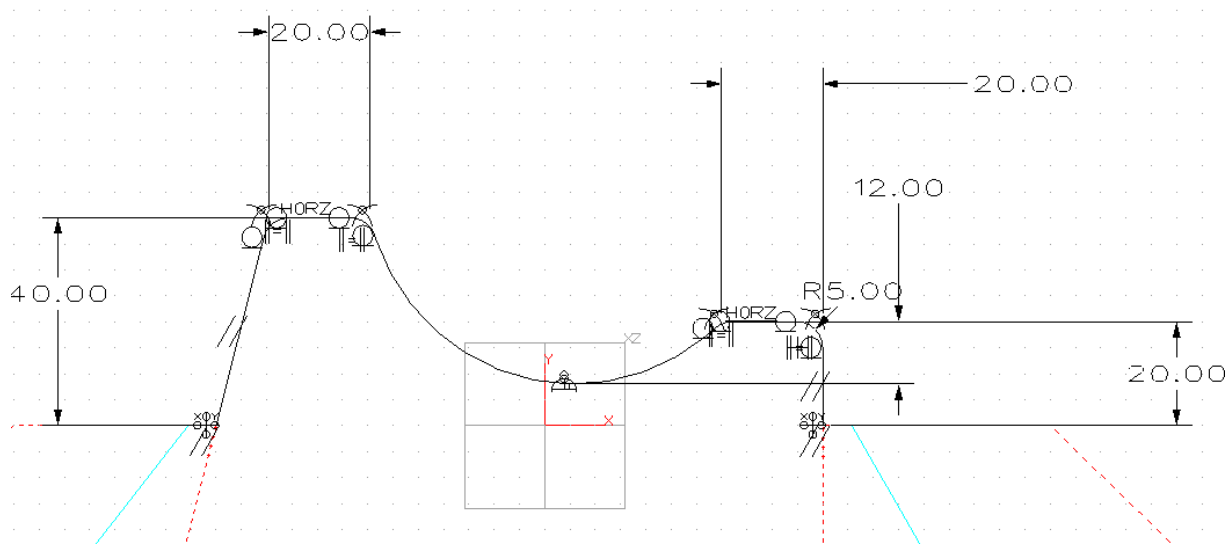
Clear the current session and edit the file “**Variational\_sweep3**”.

Edit the Sketch6 (double click on the history line) as shown below. Look carefully at both ends of the sketch, if you zoom in closely, you can see on both ends, small red dotted lines. These are reference lines that are tangent to the faces at both ends.

On the previous example you used the **Curve Intersection** command  (from the **Reference Geomertry** command) to create **reference points** that are used to control the sweep.

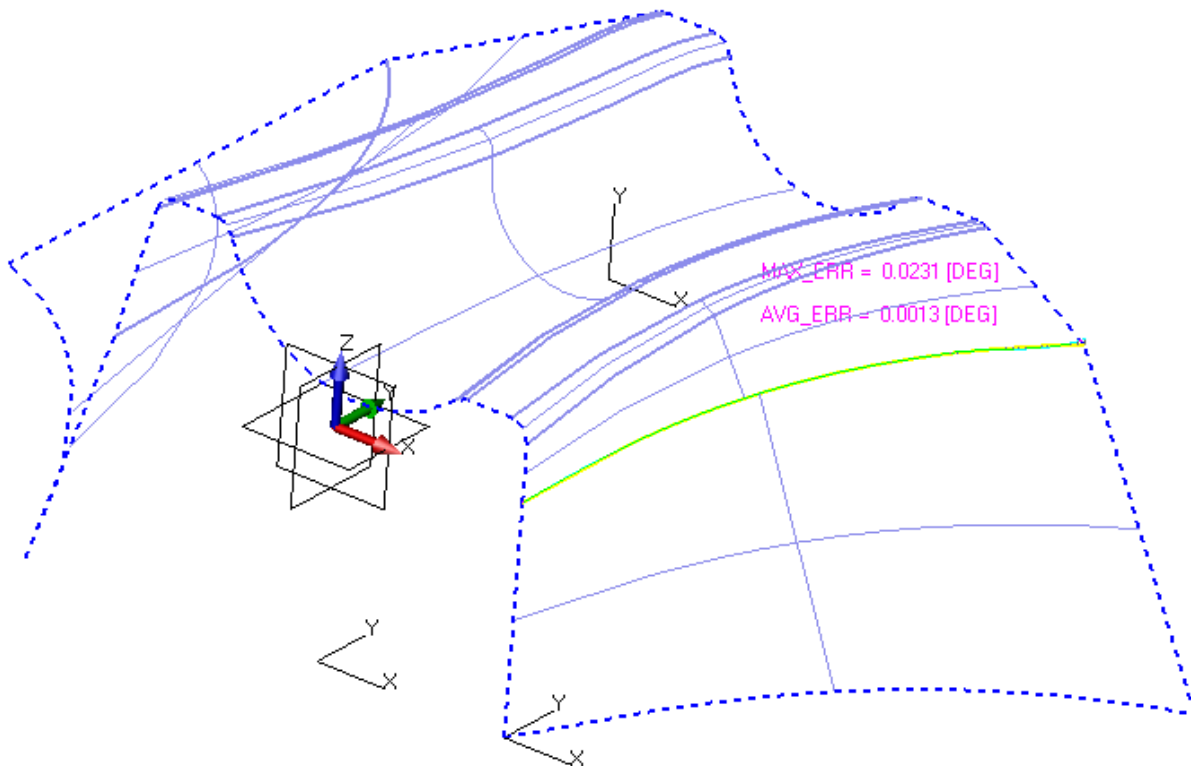
If when using the same command you picked the line option you will create the **tangent reference lines** described above.

We have created the sketch in this example for you however your instructor will explain how to create these tangent lines in more detail.



## Create Variational sweep face.

In this example we will create a swept face as shown below.



Notice from the **Inquire Tab** or pull down menus about “Display continuity error” .it is apparent that the two sides of the sketch stayed tangent to the referenced surfaces.

Let’s combine the two methods for defining a variational sweep.

Reference the variable [**~VxPathParam**] and Reference external geometry.

We need to write two equations similar to the ones we did in the first **Variational Sweep** example.

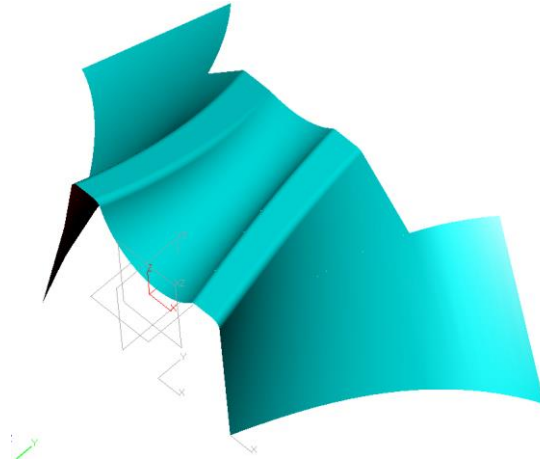
```
leftside = 40 {-20*["~VxPathParam"]+40} [0, 0]
rightside = 20 {20*["~VxPathParam"]+20} [0, 0]
```

**Right-click** on the **40** dimension > Link Variable. From the window, pick or type **leftside**.



**Right-click** on the **20** dimension > Link Variable. From the window, pick or type **rightside**.

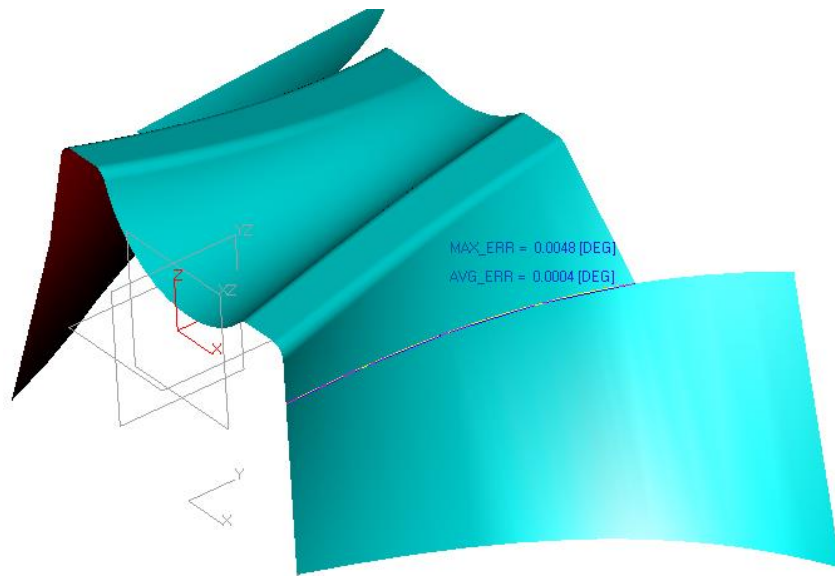
**Solve current sketch. Exit the sketch. Regen** the history.



We can now check the tangency between the sewn surfaces.

Go to **INQUIRE > Face > continuity error** and pick the edge between the old surface and the new sweep.

This confirms that the two surfaces did remain parallel throughout the entire length of the sweep.



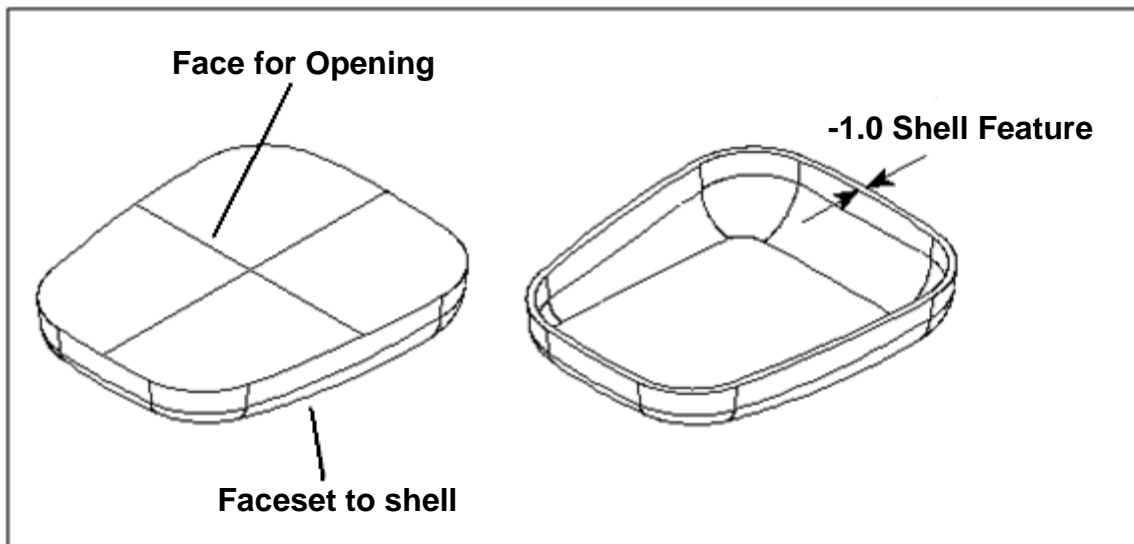
## **Chapter 4    Shell**

## Shell



(Shape Tab)

Use this command to create an offset shell feature from a solid. The offset will apply to all faces of the solid except those that have **Face offset attributes**. You can also specify which faces of the shell if any should remain open.



Shell Solid

Shells can also be applied to **open shape features**.

A negative value for **thickness** creates the wall thickness **inside** the original shape.  
(on the pink side)

### Optional inputs

#### Create side faces

Use this command option to control the creation of side faces during the **Offset Face** and **Shell** commands. When you select **side faces** the shape will be reconnected to form a closed solid (recommended).

#### Intersections

Use this command option to check for self-intersecting faces during the **Shell** commands. This option will also detect disappearing fillet faces, fillet corners, and chamfers. These checks require some time to complete depending on part complexity.

**Do not remove** – Ignore self intersecting faces.

**Remove fast** – Limited to removing self intersections on or near faces that become completely inverted, such as a fillet that is offset more than its radius. Anything beyond this is not handled.

**Remove all** – Remove self intersections and inverted regions (including inverted inner loop cases) as well as can be done. Faces that intersect other faces in the shape will be split.

Clear the current session and edit the part “**Shell\_1**”.

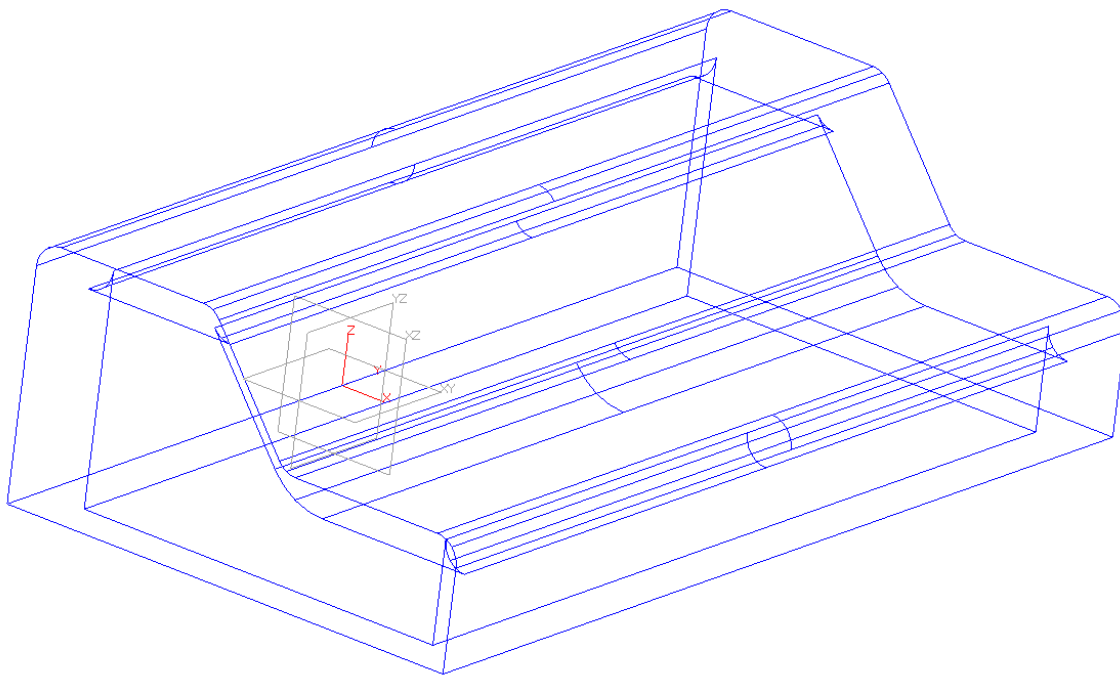
Using the **shell** command.

Shell the shape with a shell thickness of **-10mm**.

Select the **bottom** face as the face to be left open.

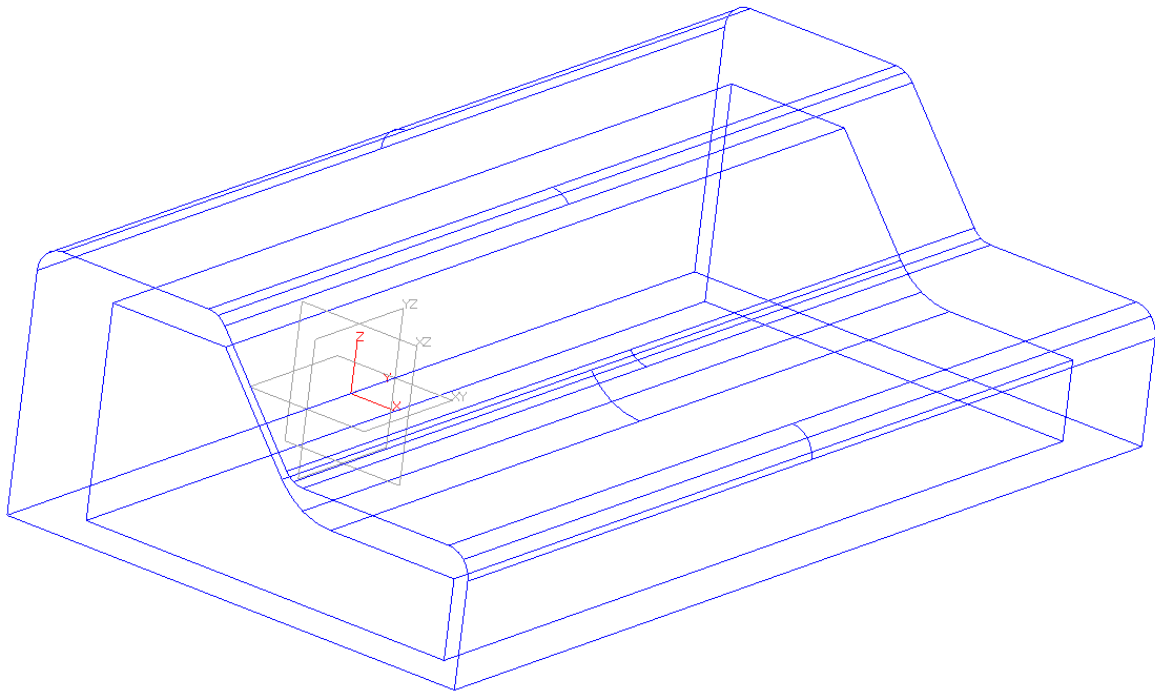
Set the **intersection mode** to, **do not remove**.

You can see that the shell is successful but has produced self-intersecting faces, and because we selected the **do not remove** option no warning was given.



**UNDO** the previous command.

Repeat the **shell** command with a shell thickness of **-10mm** and this time choose the intersection option **remove fast**. The intersecting faces are removed.



## Volume Offset



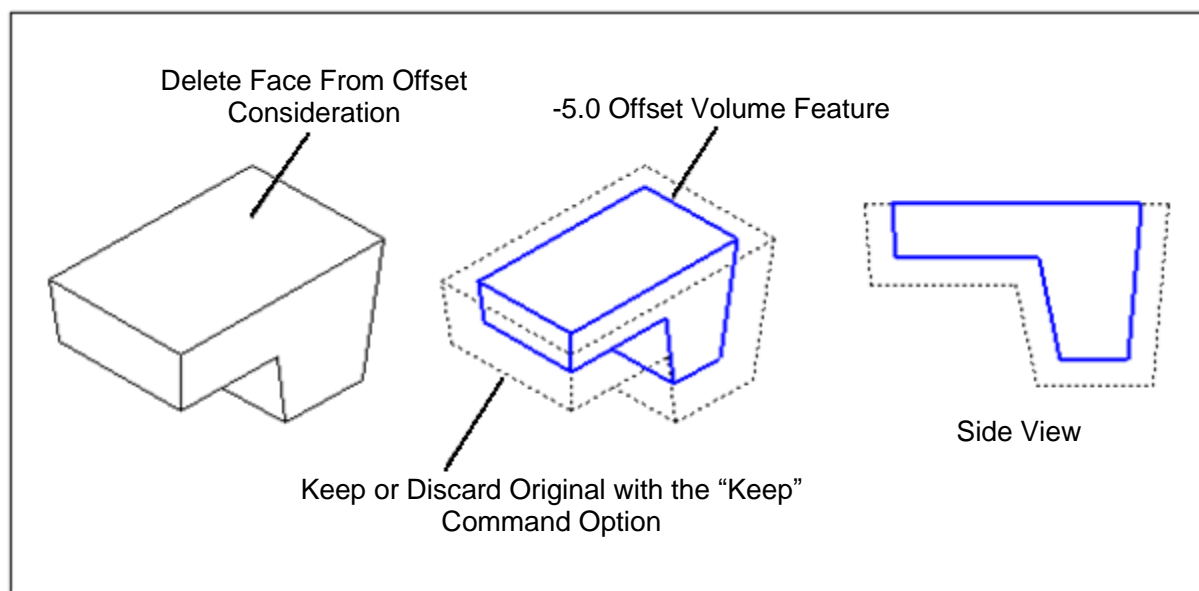
(Shape Tab)

Use this command to create a feature by offsetting the volume of one or more solids.

Select the solids to offset.

Specify the offset value. A negative value indicates an internal offset. A positive value indicates an external offset.

Select which faces if any should be deleted from offset consideration. These faces will reside in the new feature but will not be offset.



Offset Volume

### Keep original shape

Use this command option to keep the original solid(s) during the **volume offset** command. The original solid(s) will remain separate features.

When the **Keep** option is set to **No**, the new faces replace the original faces (which are deleted) and the **Name Labels** are transferred from the old to the new faces. If the **Keep** option is set to **Yes**, then new name labels are created for the new faces.

## Face Offset



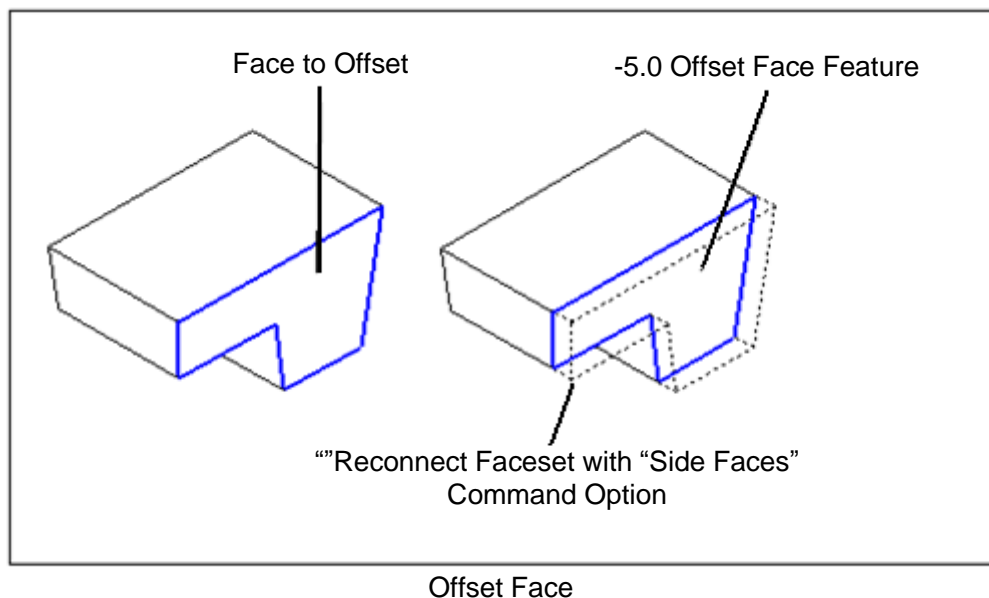
Use this command to offset one or more faces. The **Side faces** command option determines if the face remains attached or unattached.

Select the faces to offset.

Specify the offset value.

A negative value indicates an internal offset.

A positive value indicates an external offset.



### Create Side Faces

Use this option to determine how side faces (if any) are created. Side faces are used to reconnect the offset face to the source shape.

**Do not create** - Edges are exploded and the face is offset. The result will be a disconnected face or shape.

**Create** - All bounding edges are recomputed. Side faces are created where required.

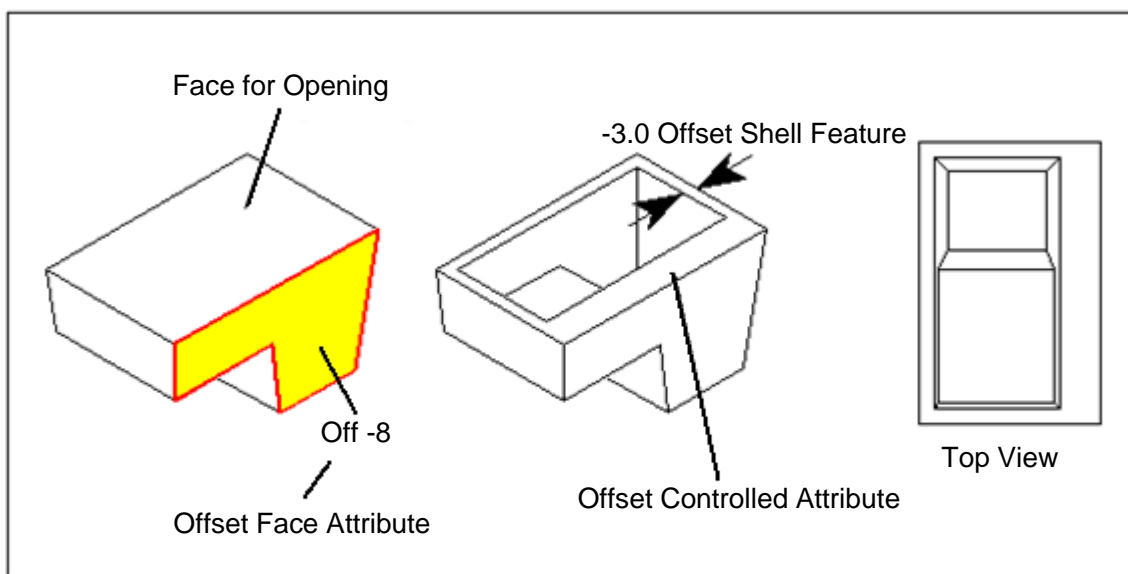
**Force** - Side faces are created in all cases. Bounding edges are not recomputed. This is similar to the **"Do not create"** option above except that the gaps are filled with new faces.

## Add Face Offset Attribute

It's used to attach an offset attribute to a face.

The attribute will govern that face despite any global offset values applied by the Shell, Volume Offset and Face Offset commands.

It allows you to control the wall thickness at selected faces independent of the part's typical wall thickness.



Adding a Face Offset Attribute

## Deleting Face Offset Attributes

To **delete** a face offset attribute after the shell or offset volume is created, select the offset feature during the rollback part history command.

When the history stops **before** the feature, use the command **Delete next operation**



to delete the face offset attribute.

Once deleted, pick **Replay all operations**



from the history replay menu to replay the remaining history without the attribute.



## ***Inlay (Shape Tab)***



The Inlay command is used to embed a set of closed loops of curves, as would be generated by an outline text font, into a face and then offset the new interior faces created by the desired distance, giving a raised or sunken text effect. Unlike the Emboss command, Inlay actually creates new faces, edges, vertices, etc., allowing for sharp transitions from the raised surfaces to the base surface. Also, the user can afterward select the new faces to assign them unique attributes, such as color. Finally, Inlay provides built-in draft and fillet options.

The curves list **MUST** lie on the surfaces of the faces selected. (The **Wrap Curve onto Face** command is useful for meeting this requirement before the Inlay command is invoked).

## ***Wrap Curve onto Face( Wireframe Tab)***



Use this command to "wrap" curves (in the form of a sketch) onto a part face. This command is perfect for placing logo on a part, maintaining its aspect ratio regardless of the contour of the part. This command is similar to placing a decal or label on the part. It also allows a functional curve that specifies rise and dwell in terms of an angle to be wrapped onto revolved surfaces such as cylinders. For example, a tread pattern could be wrapped onto a tire.

For revolved surfaces, the sketch can be interpreted as a height/angle curve. For all other surfaces, the XY coordinates of the sketch can be interpreted as UV coordinates on the face. The command supports different methods and options. See the menu above right.

Exit the current part and edit the part "Inlay\_Curve".

If we **Step through the history** of this part you will see that there are three important features.

Sketch1, Wrap1 and Inlay1.



**Edit the Sketch1 feature. If your name is Krystyn, leave it as is!!!**



Use a window pick or pick-all RMB option and erase all.

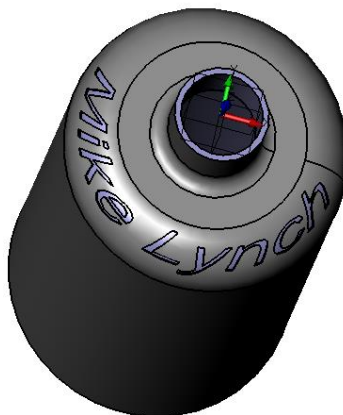
Open the **Sketch** Tab and use **Insert ReadySketch Text**. 

Type your name the way you would like to see it on the bottle.

Specify the location of (0,0).

Change the font if you like. All the fonts available for this command are True Type Fonts (TTF). They need to produce a closed boundary.

For now just look at the other options but don't change them. The curve option allows you to pick a curve and the text will follow or offset from that geometry. When was the last time you really looked at the tires on your car? Look at all the text wrapped and inlayed on your tire.



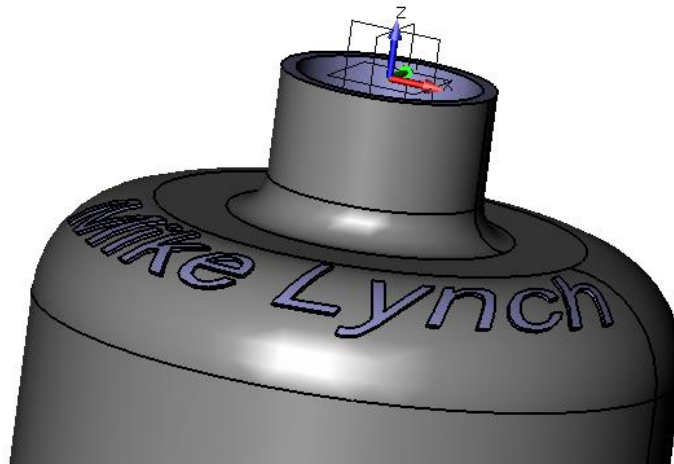
**Edit** the **Wrap1** command. Click on **origin** and center the text of the selected face. OK

**Edit** the **Inlay1** command.

We used the RMB option Pick All so no matter what the text says, all the curves will be included.

A negative offset will push into the surface and a positive offset will stand off the surface. OK.

**Replay** all operations.



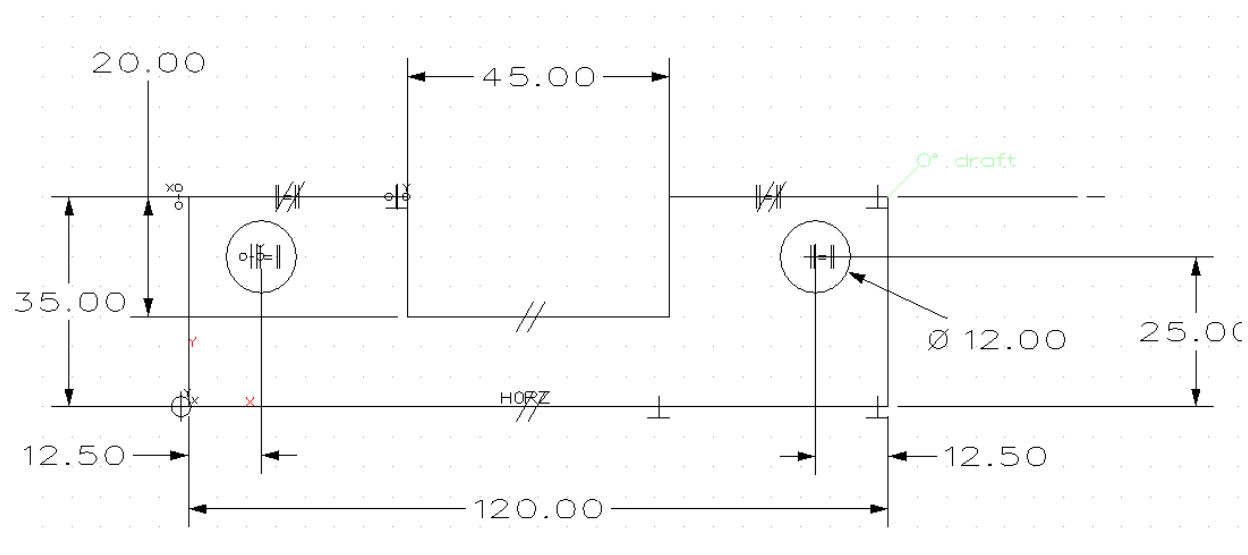
## **Chapter 5    Family of Parts**

## Part Attributes Form and creating families of parts.

Use this form to define attributes assigned to the active part. Many of the attributes can be accessed by the **vxBOMfmt** file and displayed during the **Show Assembly Parts List** command. Standard attributes, user-defined attributes, driving dimensions, and **PDM** search descriptions are supported. For further information on the **vxBOMfmt** file and how you can modify this file to change the format of **BOM tables**, please refer to the **HELP Manual**.

Clear the current session and edit the part "**Part\_family\_1**".

Double click on the part to show dimensions and sketches. (Pick filter on **ALL**).



Double click on the sketch to enter the sketcher as shown below.

When making families of parts, we can choose dimensions that can change on the new parts. Although we could use the default dimension names, it is perhaps more useful to give the dimensions meaningful names that would more easily identify the driving dimensions when they are assigned to the part attribute form.

For our example let's say we want to be able to change the following dimensions.

The **35mm** height, the **45mm** slot width, the **20mm** slot depth, the **120mm** part width and the **12mm** hole diameters.

To make these dimensions more obvious in the attribute form, **create** and **assign** variables to the dimensions as follows. **If** you are unsure how to do this, look ahead.

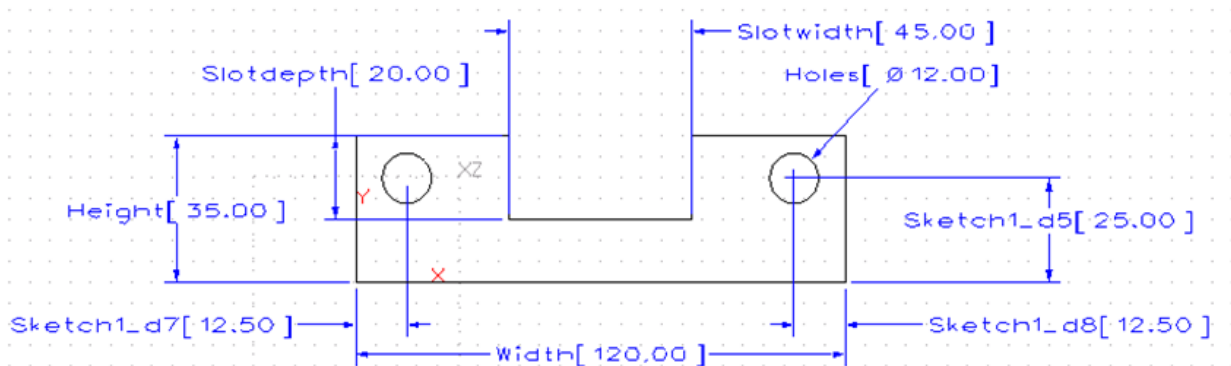
35mm dimension = Height  
 120mm dimension = Width  
 20mm dimension = Slotdepth

45mm dimension = Slotwidth  
12mm dimension = Holes

Answer: **EDIT** pull down

Edit > Edit equation then **Right Click** the **Dimension** and select '**Link to Variable**'.

The sketch with the linked variables is shown for clarity.



In the Equation form, the following **Equation** are set:

```
Height = 35 mm {35 mm} [0, 0]
Width = 120 mm {120 mm} [0, 0]
Slotdepth = 20 mm {20 mm} [0, 0]
Slotwidth = 45 mm {45 mm} [0, 0]
Holes = 12 mm {12 mm} [0, 0]
```

When you have created and assigned the variables to the correct dimensions, exit the sketcher, and then regen the part.

Next we will use the **part attribute form**(from **Attribute** pull down menus > **Part**) and attach the dimensions that will be used on our new parts, to this form.

But first we shall discuss the **part attribute form** in more detail.

We shall discuss some of the various fields in the form, and then create our own specific **part attributes** later.

For a more complete description of the **part attribute form** please refer to the **HELP Manual**.

## Standard Attributes

**Name** - The name of the active part. The current name is supplied. If you change the name in this field, the name of the part in the active file will change.

**Number** - Use this attribute to define a part number. Maximum character length = 48.

**Designer** - Use this attribute to define the part designer by name or initial. Maximum character length = 80.

**Manager** - Use this attribute to define the department manager by name or initial. Maximum character length = 80.

**Supplier** - Use this attribute to define the primary supplier. Maximum character length = 100.

**Cost** - Use this attribute to define cost information.

**Class** – This is the CAM Class attribute. See the other options. Maximum character length = 100.

**Material** - Use this attribute to define part material information. Maximum character length = 80.

**Description** - Use this attribute to define a part description. Maximum character length = 80.


**Keywords** - Use this attribute to enter keywords for PDM searches. Each keyword should be separated by a "," comma. Maximum character length = 128.

## Driving Dimensions

Key dimensions can be added to the list of driving dimensions. The part may be modified by changing the value displayed in the input field. Dimensions can be assigned to, modified from, or cleared from a driving dimension. Refer to the table below.

Using Part Attribute Driving Dimensions	
To assign a driving dimension:	Select one of the driving dimension buttons and then select a dimension. The dimension name will be displayed on the button and the dimension value will be added to the input field.
To modify a driving dimension s:	Edit the driving dimension value in the input field and pick <b>OK</b> or <b>Apply</b> to regenerate the part with the new value.
To clear a driving dimension:	Select the driving dimension button that you want to clear and then middle-click to clear it.

### Assign driving dimension from our part on the part attribute form

Toggle the display of dimensions, so that we can see all dimensions. 

Activate the **part attribute form**.

### The following is for information only.

Pick **OK** to accept the part attributes to close the form. The part will be regenerated if any **driving dimensions** were modified.

Pick **Apply** to apply the part attributes without closing the form. The part will be regenerated if any **driving dimensions** were modified. This is a quick way of modifying key driving dimensions and viewing the results on the fly, while you are still in the part.

Pick **Reset** to clear all attributes except the **Name** attribute.

### Continuing with our example.

Press the first **driving dimension** button on the **part attribute form**, and then pick the **35mm** dimension from our part.

The variable **Height** and the value **35** will be entered in the form.

Repeat for the remaining driving dimensions as shown above.

Notice the variable **Extrude1\_Base\_d3**. This is the system name assigned to the **length** of our extrusion.

When complete press the '**ok**' button to exit the form.

Now **exit** part.



## Use the existing part “Part\_family\_1” to make a new modified part

We should now be at the root object level, and the root object manager will be shown on the screen, similar to the form shown opposite.

Set the **Preview** option to **Attribute**. (Located on the top of the root object manager)

Left click on our part, **Part\_family\_1**.

The root object's part attributes will be displayed.

Note that the **Derive**, **Filter**, buttons are active.

The **Derive** option is used to create a family of parts based on key driving dimensions.

The **Filter** option is used to filter the **Root Objects List** based on standard, reference, or user defined attributes, or key driving dimensions.

## The Filter and Derive options

Use the **Derive new part** option to create a family of parts based on key driving dimensions.

For example, you may have one parent part called "**Block**." Whenever you need a new Block, you can preview the attributes of **Block**, modify the driving dimensions, and derive a new Block from it. The new part will contain all of the attributes previewed in the **Part Attributes Form**.

It will also contain all of the geometry of the part it was derived from as well as modifications to the driving dimensions. The new part will be created and added to the **Root Objects List**.

Use the **Filter** option to filter the **Root Objects List** based on the standard, reference, or user defined attributes, or driving dimensions previewed in the Part attribute form. Filtering by part attributes (e.g., by Designer) is practical for all parts while filtering by driving dimensions is practical for a family of parts.

For example, if you have created a large family of parts using the Derive option above, you can shorten the list to only those parts that contain a certain attribute or driving dimension.

Also, if a certain family of parts (e.g., cap head screws) is mixed in with other parts in a library, you can filter by a user defined attribute (e.g., Type : CHS) or a key driving dimension known to that family and list only the cap head screw family of parts.

All parts in the **Root Objects List** will be searched and only those parts that match the indicated attribute value or driving dimension value will be displayed.

You can again preview the attributes of one of the parts and filter the list even further if you wish.

Now **derive** a **new** part with these modifications.

You can change one or all values when deriving a new part.

Highlight **35** and change to **36**.

Pick the **Derive new part** option and type in the name **Part\_family\_2** and **OK**.

Pick **Part\_family\_1** again.

Highlight **35** and change to **37**.

Pick the **Derive new part** option and type in the name **Part\_family\_3** and **OK**.

These new parts are both derived from the original.

### ***Table-based family-of-parts***

You can also create a new part using a **simple table-based family-of-parts** interface. This is an alternate to the Part Attributes method of deriving new parts from an existing part. See Previewing Part Attributes and Derive a Family of Parts using the Part Attributes Form for more about deriving parts using the Part Attributes Form and the Root Object Browser.

Here is the basic procedure for creating table-driven parts:

First, set up the seed or parent part from which all parts in the family will be derived. Identify the dimensions or variables that will be used to drive variations of the parent part and add them to the **Driving Parameters Tab** of the parents Part Attributes Form.

Create a simple tab or comma-delimited file that defines a table of dimension values for variations of the parent part. This is an ASCII text file that can be exported by Excel or a similar spreadsheet editor. It has a format similar to the following:


	A	B	C	D	E
1	Eye Rod 1042256	Rod_Length	Hook_Radius	Blend_Radius	Rod_Radius
2	Eye Rod 1042256-1	200	15	35	3
3	Eye Rod 1042256-2	150	20	35	3
4	Eye Rod 1042256-3	200	40	40	5
5	Eye Rod 1042256-4	500	50	50	2

Spreadsheet of Table Data

```
Eye Rod 1042256,Rod_Length,Hook_Radius,Blend_Radius,Rod_Radius
Eye Rod 1042256-1,200,15,35,3
Eye Rod 1042256-2,150,20,35,3
Eye Rod 1042256-3,200,40,40,5
Eye Rod 1042256-4,500,50,50,2
```

Comma-delimited ASCII Text File of Table Data

The first line defines the name of the parent part (**Eye Rod 1042256**) and the names of the **driving parameters**. The subsequent lines define the names of the derived parts along with the desired values for their **driving dimensions**. Note that the commas could be replaced with tabs (i.e., tab-delimited).

Now return to the **Object Level** and pick the **Part Table** button  from the **Root Object Browser**. Select **Yes**, then the **File Browser** is displayed and you are prompted to select a part table file. This is the ASCII file that you create with the format shown above.

The part table data is then imported into the **Part Table** Form. Select the part variation to create (i.e., a row of the table) and pick OK. The first time the form is displayed, you are prompted to select a table file to load. Subsequently, the last file selected is loaded automatically. You can select a different file using the Open... button at the bottom of the form.

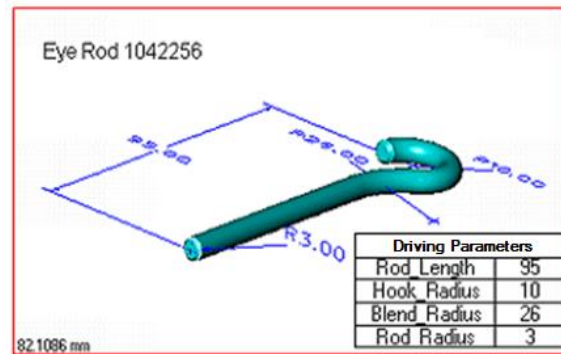
When you pick **OK**, ZW3D will perform the following tasks:

ZW3D will locate the parent part using the active file search paths.

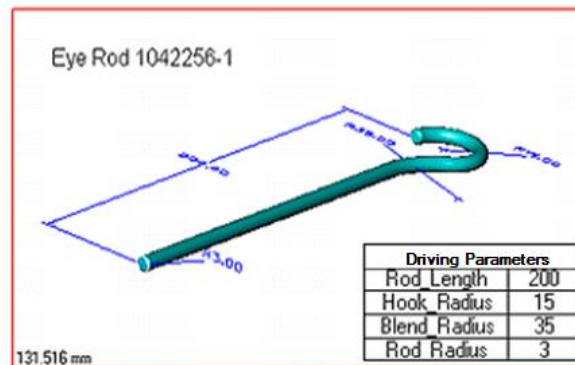
The part is copied to the active file and regenerated with its dimensions set to the values of the part selected from the table.

The part will be given the name defined as the first element of the row.

The part will then show up in the Root Object Browser list.



Parent Part for Table-driven Family of Parts



Derived Part Selected from Part Table

Here are some additional guidelines to consider:

The dimension values for a part variation must be within the envelope of valid values for the specified driving parameters (e.g., dimensions and variables). Otherwise, ZW3D may not be able to derive a part with the specified parameters.

ZW3D assumes that dimension/variable values stored in a table are given in database units (i.e., degrees and millimeters) unless otherwise specified by appending a unit abbreviation to the value, preceded by a period. Refer to the example part table file below.

```
Clamp1,Sketch1_d6,Sketch1_d1,Suppress
BiggerClamp,1.5.in,0.25.in,0
NoHoleClamp,1.0833.in,0.25.in,1
AltClamp,2.in,0.5.in,0
```

Comma-delimited Table Data with Units

## **Chapter 6    Morph**

The **Morph** command is found on the **Shape** Tab.

Use these commands to perform shape editing on and with faces.

These commands transform (morph) a shape by warping face geometry. Modifications are not limited to a single face, but work across edges maintaining the integrity of the solid shape.



Point



Curve



Offset

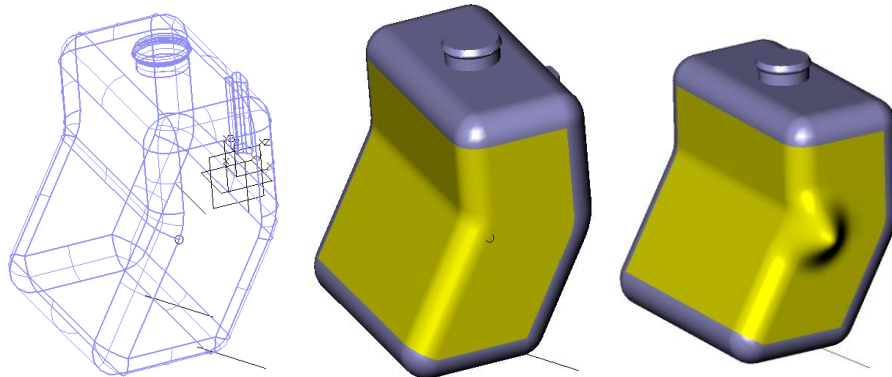


Curve to curve

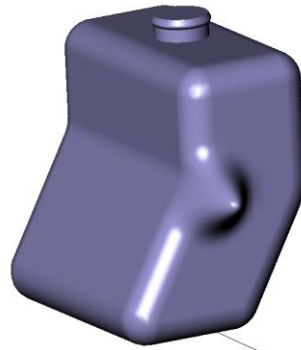
### Morph shape with Point

The transformation is performed by grabbing a point on a face and dragging it in different ways.

Move your cursor over these icons on the menu to see their descriptions.



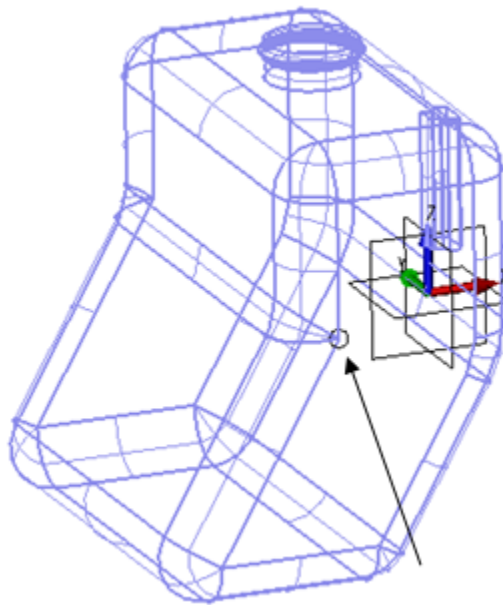
## Example 1 Morph Shape with Point



Final result

Open the file '**Parts.Z3**' found in your User\training folder and edit the part '**bottle**'.

Select the **Morph shape with point**  command.



Note : the critical point on the edge of the model as indicated above is the point we are going to morph.

Open the **filter** and the selection of the geometry Morph shape with Point and Curve allow to transform will be displayed.

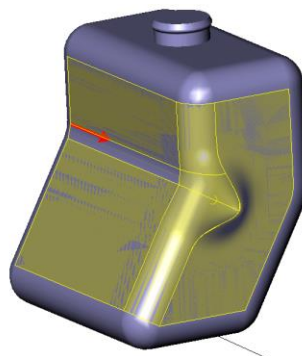
We could pick the entire shape to morph but you may not recognize the bottle when you are done. See the geometry options above right.



Pick the **Face** filter and select the seven faces shown.

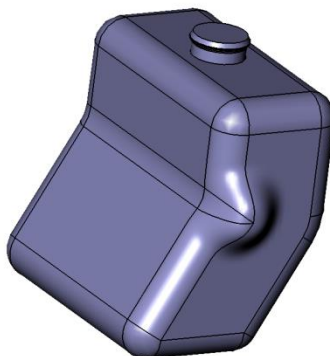
Pick the critical edge we have identified above left.

Pick the -Y direction by either picking on the geometry or right-click.



Select a distance of about 20.

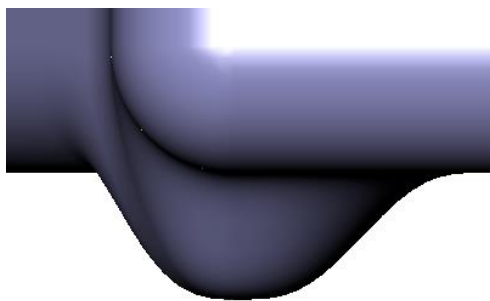
Pick **OK**.



You have now locally altered the topology of the bottle.



Try another example.



Redefine last.

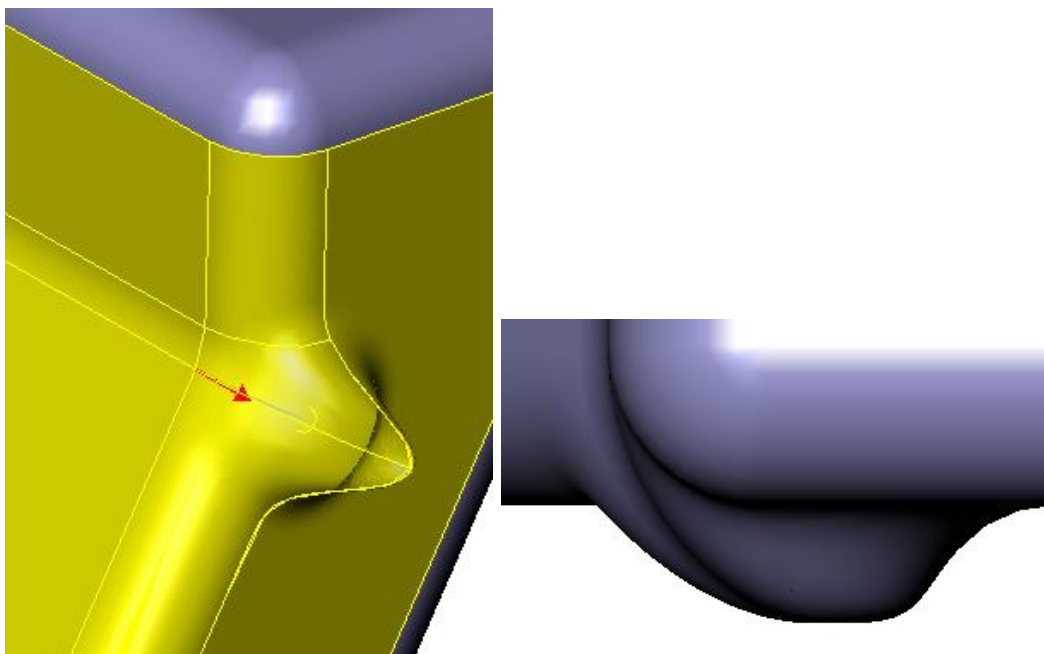
Pick the **translate along path** option. 

Use the same 7 faces.

Use the same point.

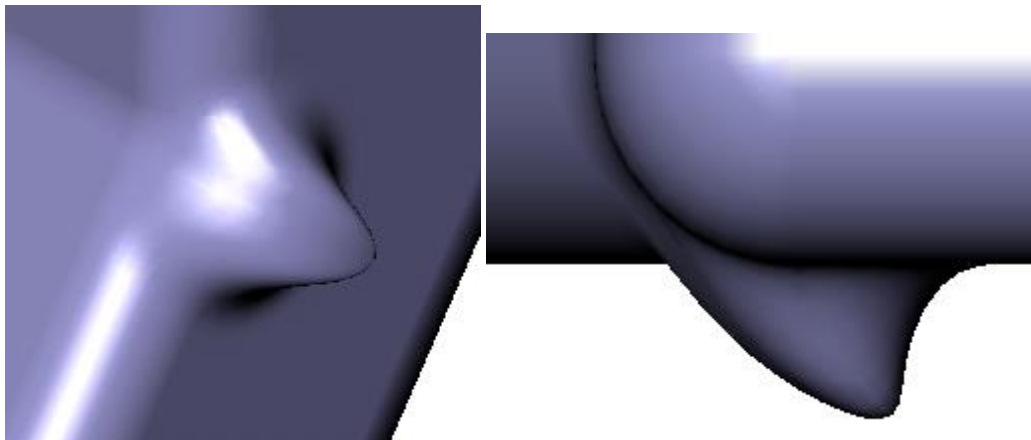
For the **path**, pick the fillet edge shown left.

Use the default **influence** (radius of influence) option of **20** and view the results below.



Redefine last.

Use an **influence** of 10.



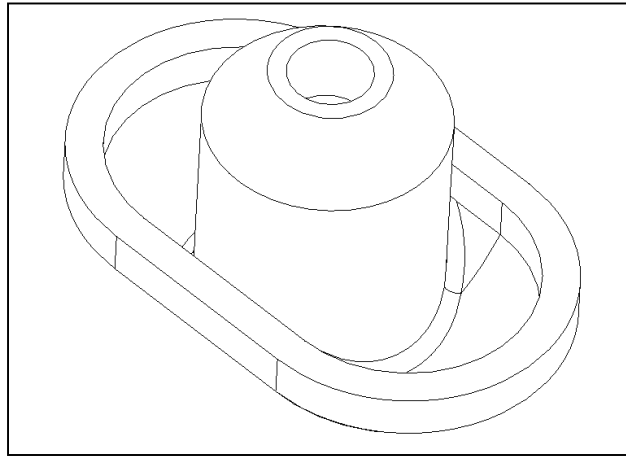
Notice that the smaller the radius of influence the more localized the change. A large radius affects more area of the model but the geometry is softer.

Now try changing the other options.

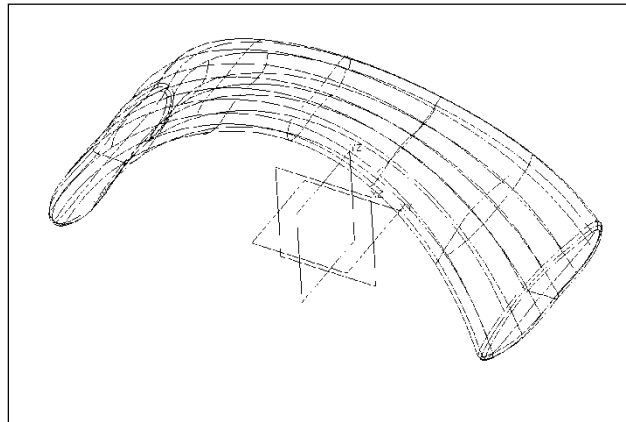
See the HELP Manual for a full description of the options.

## **Chapter 7    Additional Exercises**

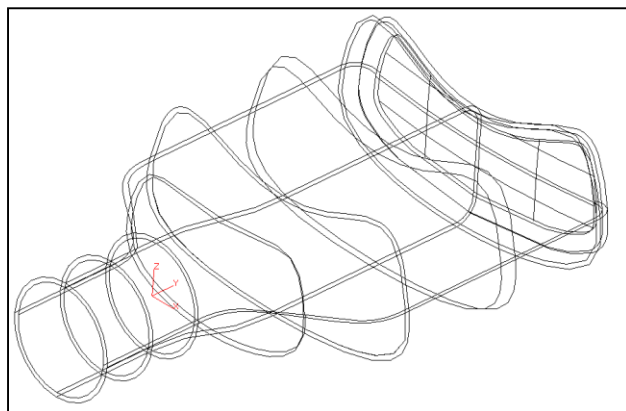
**Button**



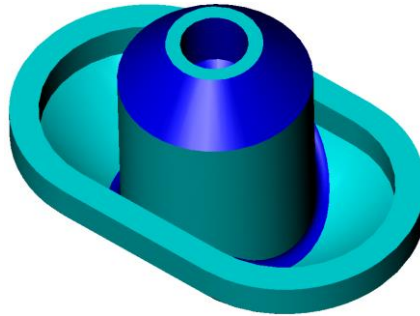
**Clip**



**Bottle**



## Designing a Button





Open a new Multi-Object file called: **Additional\_Parts.Z3**.

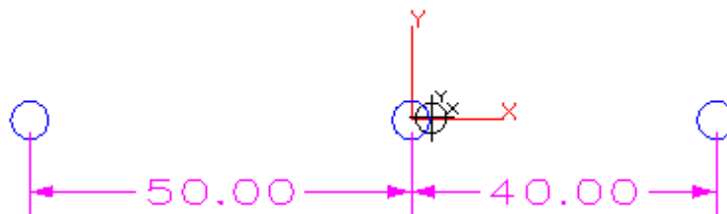
Edit new part called **Button**.

From the **Shape** Tab.

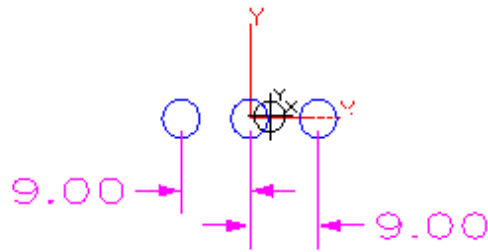
Select **Extrude** and create a new sketch on the XY Plane.



Create 2 points  as shown on either side on (0,0). Remember all new sketched now have a point and both a horizontal and vertical constraint at the origin.

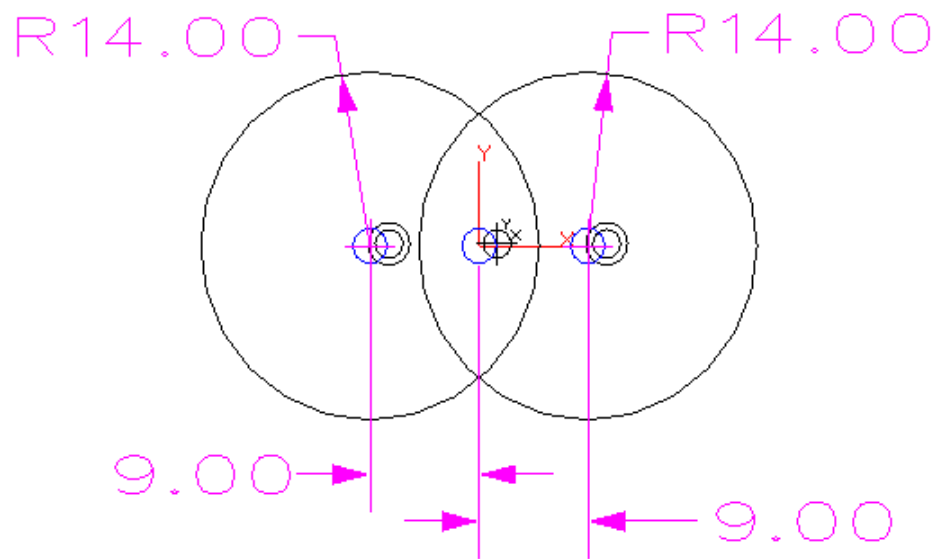
**Auto constrain**  all free geometry, picking the base point at zero.



Double click on both dimensions and change the value to 9. Optionally, you can Right-click on the dimension and pick **Modify Value**.



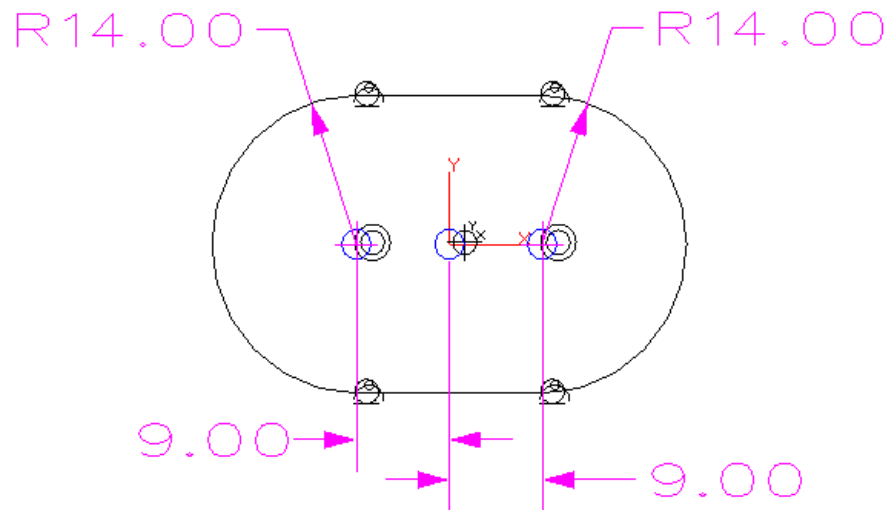
Create **circles** with a Radius=14.( Use Radius method  under Circle command )



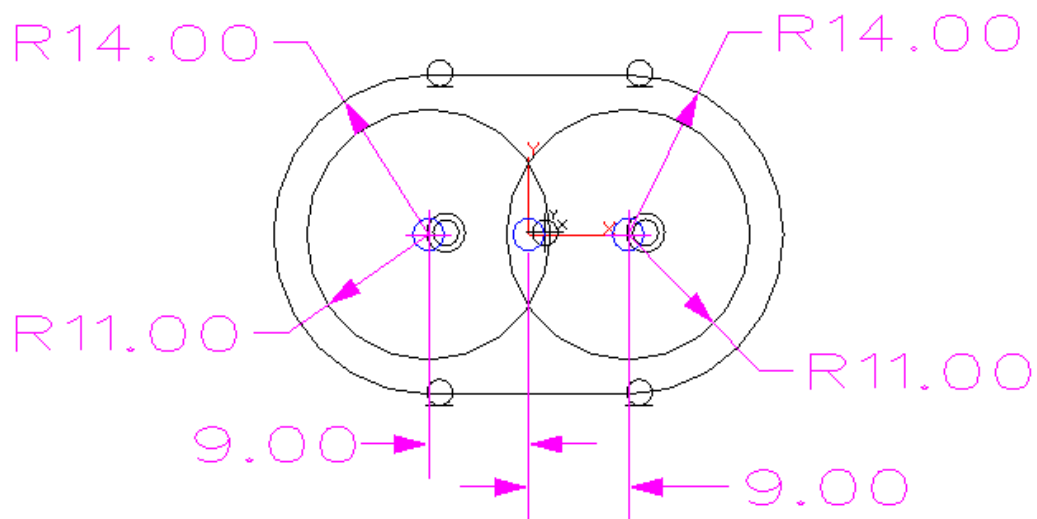
Use the **line** command  under the **Sketch** Tab to create two tangential **lines**.

With the right mouse button select the **entity** option.

Use one touch trim , or use trim/break curves  to trim as below.



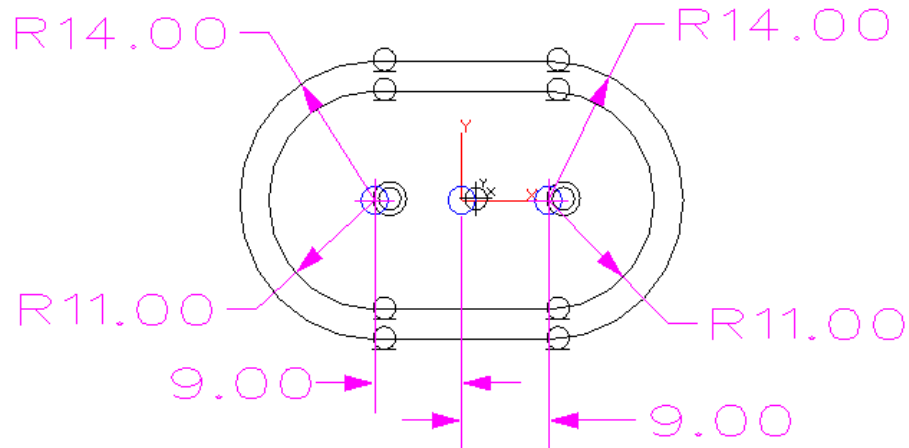
Create two circles radius 11. ( Use **Radius** method under **Circle** command)



Use the **line** command again to create two more tangential lines.


And **trim** as before.

Your sketch should look similar if you reposition the dimensions.



**Note:**

Depending on your **Design Intent** you could also create the inside profile with the

**Create Offset Curve** command  and Fully Constrain with Auto constrain and dimension command.

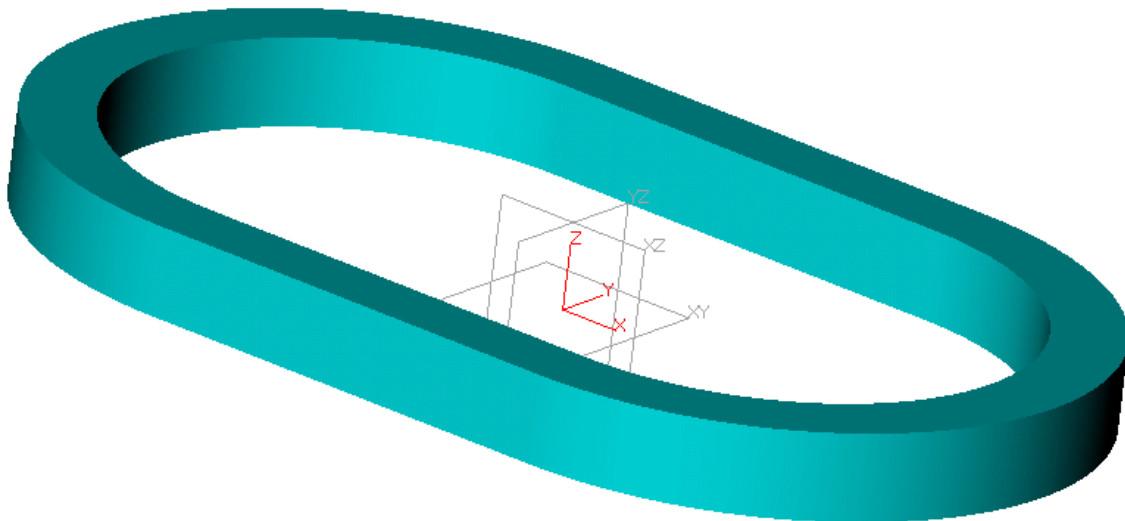
**“Exit sketch”** . (which returns to the **extrude** command.)

+Z Direction.


Start = 0.

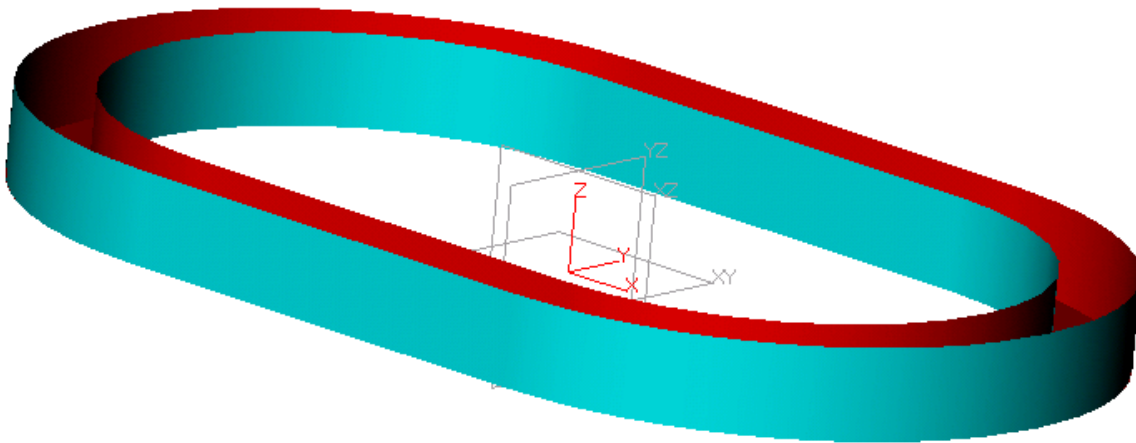
End = 3.5.

Draft Angle = 0.





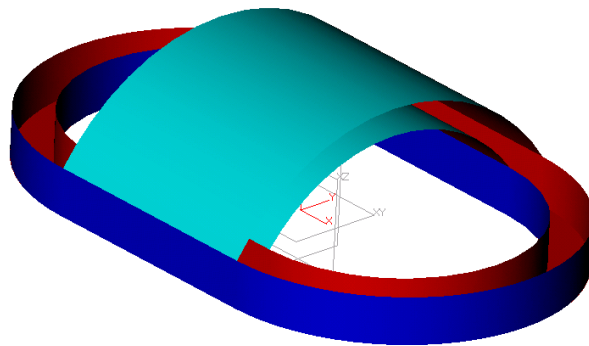
Erase the top face  .



**NOTE:** There is another Technique we could have used.

Extrude this shape and leave off the top face. Do you know what it is?

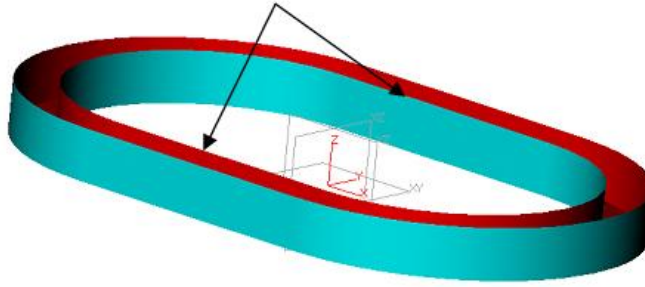
**Create the two faces shown at right**



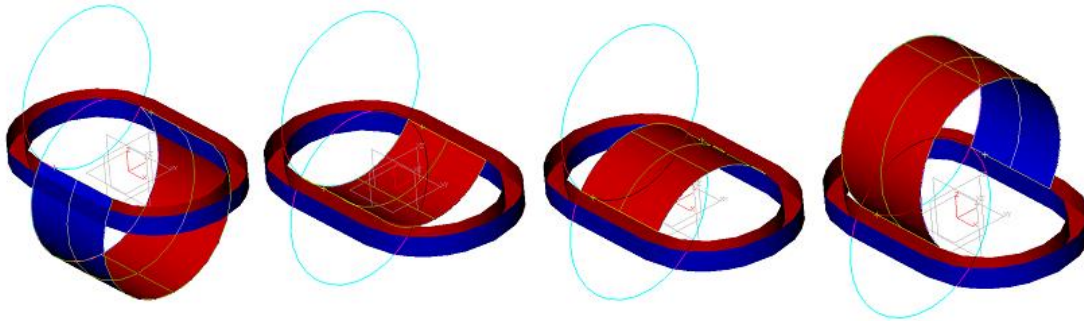
Use the **Circular bi rail** command  on the Free Form Tab.

Start by picking the two parallel edges in the inside of the shape near the left side for the paths.

Specify a radius of 14.



Solutions 1-4 are shown below. Cycle through the different solutions until you are comfortable with the command. Solution 3 is the result we are looking for.



Use the method above to create the outside face with a radius of 17.

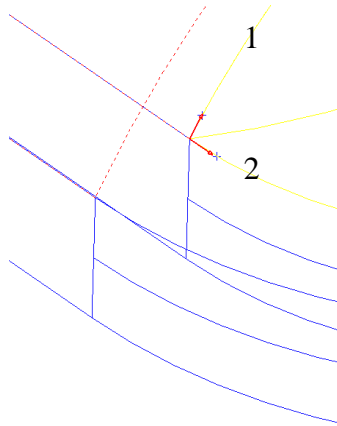
### Create four lofted faces

Create a **Loft** < **Add** .

Pick the two edges as indicated to the right.

Pick the edge of the surface we just created first.

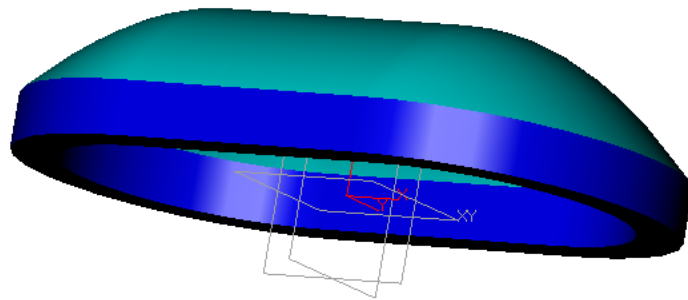
See below.




We need tangency on this first edge but not on the second.

In options tab, set the **Start** to be **Tangent**.

In options tab, set the **End** to be **None**.



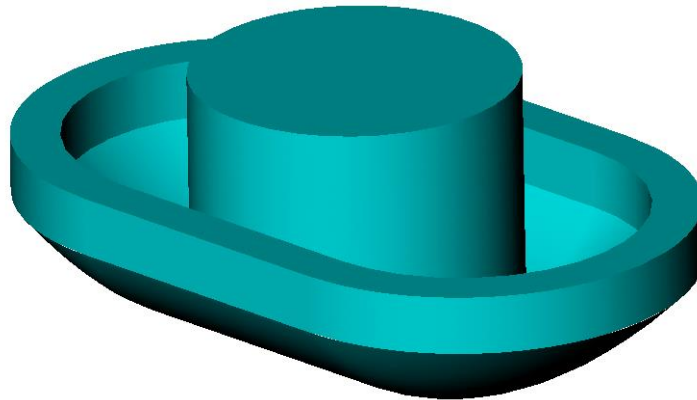
Now do the other three pairs or just two, then mirror; it's up to you.

**Note:** Check for **open edges** .

The Sew command should fail because the new faces should sew on insertion.

See your **Preferences**.

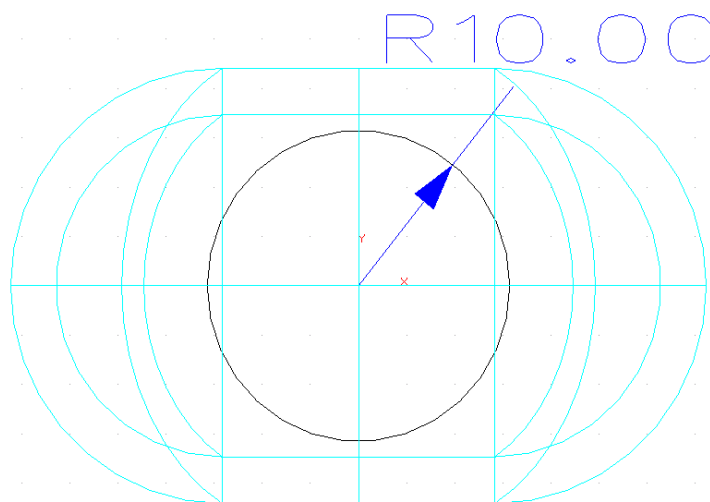
Create a feature underneath the button



Create an Extrude.

Create a sketch on the default XY plane.

Create a Circle **r=10** or **Diameter 20....your choice.** ( use **Radius** method)



“Exit sketch” and continue the extrude.

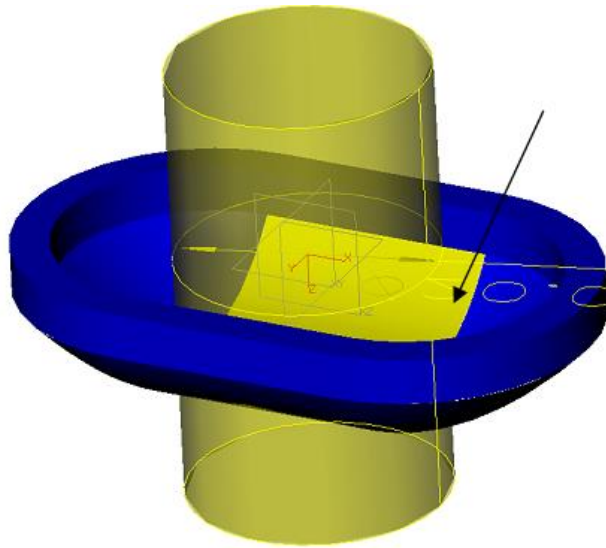
Pick **Add** method.

Start = **-15mm.**

End = **15mm.** (this distance must go entirely past the boundary face, 9 is the minimum)  
Define the **Boundary** face as the first circular bi rail face.

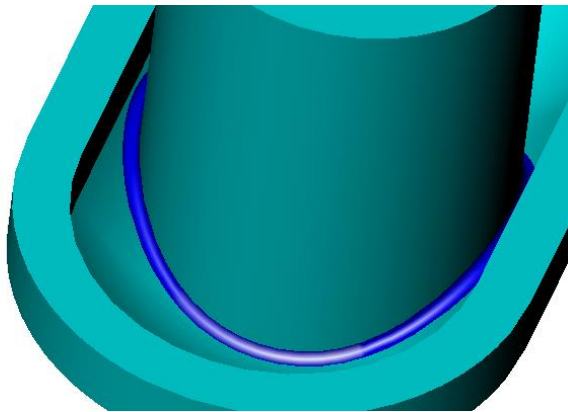
Make sure that both **end caps** are created if you were experimenting earlier.

Pick **OK.**



At the intersection between the button and the cylinder **create a Radius 0.5**.  
(See image at the right).

**Fillet with .5 radius.**



Shade the part and rotate the view similar to this.

**Right-Click** and select **Redefine Last**.

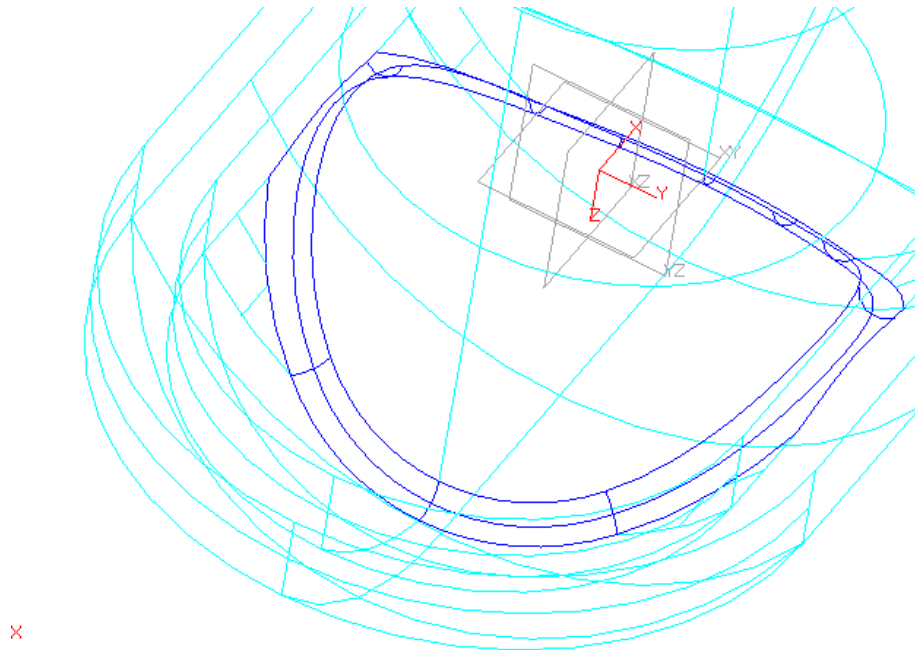
**Change the radius to 1** and select **OK**.

Now switch to a wire frame view.


Notice on the right and left side of the fillet there is not enough room for a full fillet to be generated so the fillet rolls up onto the outer face.

Check the box to **Hold fillet to edge**.

Notice the different results. Now the edge of the adjoining face is held intact.



### Create a threaded hole, in the center of the cylinder

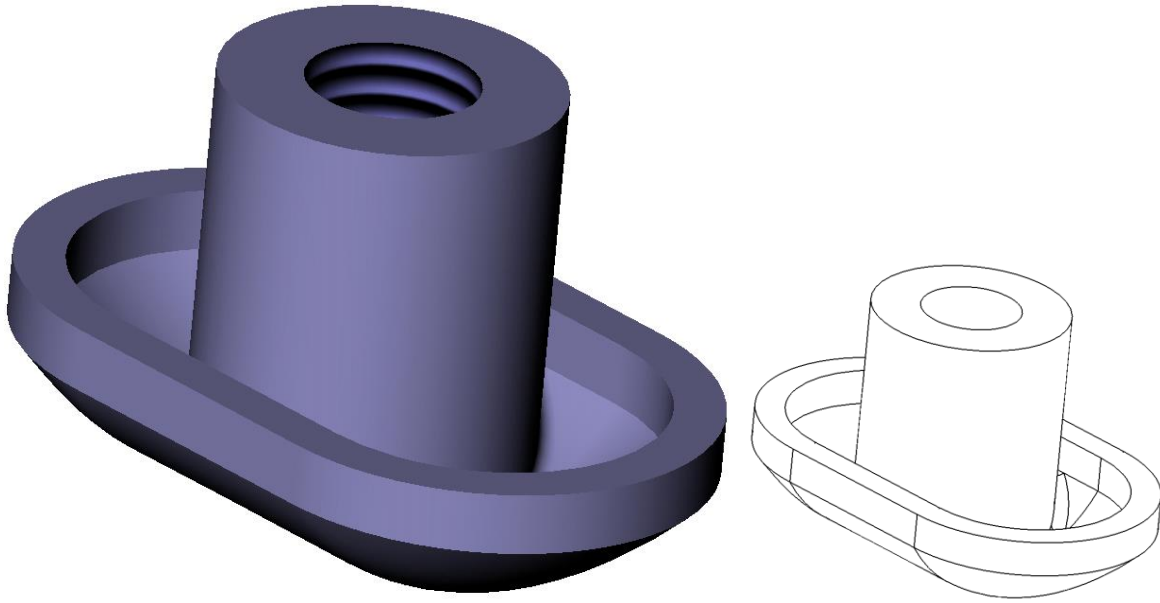
Pick the Hole command .

Start by picking the thread **M12X1.75-4H** from the field under Bundles.Z3.

Pick the top face and locate the hole at the critical center point.

Specify a hole depth of 20.

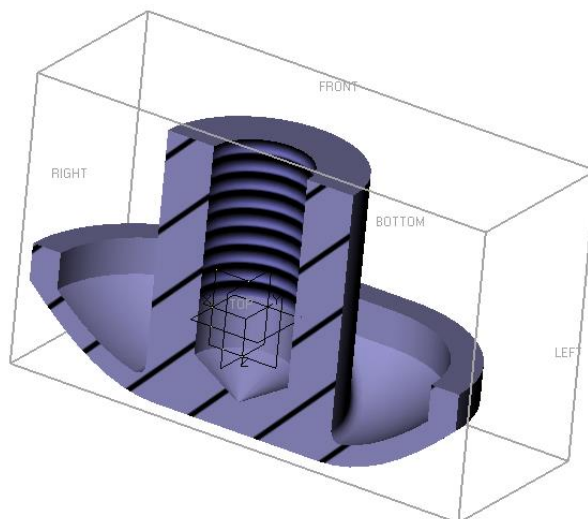
Make the thread depth 15.



To see inside the part go to the **section view** icon on the **Inquire Tab**.

Click on the front face of the block and slide your cursor back.

This view is turned on/off from the Utility Tab.

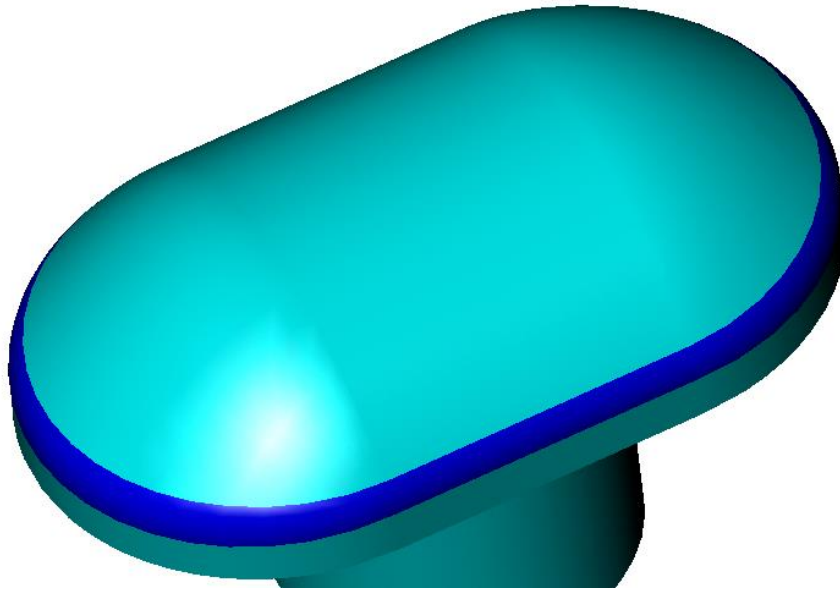


Create a constant radius 3mm as show below

**Fillet.**

**Note:** When picking the edges to fillet, pick one edge while holding down the **Shift** key. This will pick all tangent edges in one hit.

Radius 3 .



Create a Chamfer 5mm on the outside edge of the cylinder.

**Chamfer.**

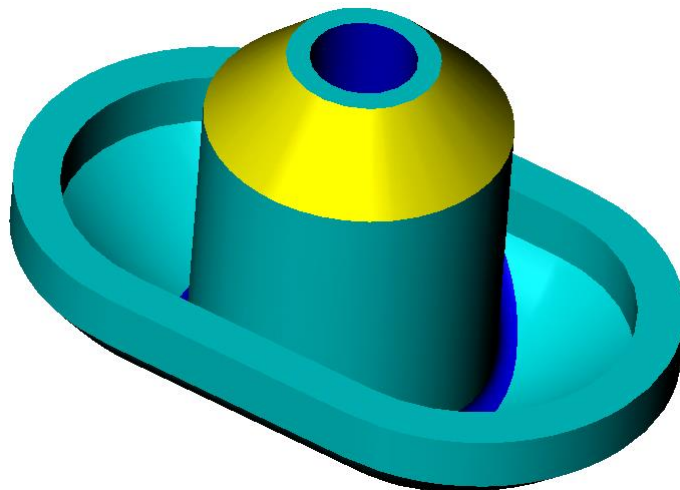
Setback = 4.

Save your file .

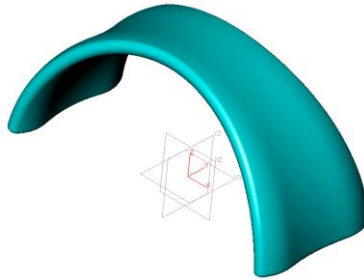
“Exit part” .

End of training exercise, “Button”.







## Designing a Clip



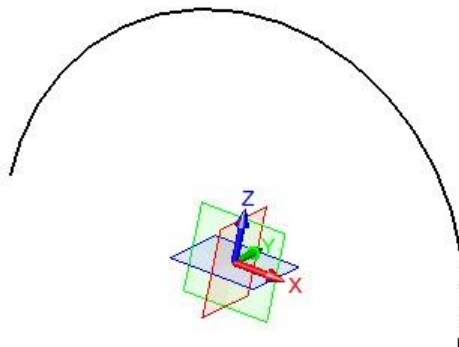
Edit new part called “**Clip**”.

Pick the WireFrame Tab.

Create an Arc  $R = 22$  mm.

Use **Radius**  method within the **Arc**  command under **Wireframe** Tab to create Arc (Start, End, Radius):

As follows:



Select a point on plane of new Arc:

Arcs are co-planer. Arcs that are created in the sketcher lie on the sketch plane.

However if you are creating 3D wire-frame Arcs, you must specify which Plane the Arc lies on, in our example the Arc lies on the XZ-Plane. It is defined through three points, the start and end points and a third point.

Right click and pick On Entity , Pick the XZ plane.

Select start point: -22, 0, -3

Select end point: 22, 0, -3

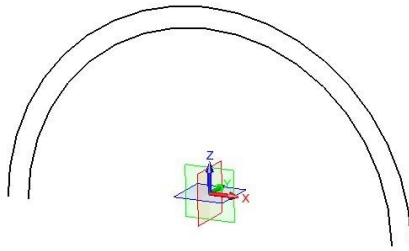
Radius: 22

Select the appropriate arc (positive Z side).

Create a second arc R = 19.8 mm.

Use the **Radius** command within the **Arc** command again to create Arc (Start, End, Radius):

As follows:



Right click and pick On Entity , Pick the XZ plane.

Start point: -19.8, 0, -3

End point: 19.8, 0, -3

Radius: 19.8

Select the appropriate arc (positive Z side).

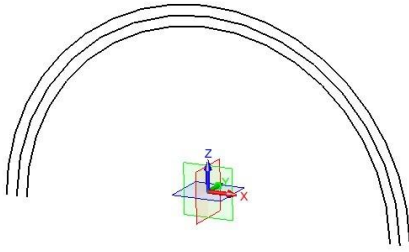
**Create a medial curve between both arcs.**

Create 3D medial curve: 

As follows:

Select first curve: Arc with R=22 mm

Select second curve: Arc with R=19.8 mm



## Create a new sketch containing two circles, on the XY Plane

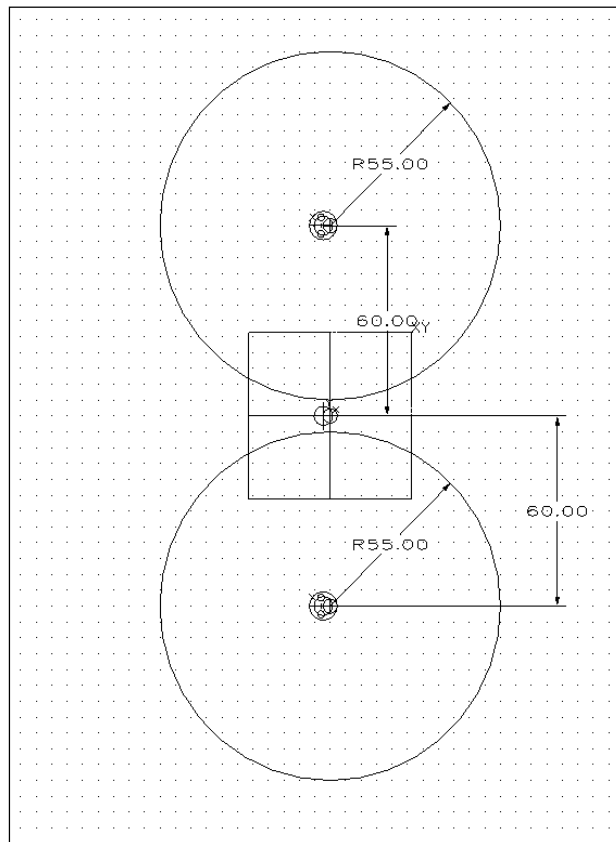
Create Circle (Radius).

Radius = 55;

Center: 0, 60

Center: 0,-60

Add the two vertical dimensions to fully constrain the sketch.

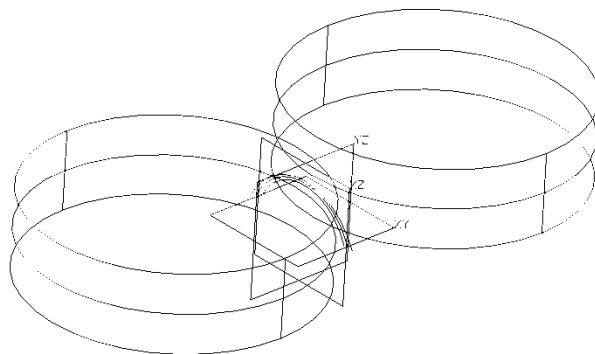


Exit Sketch.

**Extrude the two circles from the Sketch in the Z-Direction**

**Extrude.**

As follows:



Select the sketch containing the two Circles.

Start: -3

End: 30

**Extrude a face from the Medial curve in the Y-Direction.**

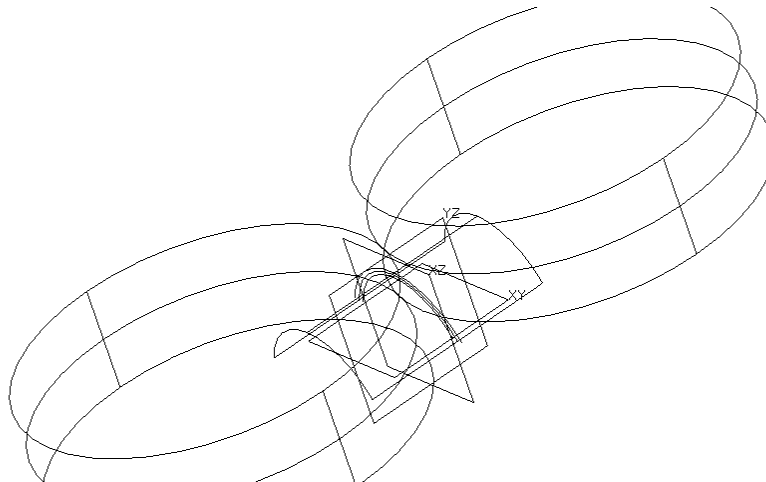
**Extrude > Base.**

As follows:

Select the Medial curve (between both arcs).

Start: -40

End: 40



### Create intersection curves between the faces.

In order to clearly see the new wireframe geom. We will change the Line color attributes

**Attributes pull down menus > Line > Color**, pick white.

OK.

All new lines/curves will now be created with a White color.

### InterX Curves (Wireframe Tab):

As follows:

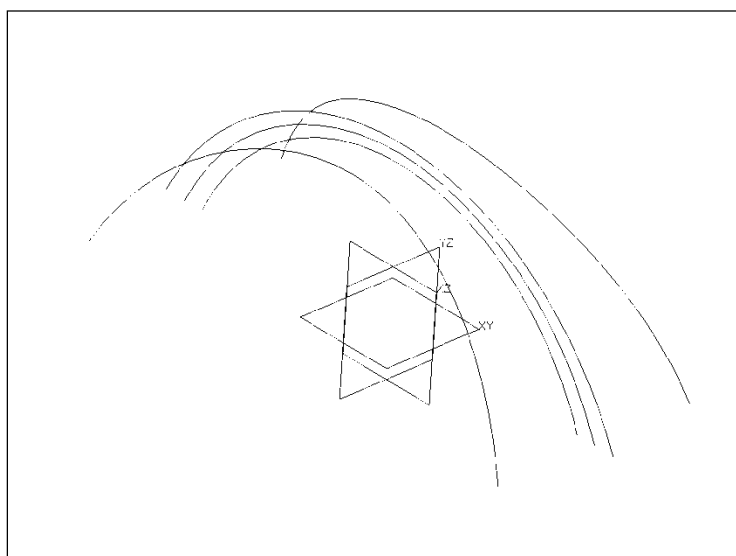
1. Shpe: Face created from the Medial curve.
2. Shpe: Filter on Face - Both cylindrical faces created from the sketch.



**In constructing the rest of the Clip we will only need the curves**

**Erase** all Faces from the Part.

Note: Set the **Filter** to **Shape**.



## Create a Swept face from a curve

As follows:

Create a Sketch in the XY Plane.

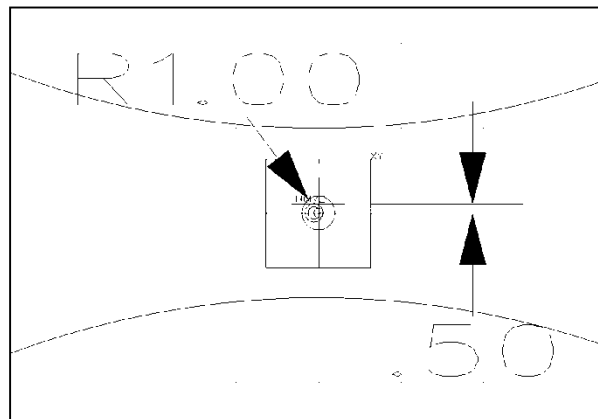
Right mouse button > **Insert Sketch**.

Show Target: 

**Blanks** out everything except the current sketch geom.

Create Circle.

Radius: 1  
Center: 0, 0



Turn the dimensions off.

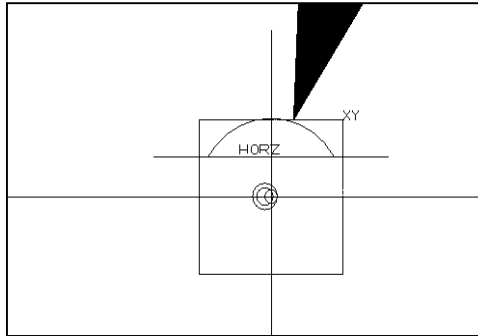
Use the **line** command under the **Sketch** Tab to create Line.

Create a line through two points, at 2, 0.5 and -2, 0.5.

One Touch Trim.

**One touch trim** away the lower portion of the circle.





**Erase** the line.

Exit the sketch.

**Create a swept face from the newly created arc**

Create Swept Faces .

As Follows:

Pick the small 1mm radius arc.

For the sweep path pick the right hand (as in sketch below) intersection curve.

For the sweep frame method, select Natural.....under the Orientation tab.



**Natural** - The default reference frame of the profile or entity is used to control the sweep along the curve. The sweep begins at the start of the curve path.

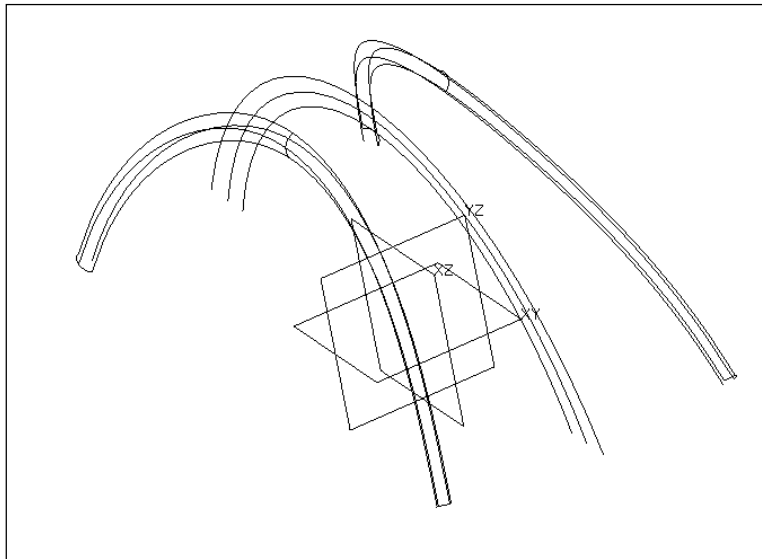
### Mirror the Swept Face in the XZ Plane

**EDIT** pull down menus > **Mirror**. Or pick the mirror command icon from the Tool Tab




As follows:

The entity to mirror is the swept face.  
The Mirror plane is the XZ-Plane.



Create four curves through the end points of the four 1mm face edges and the end points of the middle curves. Ensure that the curves are created with start and end tangency with respect to the 1mm radius edges of the two faces.

Through Point Curve (Wireframe Tab) .

Specify points, Right Mouse > **Critical**.

Pick the 3 points shown at the right starting with the right end point of the select edge.

Click on the **Degree** option and choose Degree 2.

Click on the **Start tangent** option.

Right mouse > **Curve Tangent**.

Pick the first face edge (Right hand 1mm radius edge as seen in auxiliary view below).

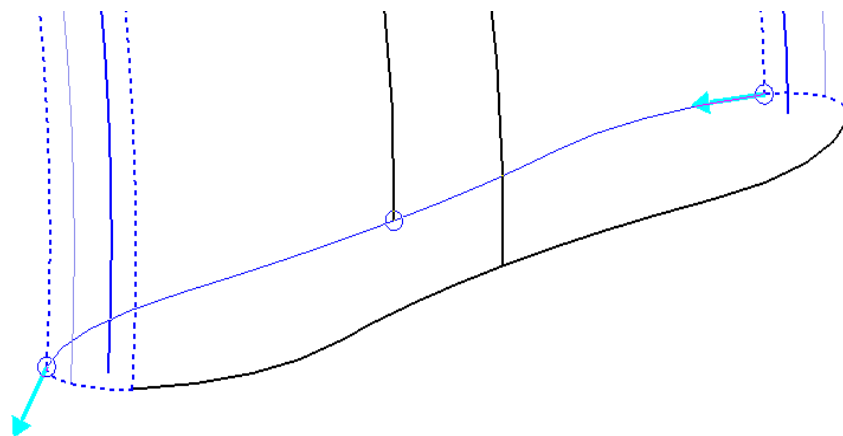
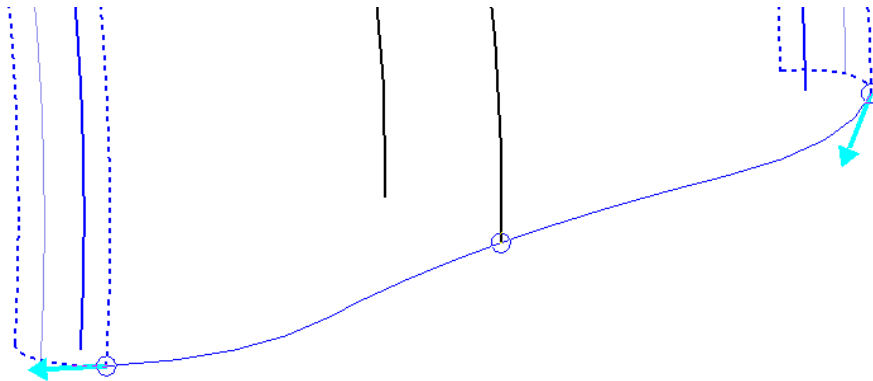
Pick the start point of the curve.

Pick the direction for the tangent as shown above (right). **Notice, the tangent arrow is in the same direction you are drawing the curve.**

Click on the **End tangent** option.

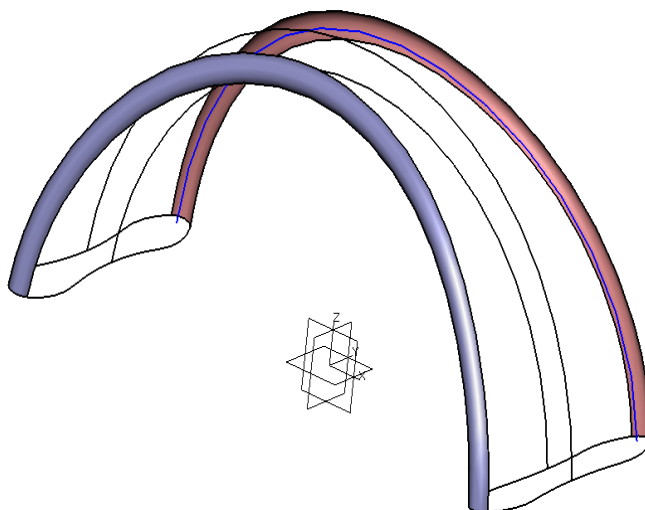
Pick the second face edge.

Pick the direction for the tangent **in the same direction you are drawing the curve.**



Repeat the same method to create the three remaining curves.

See the result below.

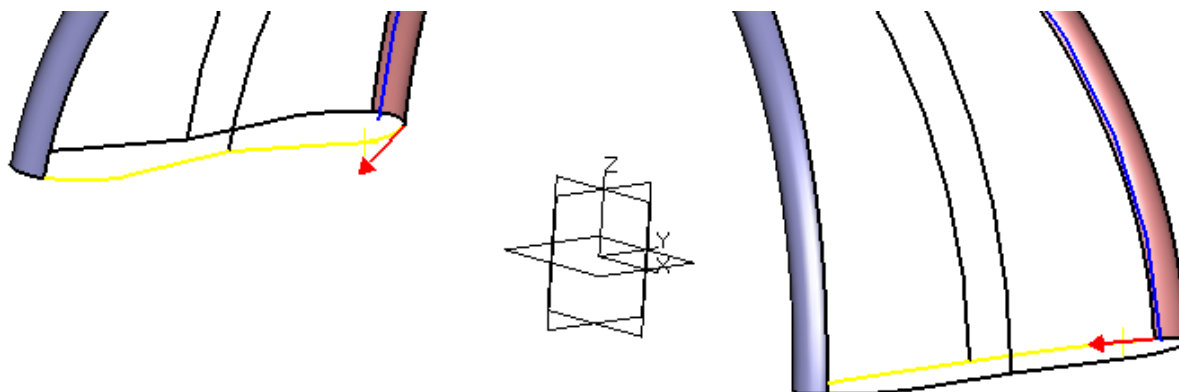


### Create two Mesh Faces from the individual curves and edges

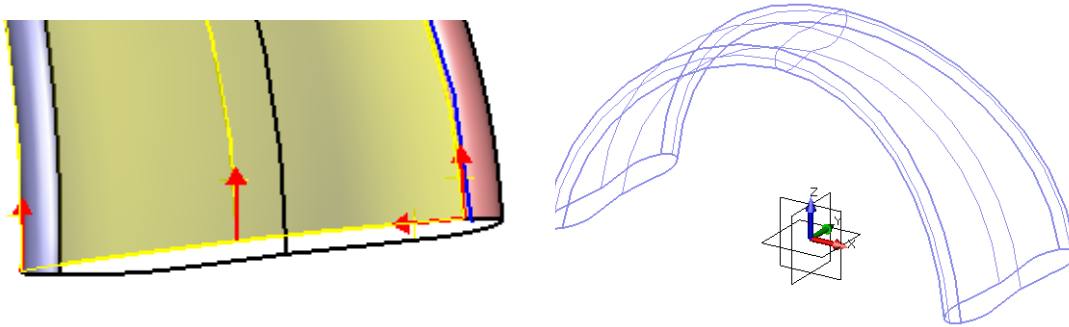
From the Free Form Tab.

Create faces from **Curve Mesh**.

Select list of curves (2) in the first (U) direction.



Select list of curves (3) in the second (V) direction.




Repeat this procedure for the outside surface.

**Erase** the curves used for construction.

The bottom faces are open and they extend below the XY plane.

Our design intent calls for a solid model that sits on the XY Datum so we will need to TRIM the model with the datum then round the edges.

Use **Trim ( Shape Tab)** picking your model then the **XY** plane .

**“Cap trimmed region”** should already be checked. This will add planar faces to both ends.

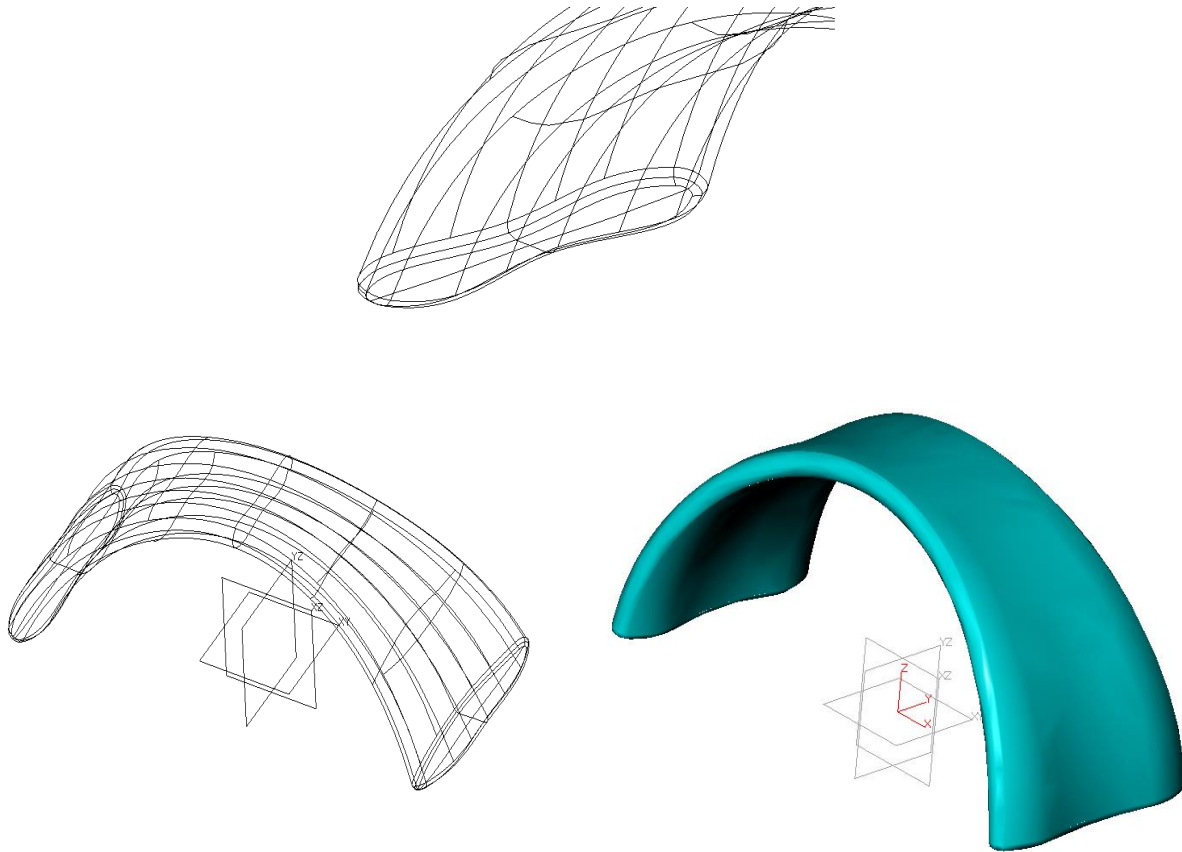
Make sure the direction arrow is pointing **UP**; this is the side to **keep**.

Create 0.5mm **fillets** on the bottom faces as shown.

Create **Loop Fillet**  method within the **Fillet** command.

Radius **0.5**.

This command only needs a face as input; Pick the newly created face at both ends of the Clip.

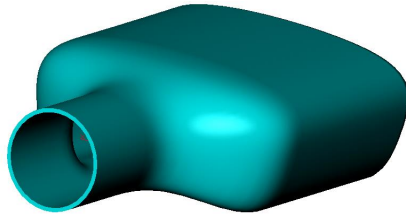


**Save** your file.

Exit part.

End of training exercise “Clip”.

## Designing a Bottle



Create a new part called **Bottle**.

Create an Extruded Face.

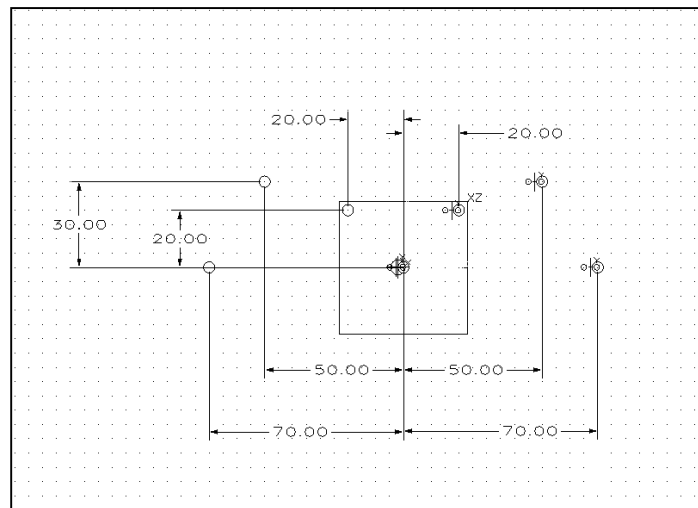
Select **Extrude** and create a new sketch on the XZ Plane.

Create Points (Sketch Tab).

Create six points as shown below, but first, set the point attributes through **Attributes > Point**. Change any or all of the attributes.

**Auto constrain** all free geometry. Pick the zero point as the base point.

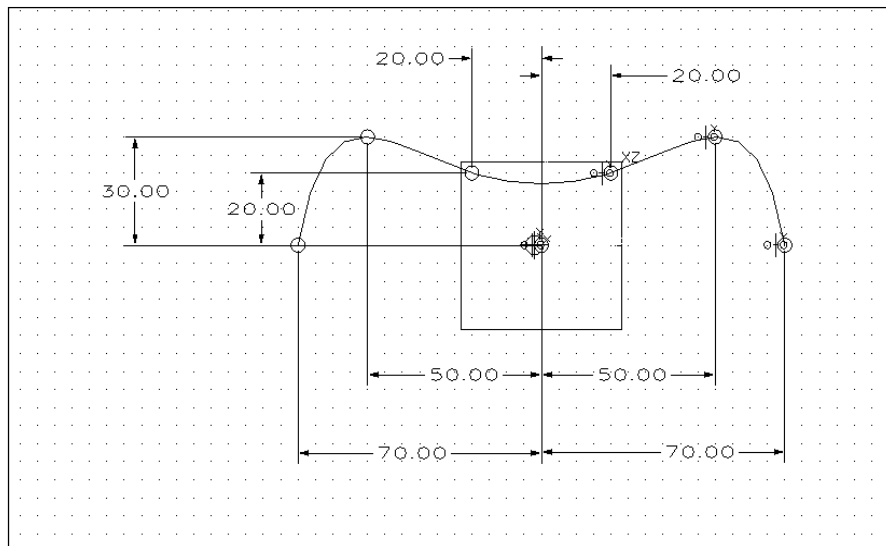
Change the values and text position of the points below.



## Create a Curve through points

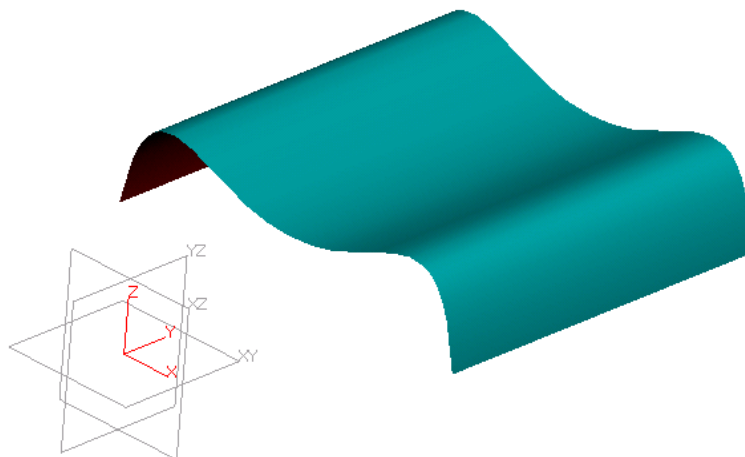
**Create curve through points.** Note: The Curve control points could have been inputted as the curve was created (not separate points) and still dimensioned as below.

This may have been set to degree 3 from the last exercise.



Exit Sketch.

Start: 70  
End: 200  
Direction: Right Mouse / +Y Direction





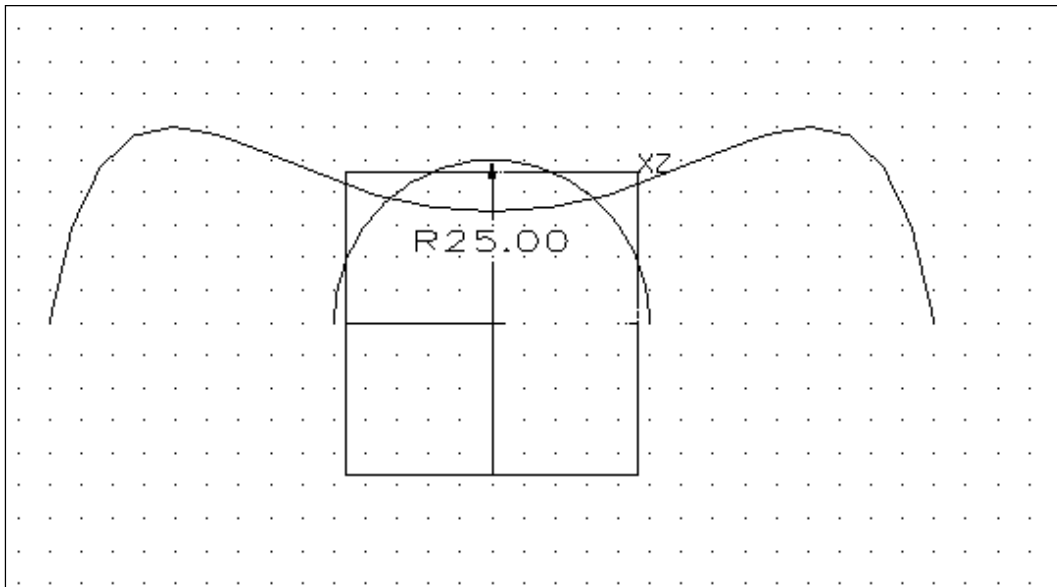
### Create an extruded face for the bottle neck

Select **Extrude** and create a new sketch on the XZ Plane.

Use the **Center**  command within the **Arc**  command under the **Sketch Tab** to create Arc.

Center point:	0 , 0
Select start point/radius/diameter:	25 , 0
Select end point/swept angle/ "to curve"	-25 , 0

(Optionally dimension and constrain arc about zero as shown below)



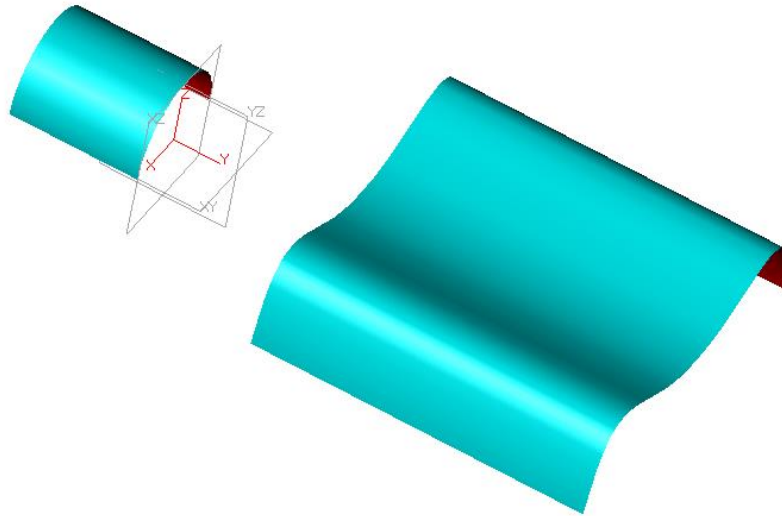
Exit Sketch or Right-click > Exit Sketch.

Continue with the Extrude:

Direction: Right Mouse / -Y Direction

Start: 0

End: 50



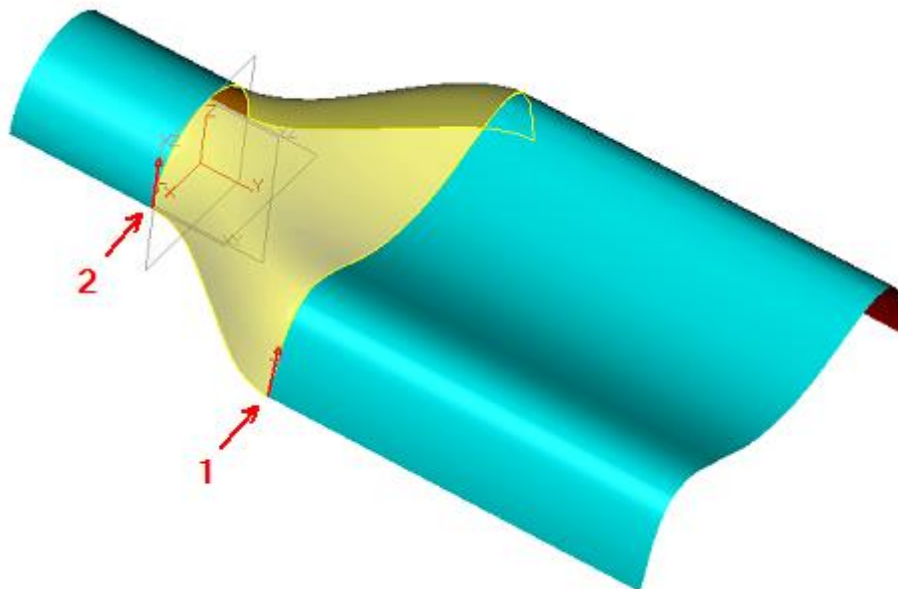
**Create a curvature continuous Loft Face between the two Faces**

**Loft (Shape Tab) > ADD.**

Pick the edges, as shown below.

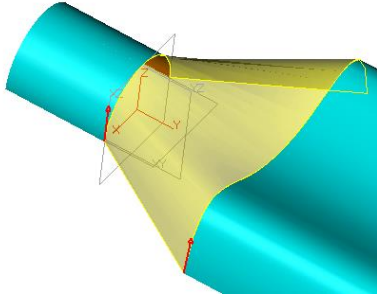
Select Continuity – Curvature from the option button for Both Ends.

The choices are None, Tangent, or Curvature.

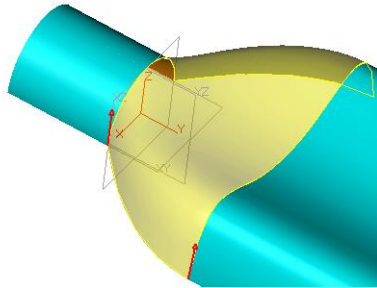


1: Start 2:End

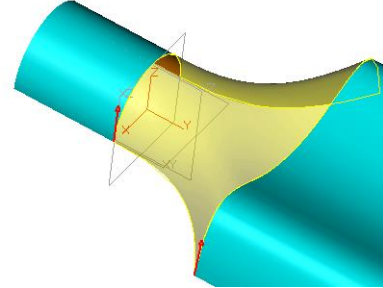
The order in which you pick the edges determines the start and end conditions. You can select a different continuity tag for each end or assign the same tag for Both ends.



"None" at Both ends



"Curvature" at the start  
"None" at the end



"None" at the start  
"Curvature" at the end

### Create a circular curve/curve face on two linear edges of the neck face

From the FreeForm Tab.

Create a **circular bi rail** face.

"None" at Both ends

"Curvature" at the start,  
"None" at the end

"None" at the start  
"Curvature" at the end

As follows:

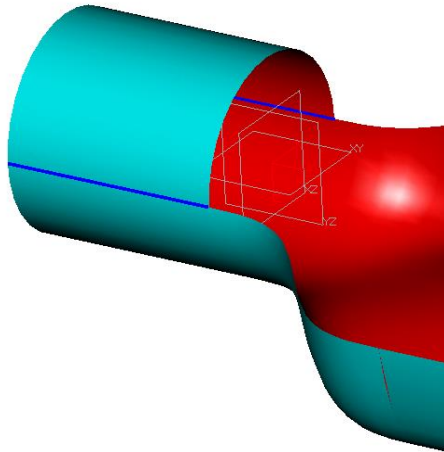
Select the first path = First linear edge of the neck face (see below).

Select the second path = Second linear edge of the neck face.

Enter the constant Radius = 25.

Select the solution number = Use the up arrow key on the menu until you see the correct face.

Note: Remember, there can be 4 possible arcs depending on the radius.



Alternatively, Undo the last feature and create an Extruded Face.

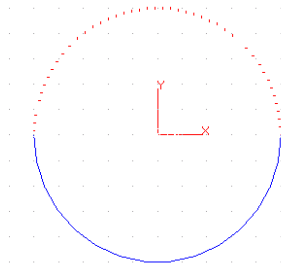
### **Extrude.**

Create a Sketch on the XZ plane.

Create a reference curve from the edge of the cylindrical face.

### **Show target.**

Create an arc opposite the reference arc.



### **Exit the sketch.**

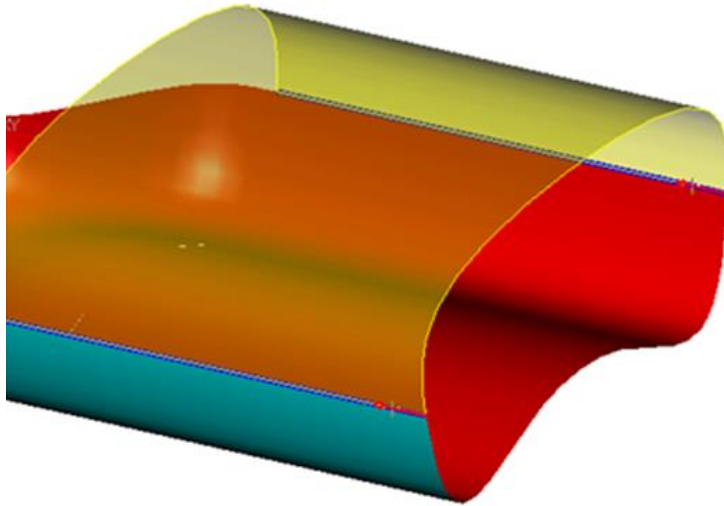
Finish the Extrude using the Right Click – **To Point** option for the Start and End points.

**Create a curvature continuous Loft on the linear edges of the bottle body face**

**Loft > ADD .**

Select edges <Enter>.

Continuity = Curvature Continuous (Continuity = Curvature) at both end.

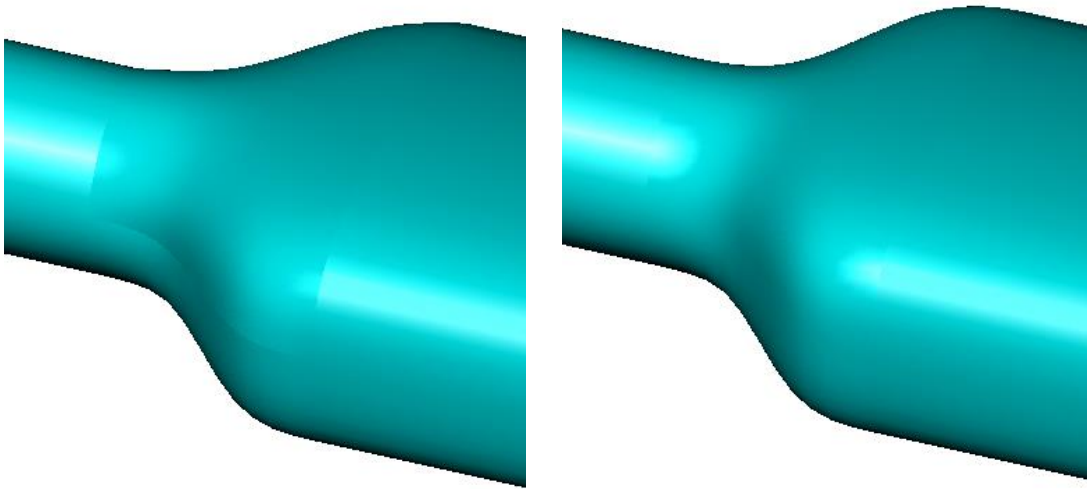


**Create a Face between the neck and the body of the bottle.**

Create an N-sided face using the **Free Form** Tab (Don't check **Tangent at boundary**).



Redefine last and now check the option **Tangent at boundary**. See the difference?



Sure you can!

Alternatively, create a mesh face.

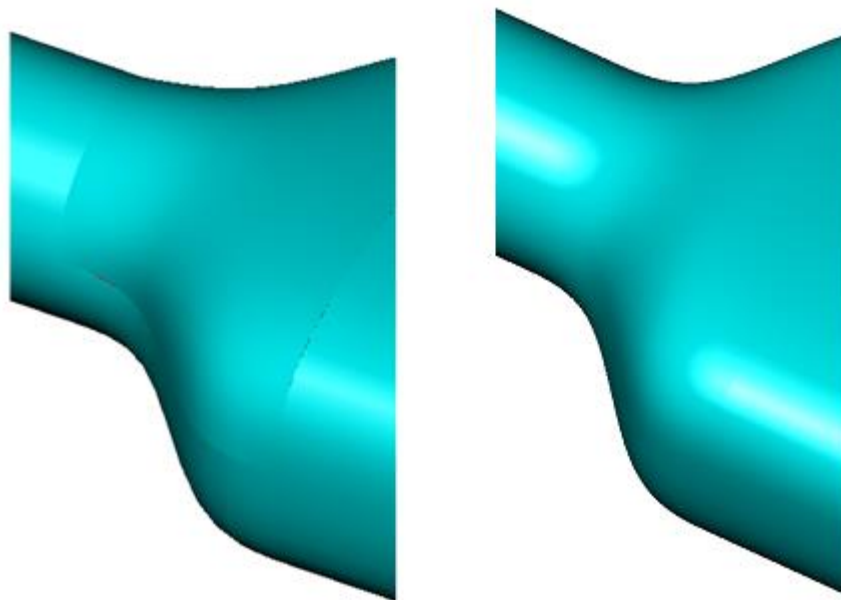
Create Faces from a mesh of curves (using NONE for continuity). ( **U/V mesh** command on the Free Form Tab)

Better would be: -

Redefine Last (using TANGENT for continuity).

Notice in both examples, when you specify tangent continuity the light reflects off the surface showing a smooth transition.

The use of **Curvature Continuous Continuity** where all 4 edges are predefined does not give as good a result as **tangent**.



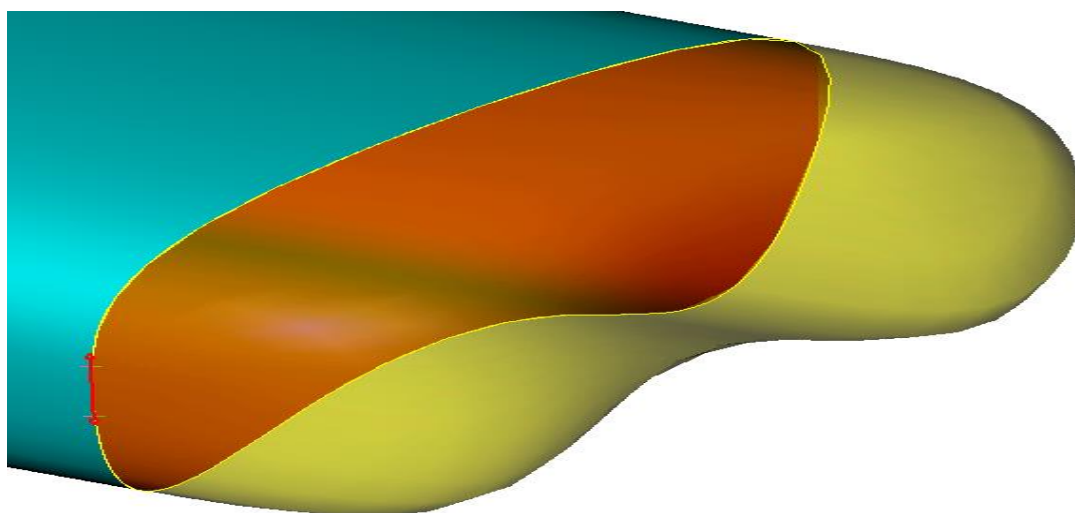
### Loft.

Create a lofted face on the end of the bottle body .



For the start, pick near the end points of both edges.

For the continuity select continuity **curvature**.

You can also adjust the weight.....or not.



Alternatively, create a planar face on the body end and then create a 10mm fillet.

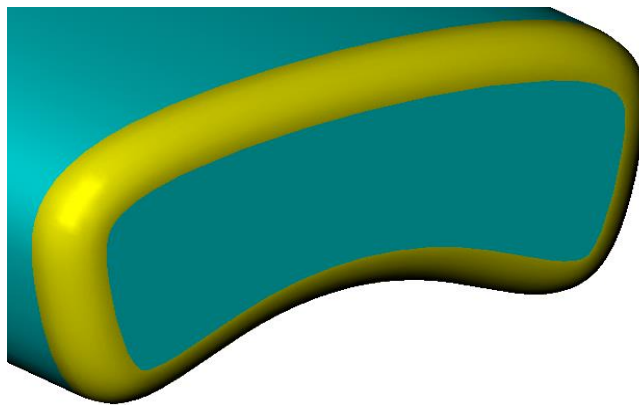
Create trimmed plane  (Free Form Tab) or  Fill Gap (Heal Tab) - both give the same results.

### Loop fillet.

Create a 10mm fillet radius on the bottom face of the bottle body.



Notice how the powerful hybrid capability allows you to work with surfaces as if they are solids.

Remember the top is still open.



Alternatively,

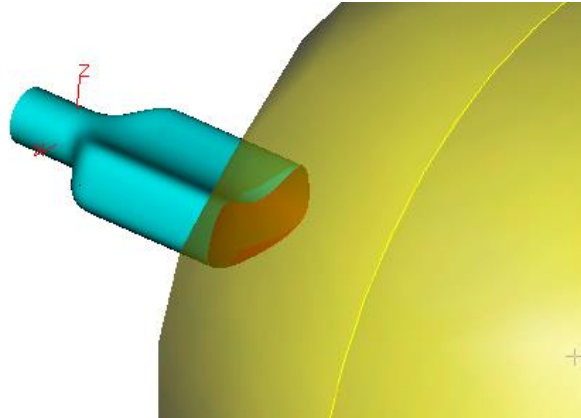
**Undo** the fillet and face creation.

Create a Sphere  with the remove option (Shape Tab) .

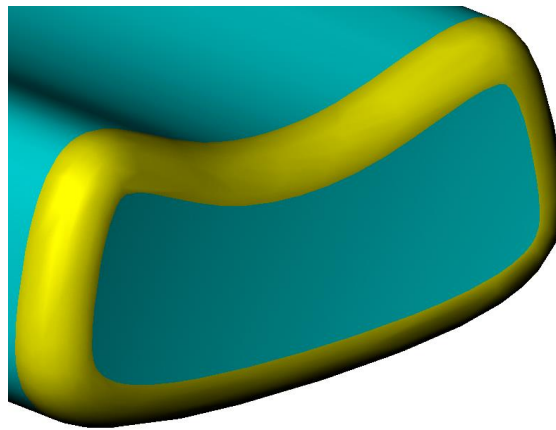
Location: 0,540,0

Radius 360





**Loop fillet radius 10.**



Create a wall thickness (Shell).

**Shell.**

Select shape(s) to shell.

Specify shell thickness = -2 mm. (Positive - to the outside, negative – to the inside)

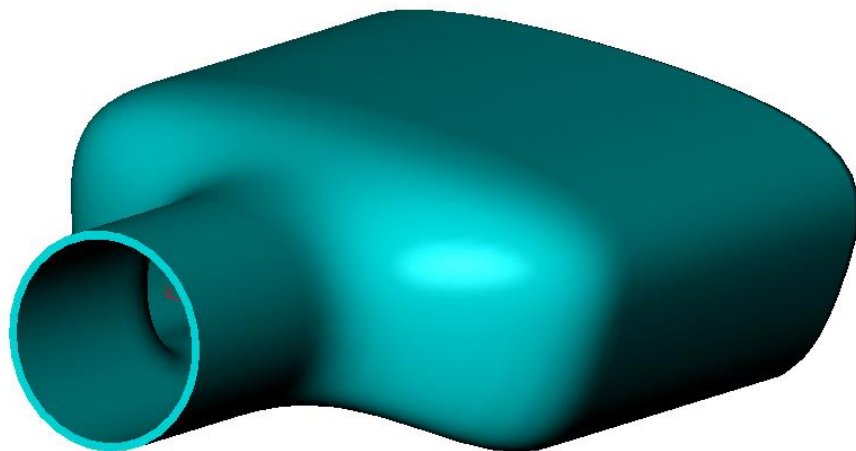
Open faces – None.

Highlighting again the power of Hybrid modeling.

**Save** your file.

**Exit** part.

End of the Bottle exercise.

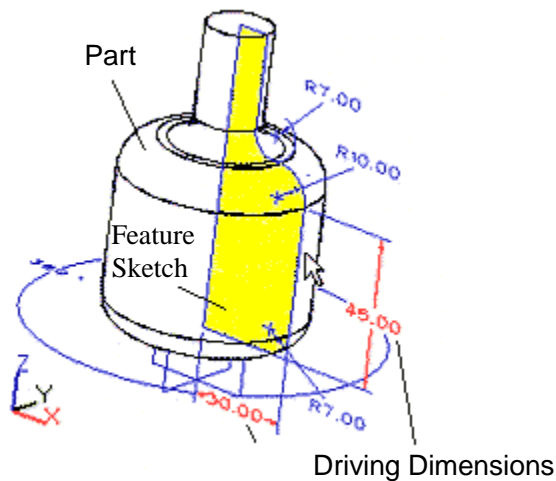


## **Chapter 8    Design Optimizer**

## Design Optimizer (3D)

**Invoked By:** Tools pull down menus (Part Level).

Use this command to optimize a part design to achieve a desired target parameter. For example you can optimize the design of a bottle to achieve a desired volume. You can select several "driving" variables and/or dimensions along with a "target" variable. When you run the Optimizer, ZW3D automatically modifies the driving dimensions and/or variables to minimize or maximize the value of the target variable or to drive it towards a specified value.



### Use the 3D Design Optimizer to drive history operations.

A numeric field of a history operation can be driven by the Design Optimizer. When prompted to select a design variable, right-click on the desired history operation input field (from the history tree) and select the "Send" option.

The Optimizer performs a complete part regen every time variables are tweaked (considered one iteration). Multiple iterations are performed so the optimization can take some time. There is a check box to disable the undo history to reduce memory consumption. There are +/- buttons and a step size value so that you can interactively tweak dimensions to drive the target variable up or down. You can also step through design optimization iteration at any time and save the optimization setup with the active part. It will be recalled automatically when the command is invoked again.

The **3D Design Optimizer Form** below shows the bottle design in the figure above being driven toward a desired volume target parameter. The variable named

"Part\_Volume" was created automatically during the [Inquire Mass Properties](#) command. The "BottleHeight" and "BottleWidth" are name tags that were attached to the driving dimensions prior to the optimization using the [Attach Name Tag](#) command. You can drive the Optimizer to reach a specified volume or determine the min or max volume based on min/max values of the driving variable or dimensions.

## Driving Variables & Dimensions

Use this section to define the driving variables and/or dimensions and their values that will be considered during the optimization (i.e., sets the criteria).



Place a check in the box next to each driving variable and/or dimension that you want to be considered part of the optimization. This makes it easier to set up the form with several variables and dimensions and then run different optimizations using sub-sets.

### Name

This displays the name of the driving variable or dimension. You can enter it into the field directly if it exists or pick the "<" button (see below) to select a dimension or browse for existing variables.

### Min, Max

Enter the minimum and maximum values to be considered for the variable or dimension specified in the **Name** field. The Optimizer will not exceed these values while converging toward the **target design parameter** (see below).


### Value

Prior to optimization this is actual value of the variable or dimension specified in the **Name** field. During optimization, this value will dynamically update indicating the values being considered.

### Step

This sets the Step size. Each time you pick the "+" or "-" buttons (see below) the **Value** (see above) will increase or decrease by this amount.



Pick this button to select a variable or dimension and add it to the optimization. You can select a dimension directly or [right-click](#) to browse for [variables](#) or show dimensions. Only variables or dimensions marked with a check  to the left of the **Name** field will be considered during the optimization.



Increases or decreases the **Value** field by the amount specified in the **Step** field. You can **Stop** the optimization, change the **Value** and then continue to **Run** the optimization. This allows you to skip to that part of the optimization with a new value.

## Target Design Parameter

Use this section to define the target design parameter for the optimization (i.e., what you are looking for such as a specific volume). During each iteration the target value is dynamically updated and displayed in the field indicated with "----->" The message field indicates the memory usage during the optimization and will signal when a successful design optimization is achieved.



Pick this button to select a variable or dimension to use as the target design parameter. Again, you can select a dimension directly or [right-click](#) to browse for [variables](#) or show dimensions. You can enter its name in the field directly as long as it exists.



Steps through one complete design optimization iteration. The Values of the driving variables and dimensions are tweaked, the part is regenerated and the value of the target design parameter is recalculated and displayed.



This runs multiple design optimization iterations until the target design parameter value is reached (without being exceeded).



Completes the current design optimization iteration (including a regen) and then stops the optimization. The latest value of the target design parameter is displayed.

## Minimize, Maximize

Select either of these options and then pick the **Run** button (see above). This will drive the optimization toward the minimum or maximum value allowable for the target design parameter without exceeding the Min/Max values of the driving variable and dimensions selected for the optimization.

## Drive To

Select this option and then enter a value in the field to the right. This will drive the optimization toward this value. The optimization will stop when this value is reached (but not exceeded) or when the Min/Max values of the driving variable and dimensions selected for the optimization are reached (but not exceeded).



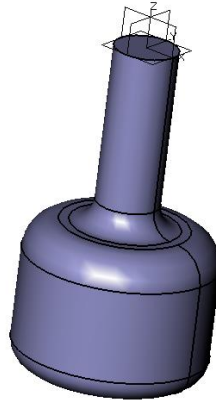
This disables undo/redo to save memory. Each iteration completed by selecting the **Step** button is considered one undo/redo operation. All iterations completed when the **Run** button is picked (until the **Stop** button is picked or when a conclusion is reached) are also considered one undo/redo operation.

For extensive optimizations this can consume considerable amounts of system memory. Check this box to disable undo/redo during optimization.

### **Save this Setup**

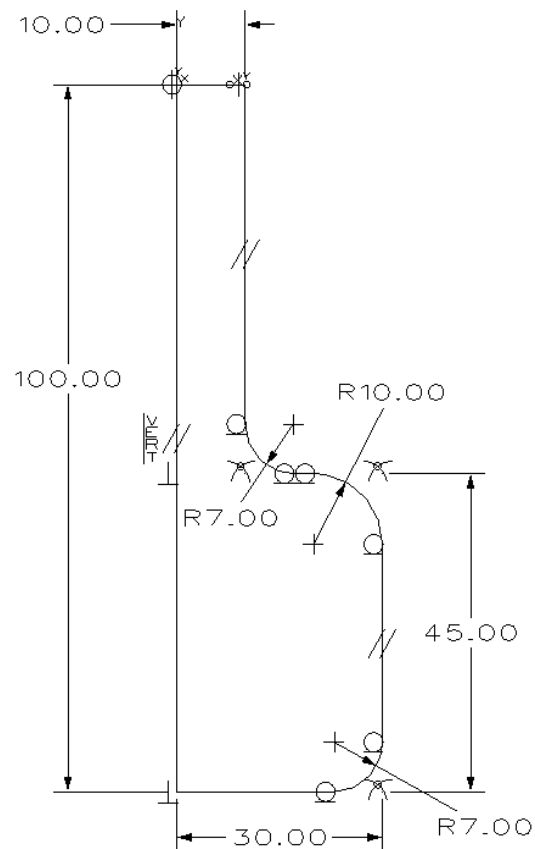
Pick this button to save the currently displayed design optimization setup. The setup is saved with the active part and will be displayed the next time this command is invoked for the part.

## Design Optimizer exercise:



Open a new file and part in mm units.

**Sketch the shape** shown below in the **XZ** plane.



Verify that it is fully constrained.



Make subtle changes to the 30 and 45 dimensions to test the robustness of your dimensioning scheme. Notice the short horizontal line on the shoulder of the bottle. Do not reduce the 30 horizontal dimension too much and lose this feature. We are going to use these two dimensions and the overall height to design a bottle with a specific volume.

Continue to the next step, but do not exit the sketch...yet.

Pick Edit pull down Edit > Edit equation to open **Equation set** form

You will now **write 3 Equations** as follows:

Height\_Total = 100 ( Name = Height\_Total, Expression = 100.mm)

Height\_Shoulder = 45 ( Name = Height\_Shoulder, Expression = 45.mm)

Radius = 30 ( Name = Radius, Expression = 30.mm)

ZW3D is a metric system. If you change the input units to Inch it is suggested that you enter the expression like 100.in

Use the Equation Input section of the menu.

Check the “**Is distance**” box.

Click **Submit Equation Input**.

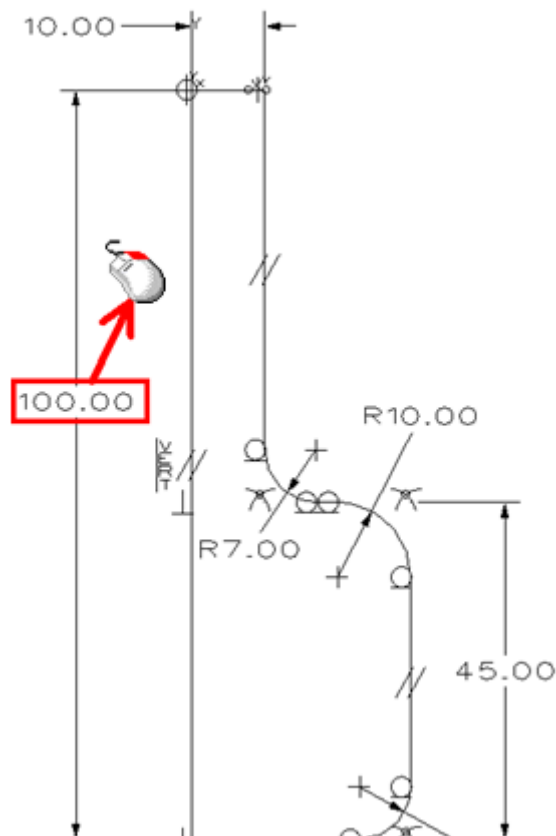
Click on **Apply**.

**Repeat** this for the other two dimensions **then click OK**.

```
Height_Total = 100 mm {100.mm} [0, 0]
Height_Shoulder = 45 mm {45.mm} [0, 0]
Radius = 30 mm {30.mm} [0, 0]
```

After your three equations are written you need to associate the dimension on the screen with the Variables you just created.

Right-click on the dimensions and pick **Link to Variable** as shown below.



The Variable Browser will open.

Pick the appropriate variable, for example, choose “Height\_Total = 100” in the Variable Browser for the “100.00” dimension above.

**Repeat** this for the other two.

Pick the **Solve Current Sketch** icon.

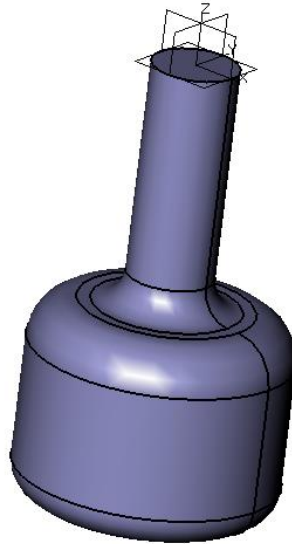
**Exit Sketch.**

**Revolve** about the **Z-axis** or the centerline.

Start Angle of 0 is default.

End Angle of 360 is default.

So just Middle-click through the command or hit **OK**.



Run the **Inquire** pull down menus > **Mass Properties** command.

Pick **Record Inquired Information**. This will automatically create a variable called **Part\_Volume**.

Close the Mass Property form.

Open the **Variable Browser**.( **Tools** pull down menus)

You will now use the Design Optimizer command to let the system massage or iterate your driving dimension to obtain a specific **Volume** for your part.


**Turn on the dimensions**. You can optionally pick them in the next step.

Press the  icons indicated above and select the driving variables from the browser or the dimensions from the screen.

See below for reference.

Name	Min	Max	Value
Height_Total	0	0	100
Height_Shoulder	0	0	45
Radius	0	0	30


Notice the boxes on the left are automatically checked. This indicates that all three variables will be used in the optimization. You can optionally uncheck them and run an optimization

Press this icon  in the “Target design parameter” area of the menu and select **Part\_Volume** from the Variable Browser.

The current volume is displayed in the field.

Click on the radio button next to “Drive To” and enter 160000.

Uncheck the “Height\_Total” and “Radius”. For the first optimization we are only going to vary the “Height\_Shoulder”.

Press the “Run” icon to start the optimization  .

Result:

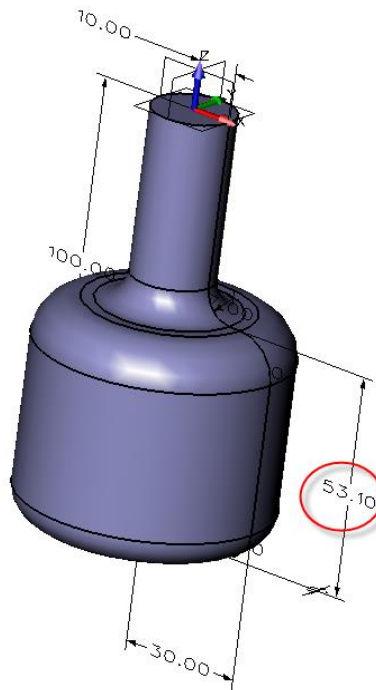
Name	Min	Max	Value
Height_Total	0	0	100
Height_Shoulder	22.5	90	53.0952
Radius	0	0	30

As the optimizer is iterating the design you will see the variable in the browser continually updating as well as the Part\_Volume in the menu above.

The first thing the system does is if you have not already specified them is to calculate the minimum and maximum values for the Driving Variable so the bottle doesn’t break...so to speak.

You can enter these values yourself before the calculation. This is especially helpful when you are iterating more than one variable. When one of the limits is reached the iteration will continue changing the other variables.

At the end of the iteration process you will see a message announcing a “Successful design iteration”. Notice the new value of the “Height\_Shoulder”.



Next , setup the Design Optimizer as shown below.

Name	Min	Max	Value
Height_Total	60	200	100
Height_Shoulder	25	90	53.0925
Radius	30	60	30

Part\_Volume = 160000, Drive to 120000.

This time we are going to create a shorter bottle but limit the “Height\_Shoulder” to be no less than 25.

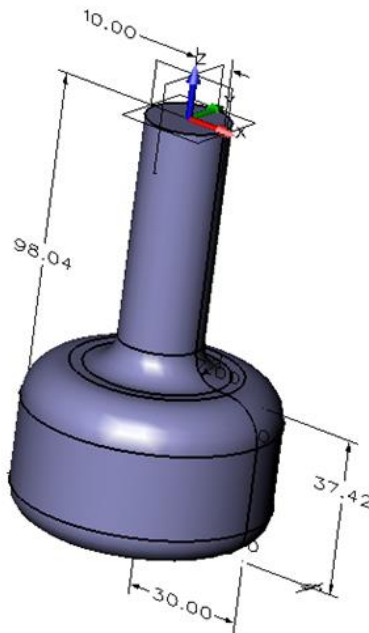
check the box next to “Height\_Total” and “Height\_Shoulder”. This time we will only exclude the Radius of the bottle in the calculation.


You may want to check **Disable Undo to save memory** in the lower left to save memory if you feel that would be a concern. If you check the box you will not be able to use the Undo/ Redo feature but your memory usage will be minimized.

Press the “**Run**” icon to start the optimization.

Result should be:

Name	Min	Max	Value
Height_Total	60	200	98.0412
Height_Shoulder	25	90	37.4245
Radius	30	60	30



During the iteration process the “Height\_Total” did calculate a little under 100. If you want to round it off to 95, set the “step” size to 3.0412 and hit the minus sign . Watch the part.

With the height adjusted to 95, uncheck the Height\_Total box and recalculate a new Height\_Shoulder.

It will now settle in at 37.80.

## **Chapter 9    Point Cloud**

(Only for Professional Version&Premium Version)

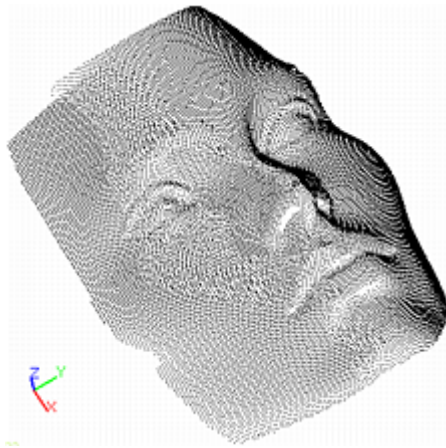
## ScanShape Tools (Point Clouds) (3D)

### Creating Surfaces from Point Clouds

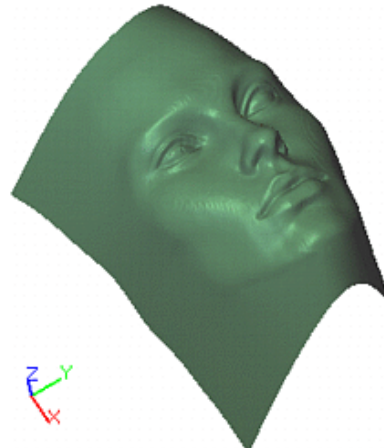
Invoked By: [Point Cloud Tab \(Part Level\)](#)

These commands create surfaces that pass through a set of points referred to as a point cloud. You can create the point cloud using ZW3D point commands or use a point cloud file. Imported STL data can also be used. See **Tip & Techniques** below. Both patch and cylindrical faces with a shared seam are supported. The commands function in a similar manner and share many of the same optional inputs.

These commands can be used to reverse engineer designs using CMM (Coordinate Measuring Machine) point data to drive ZW3D features. The **References** section below contains links to other commands that work with point clouds and CMM data.



**Point Cloud (30,900 Points)**



**Surface (2,808 Control Points)**

### Fit Face

Use this command to create a trimmed face that passes through a set of points or a point block referred to as a point cloud. With this approach you can divide up an overly complex set of point data into separate sections. Instead of one very large surface, several smaller, tangentially connected patches may be generated. Each patch can be based on a portion of the input point data for which a common projection direction does not exist.

This should both speed up the overall process and also enable the processing of "360" degree scan data. The only required inputs are the input points the face must pass through and the set of boundary curves used to trim the face. This command can be used on imported STL data. Refer to the Optional Inputs section below.





## Required Inputs

### Points

Specify a set of points or a point block.



## Optional Inputs

### Normal

Use this to define the base face direction. This allows you to re-orient the overall direction of the fitted shape, in case the built-in method does not detect it properly or you wish to use a different direction as a preference. Occasionally, less than ideal scan data can cause such ambiguities.

### Curves

Specify the set of boundary curves. So, the trimmed surfaces are created by the points and the boundary curves. Collect points and Tangent at boundary can be checked only if this option isn't empty.

### Collect points

This option is available only if Curves field is not empty. Check this option to collect points to fit face.

### Side

Select a point specifying the side of the boundary curves that you wish to collect the points from.

### Face

Select an untrimmed base face to modify. If a face is selected, edges are handled automatically such that the tangency is largely maintained across a seam. The continuity across a seam edge depends on the point data near the edge.

### FEM Settings

Spring Constant, Bending Resistance

These commands create surfaces based on energy minimization. Use the slider bars to set the spring constant and bending resistance values. See Finite Element Method (FEM) for Surface Creation for more information.

### Tangent at boundary

Check this box to enforce tangent continuity across the resulting surface boundary.

### U degree, V degree

This is the degree of the resulting surface. Lowering the degree values will generally result in a simpler surface.

**Specify sample density**

Use this to set the sample point density.

If you check this box

In the Density field, enter the average number of sampling points on the boundary curves. ZW3D then computes the actual number of sampling points according to the relative length of each curve with regards to the total length of all curves.

If you do not check this box

ZW3D automatically uses a discretization method for sampling the points on the boundary curves.

**U-direction, V/U ratio**

The U-direction is used in conjunction with the U/V ratio to better distribute the available control points. Once the U-direction is specified, the U/V ratio can be used to control how many points there are per row in the U and V directions. The allowed value range is 0.01-100.

For example, the resulting face is to be about 10 times longer in the V direction. Increasing the U/V ratio to 10 will assign more control points to the V direction. The maximum value of 100 will work for even larger ratios since most of the control points will be assigned in the V direction anyway. Refer to the Notes section below for more information.

**Keep curves**

Check this box to Keep curves selected during the Required Inputs. Otherwise it will be deleted.

**Refit base face**

When the Face option (see above) is used, check this box to refit the base face during the command. This means that more control points can be added to the surface, if necessary, to better fit the point cloud data.

If unchecked, you can prevent the command from adding more control points before the surface is fitted through the point data. It will use existing control points as degrees of freedom and try to fit the data.

This will speed up the process and will also keep the same UV surface parameter values for the sample data points throughout multiple surface fitting iterations. This helps to prevent a "shifting" or "pinching" effect that can arise if slightly different UV parameter values are assigned during each iteration.



## Tip & Techniques



### Point clouds containing 50 million or more points

If you regularly work with point clouds containing 50 million or more points, please contact ZW3D technical support. They may have solutions that can help process your point clouds more efficiently.



### You can create surfaces from imported STL data

An imported STL object can be selected as input for the Face from Point Cloud command. It is important to note that this is not a general purpose STL to NURBS conversion and/or translation tool. It can be used to that effect, however, for certain supported cases.

The supported cases include those that the Surface from Point Cloud command supports. Base faces can still be used, including ones with seam edges. You will need to provide a base face in cases where the STL data cannot be projected to a common plane adequately. See Finite Element Method (FEM) for Surface Creation under the Notes section for more information.

## Tessellate



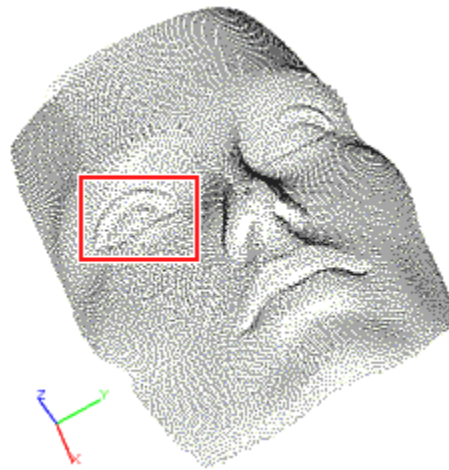
Invoked By: [Point Cloud Tab \(Part Level\)](#)

Use this command to tessellate 3D point data into inter-connected and therefore traversable vertices and edges. The connected edges form a series of triangles. If a set of STL data is tessellated it will result in the same exact set of triangles but they will be fully inter-connected and sewn together (i.e., no overlapping edges or redundant vertices.) This provides a structured way of traversing random input data for further processing. You can set or edit tessellated/STL block attributes by selection **Attributes > Face or Edit > Attributes**.

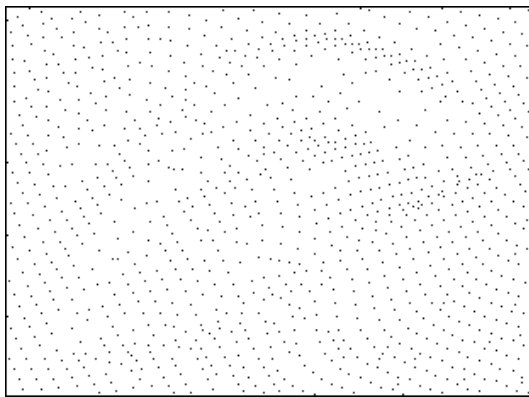


### You can Export STL Blocks

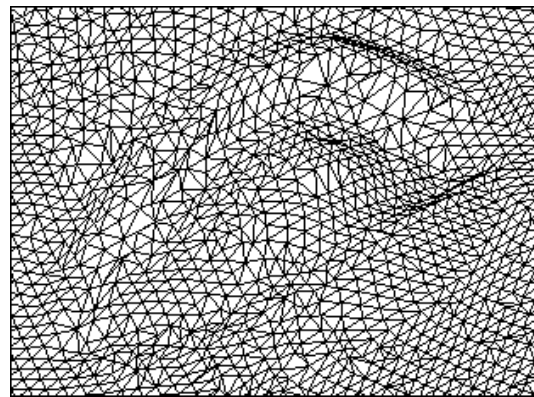
ZW3D provides export functionality for STL block objects. STL block objects, usually created via STL import or tessellating point cloud data, can be exported back as a standard STL file. This functionality supports any combination of multiple STL blocks and B-Rep solids in a part. The STL Export "1st quadrant" option, as well as both binary and ASCII modes are supported. This allows you to selectively export STL objects by blanking the unwanted ones (i.e., i.e. if an STL block is blanked, it will NOT get exported).



### Scattered Data to Tessellate - With Area of Interest



**Scattered Data to Tessellate - Zoomed in Area**



**Tessellated Data - Zoomed in Area**



### Optional Inputs

#### Delete input points

Check this box to delete the points used as input for this command. Only the tessellated triangles will remain.

#### Use maximum STL detail

Check this box to tessellate at the original (maximum) detail level of the imported STL object. Otherwise the detail from the current zoom level is used.

#### Method

**Vertex** - This method leaves the input point data intact, inter-connecting them to build the triangulated faces.

**Uniform Grid** –This method builds a small grid of uniform facets that are then used to find a closely fitting overall surface. The resultant vertices of the triangulation do not necessarily pass through the original input points.

The main difference between these two methods is whether or not the original vertices are preserved. The results of each method will depend on the input data. As a rule, the **Uniform Grid** method will do a better job producing closed solids from very complex data and provides additional options. However, some of the "gridness" will be apparent if the data is sparse enough. Also, with the **Uniform Grid** method vertices are not preserved and the planes are "close" fits only. With dense enough data, the **Uniform Grid** method will be very close to the original set of points.

### **Assume no undercuts**

If the input point data can be projected along one direction without interfering with each other, this option will utilize an optimized **Uniform Grid** method for much faster results.

### **Resolution**

For the **Uniform Grid** method (see above), the resolution of the "grid" can be set from coarse to fine.

### **Custom**

This allows you to specify the "resolution" input using a numeric distance value rather than a slider setting. Use this option if the coarse to fine setting of the slider does not produce the desired results. This value will always override the slider setting.

### **Smoothness**

For the **Uniform Grid** method (see above), the smoothness of the resultant facets can be set from smooth to sharp.

### **Neighbors**

This option only applies to imported point cloud data. This command needs to determine the number of edges that can connect points before triangles can be formed. This option lets you control how many neighbor points are considered for generating "candidate" edges per a given point. The more "neighbors" allowed, the more candidate edges considered, which in return allows more combinations of valid triangles. Also, when more "neighbors" are allowed, less missing triangles are present, but it also decreases computation speed.

### **Fill remaining holes**

After tessellation, some remaining holes can be filled to build a "water tight" solid.

### **Max. Edges**

Use this option to define the holes for the **Fill remaining holes** option above. Only holes that consist of a specified number of tessellated edges or less will be filled.

## To Keep

After tessellation, there may still be some undesirable facets that form one or more completely separate bodies. If you set this option to zero (default) the routine will determine which ones to keep and which ones to delete. If a positive number is specified, the routine will keep the same number of bodies (in the order of overall size) and delete the rest.

## Analyze Point Cloud Data Surface Fit



### Invoked By: [Point Cloud Design Tab](#)

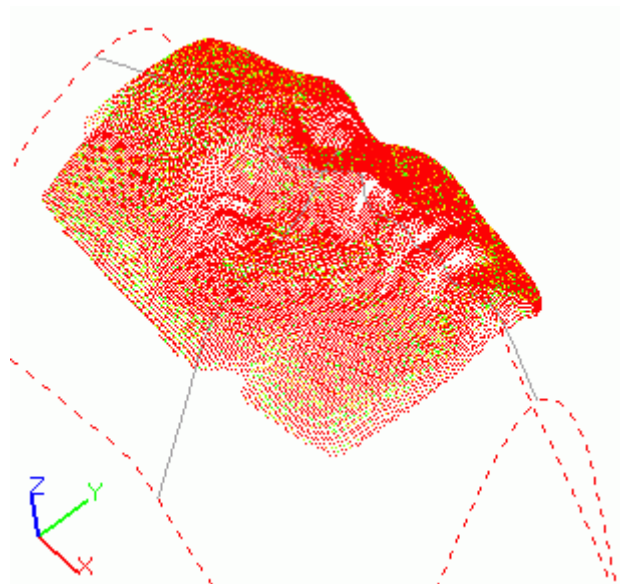
Use this command to analyze the distance between individual points in a point block and a given surface that was fitted through the point block. The ratio of a given point distance to the surface and the user specified max. distance range value determines the color of a point.

A "perfect fit" (or ratio value of 0.0) is green, a "1/2 of max range" (or ratio value of 0.5) is yellow, and a distance outside the max. range ( $\geq 1.0$  ratio value) is red. Any value in between will be mapped to the standard linear RGB color gradient of green->yellow->red.

This command also shows the min/max/average distances in the **message area**.

If STL or tessellated data is selected as the "input points" they are first blanked before the resultant color mapped vertices are displayed. They can be accessed by unblanking them.

.





## Optional Inputs

### Color Range

This shows the color range values set by Max. Range below.

### Max. Range

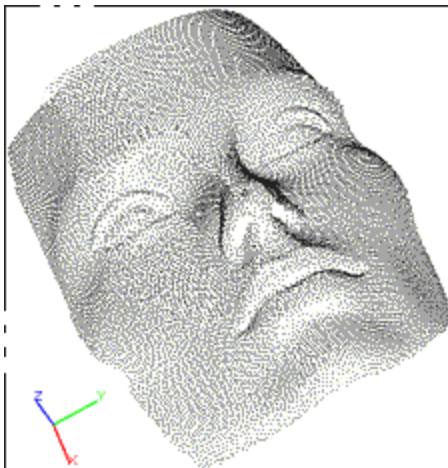
This sets the range of distance values that can be shown with the available gradient color. Any measured distance equal or larger than the value set in this field will be shown in red. You can use this to set the analysis threshold.

### Reduce Point Set

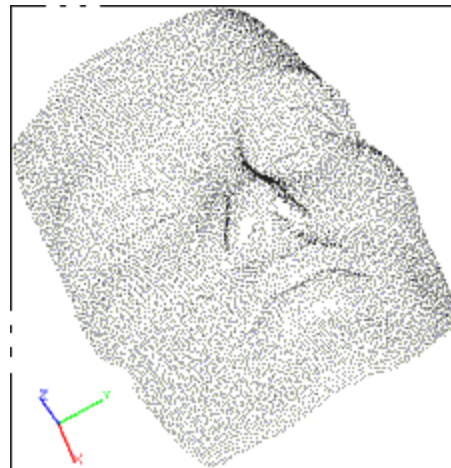


Invoked By: [Point Cloud Design Tab](#)

Use this command to reduce the total number of points in a point cloud. The points can be created by ZW3D or imported from CMM (Coordinate Measuring Machine) data. First select all of the points to consider. Optionally, you can set the reduction tolerance, keep the original points and create a point block for a more compact method of storing the point data. See below for more information.



Before (30,900 Points)



After (16,375 Points)



## Optional Inputs



## Manual/Automatic

The **Manual** method allows you to set the **Tolerance** for removing points. The **Automatic** method is determined by ZW3D.

## Tolerance

Used with the **Manual** method above. This command spatially sorts points and keeps only one point in each tolerance-sized region. If the tolerance is set to zero or left blank, then a guess is made at a reasonable tolerance. In a sample point file, a tolerance of 0.01 reduced the total number of points by 85% (from 17,697 to 2,761). However, fewer points mean less accuracy but greater system performance. For lower tolerances, use the **Create point block** option below.

## Use uniformly spaced points

Check this box to keep all remaining points uniformly spaced apart.

## Delete input points

Check this box if you do not want to keep the original point cloud. Otherwise, all of the original points will be kept.



With a little preparation, you can use different layers to help manage point cloud data before using this command. Using the **Layer Manager**, create a new layer with a name like "**Original Point Cloud**" and assign it to the original point cloud. Then create a new layer with a name like "**Reduced Point Cloud**" and make it the active layer. Then when you execute this command the point clouds will be kept separate and manageable.

## Create point block

Check this box to create a 3D point block of the reduced point cloud. This is a more compact form for storing point data. "**Block**" is listed as a Pick Filter list for move, erase, blank and inquire commands. A point block is not created for the original point cloud.

## Group into Point Block



Invoked By: **Point Cloud Tab**

Use this command to group a set of points and/or point blocks into a new point block. Optionally, the input points and/or point blocks may be deleted after the point block is created.

This command adds flexibility in handling a large quantity of point data by allowing you to group certain points into one object called a "point block." You can do this at any time, not only during the [Import Points from File](#) and the [Reduce Point Set](#) commands.



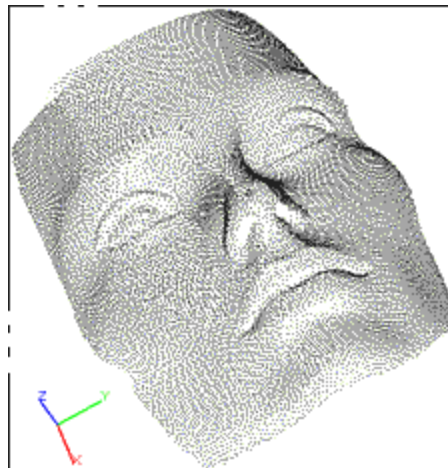
This can be used for situations when raw point cloud data needs editing before it can be "committed" to a point block (i.e., noisy data removal), or when a set of points needs to be separated into more than one region/section on the fly and so forth.

You can set or edit point block attributes by selection **Attributes > Point** or **Edit > Attributes**.



### Create a Point Block from Imported STL Data

You can create a point block from imported STL data using this command. Once created, you can then edit the point block using all of the current point block editing commands such as Explode Point Block and Remove Points from Block. The STL data can also be edited during this command using the Optional Inputs to **Delete** or **Reduce** (i.e., thin out) the STL point data.



**30,900 Point Entities = 1 Point Block**



### Optional Inputs

#### Delete input points

Check this box if you want to delete the original point cloud. Otherwise, all of the original points will be kept.



### Use Layers to manage point block data

If you plan to keep the original point cloud, you can use different layers to help manage point cloud data. Do the following before using this command. Using the **Layer Manager**, create a new layer with a name like "Original Point Cloud" and assign it to the original point cloud. Then create a new layer with a name like "Point Cloud Group" and make it the active layer. Then when you execute this command the point clouds will be kept separate and manageable.

### Reduce point set

Check this box to reduce the total number of points in the block and enter a reduction tolerance below.

### Manual/Automatic

The Manual method allows you to set the Tolerance for removing points. The Automatic method is determined by ZW3D.

### Tolerance

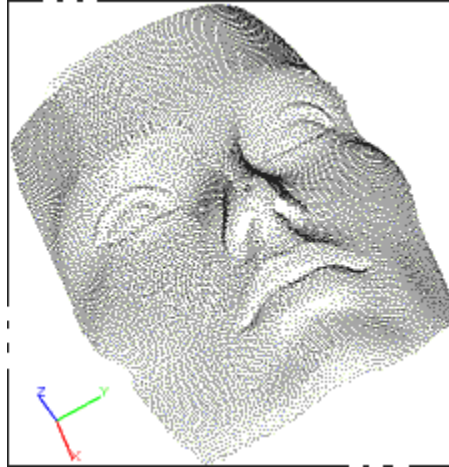
Used with the Manual method above. This command spatially sorts points and keeps only one point in each tolerance-sized region. If the tolerance is set to zero or left blank, then a guess is made at a reasonable tolerance. In a sample point file, a tolerance of 0.01 reduced the total number of points by 85% (from 17,697 to 2,761). However, fewer points mean less accuracy but greater system performance.

### Explode Point Blocks



Invoked By: **Point Cloud Design Tab**

Use this command to explode (i.e., separate) one or more point blocks so that all points are once again separate entities. This allows you to edit individual points, reduce the point cloud or separate the point cloud into individual regions. You can then regroup the points using the Group Points into Block command.



1 Point Block = 30,900 Point Entities



### Optional Inputs

#### Delete input point blocks

Check this box if you want to delete the original point block.



If you plan to keep the original point block, you can use different layers to help manage point cloud data. Do the following before using this command. Using the **Layer Manager**, create a new layer with a name like "**Original Point block**" and assign it to the original point block. Then create a new layer with a name like "**Exploded Point Block**" and make it the active layer. Then when you execute this command the points will be kept separate and manageable.

#### Remove Points from Point Blocks



Invoked By: **Point Cloud Tab**

Use this command to remove (split) a set of points from one or more existing point blocks. The points are "picked" by enclosing them with a series of resizable spherical volumes. An optional input is available to delete the selected points (i.e., trim the point block). Otherwise the selected points are retained as separate point blocks. This command works across multiple point blocks. Each will be split or trimmed as desired.



### Optional Inputs

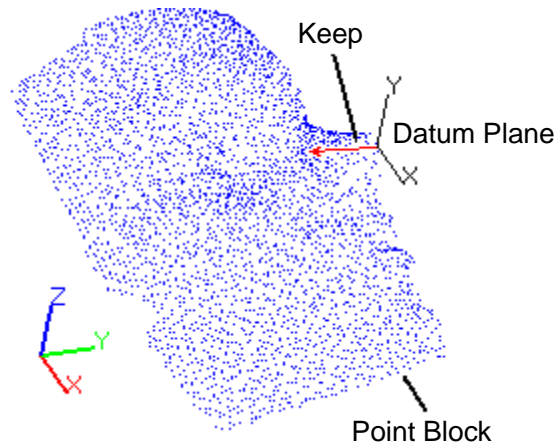
**Radius** - Specify the radius within which points are selected. When you select a point, all points within this spherical radius will also be selected.

**Delete input points** - Check this box to delete the selected points. Otherwise the selected points are retained as separate point blocks.

### Trim Point Blocks to Planes

Invoked By: **Point Cloud Tab**

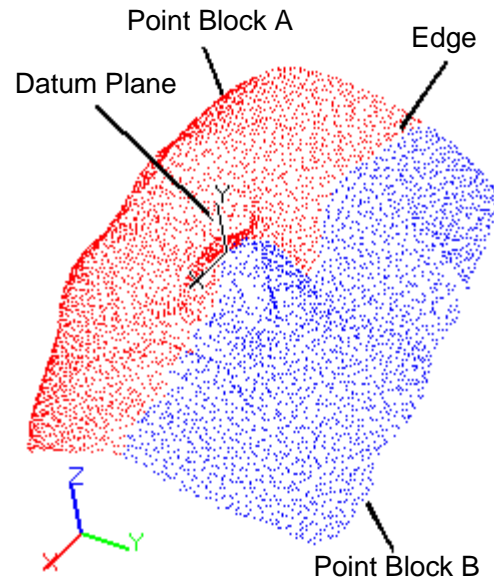
Use this command to trim a point block to a datum plane, planar face or a sketch. First select the point block then select the plane. An arrow will indicate what side of the point block will be kept. You can change the direction of the arrow using the **Flip side to keep** option. Any number of point block objects may be split simultaneously via any combination of datum planes, planar faces, and sketches.



### Split Point Blocks with Planes

Invoked By: **Point Cloud Tab(Part Level)**

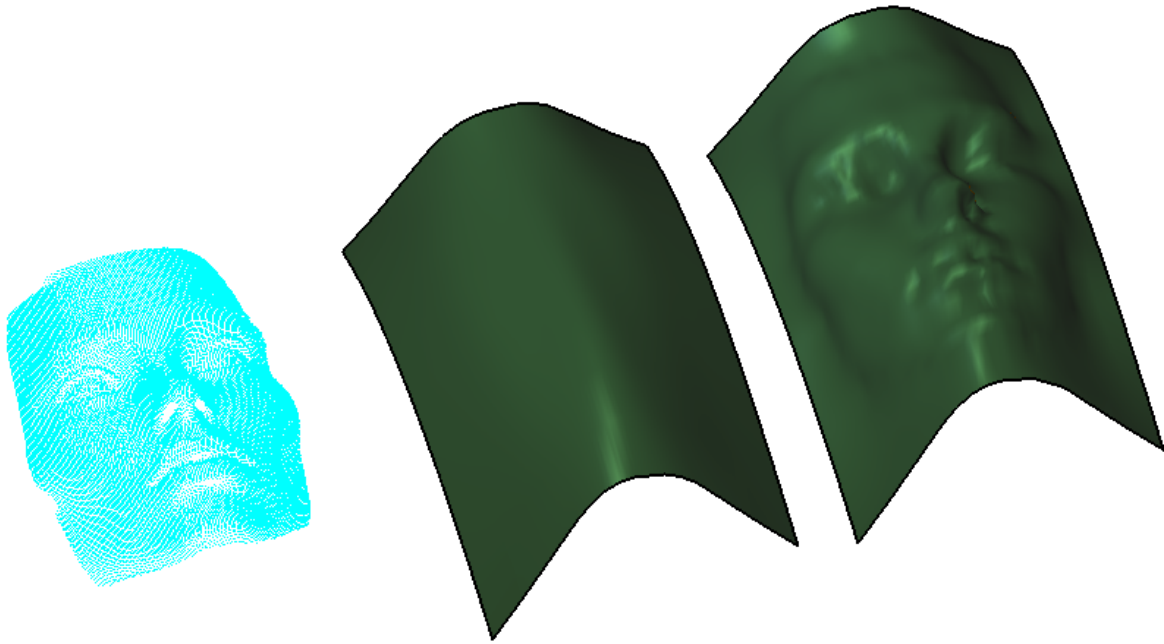
Use this command to split a point block. First select the point block then select a plane. Any number of point block objects may be split simultaneously via any combination of datum planes, planar faces, and sketches.



## Point Cloud exercise:

Clear the current session and open the file **face.Z3** in the training folder.

Edit the part **01\_face**.



This part shows the progression from scanned points to a ZW3D surface model.

## **Conclusion.**

We hope that you found this training course beneficial, any suggestions or improvements that you may have to the course or course material will be gratefully received.

On a final note, please take the time as often as possible to study the HELP Manual. This should always be your first step whenever you come across any questions.