# Conform Modifier

-  Modify panel > Make a selection > Modifier List > Object-Space Modifiers > Conform
- Default menu: Make a selection > Modifiers menu > Parametric Deformers > Conform


The Conform modifier lets you move a spline or mesh onto the surface of one or more other meshes.

The Conform modifier performs actions parametrically, letting you change the relationship between the conform object and the target surface after the initial conform operation has been performed. This lets you manipulate the look and feel of your model without any explicit modeling, like bending or shaping. As a result, you save time because basic modeling steps, like roughing out geometry and performing manual refinements, are eliminated.

You can also animate your deformations created with the Conform modifier. For example, a conform object can be animated as it moves across a target surface.

Two projection methods, Volume and Shrink Wrap, control how the spline or mesh (the conform object) is moved onto the surface of the other mesh(es) (target object(s)):

- **Volume**: Tries to preserve the overall shape of the conform object after it's projected onto the target mesh.
    - *Example*: Attaching a flange to a pipe.

The Volume method enables the base of the flange to conform to the pipe, without deforming the top portion of the flange.

- **Shrink Wrap**: Shrinks the entire conform surface around the target, like wrapping a piece of cellophane (the conform object) around another object (the target object). As a result, the volume of the conform object is not preserved. Between the two different methods, Shrink Wrap is the most commonly used.
    - *Example*: Attaching text to a target object, like a logo to the back of a car or a label on a bottle.

**Rollouts**

For more information about Conform modifier parameters, see Conform Modifier Rollout and Conform Modifier Advanced Rollout.

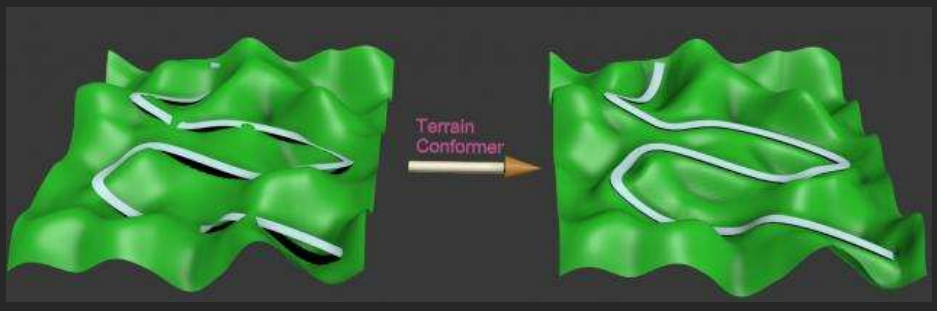
**Procedures**

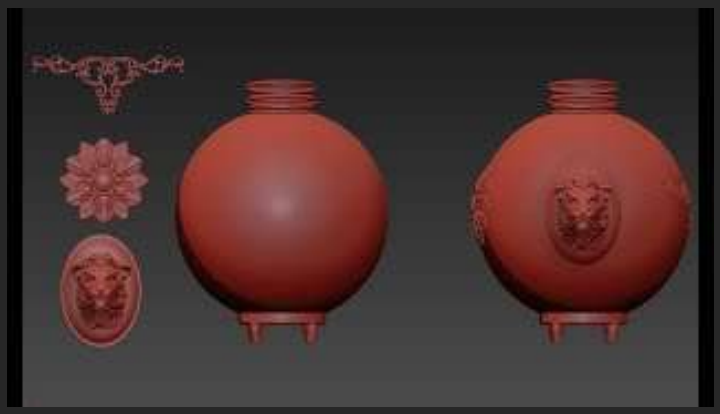
The following workflow examples illustrate how to use the Conform modifier. They highlight the two projection methods and use parameters from different Conform modifier rollouts:

- Conform objects to a surface using the Conform modifier (Volume projection method)
- Create a handle with bands using the Conform modifier (Shrink Wrap projection method)
- Create a sphere traveling through a tube using the Conform modifier (Shrink Wrap projection method)

**Notes**

- The target object and conform object, whether spline or meshes, must contain enough vertices or faces to deform.
- Conform objects can be linked or attached to target objects using a surface constraint.



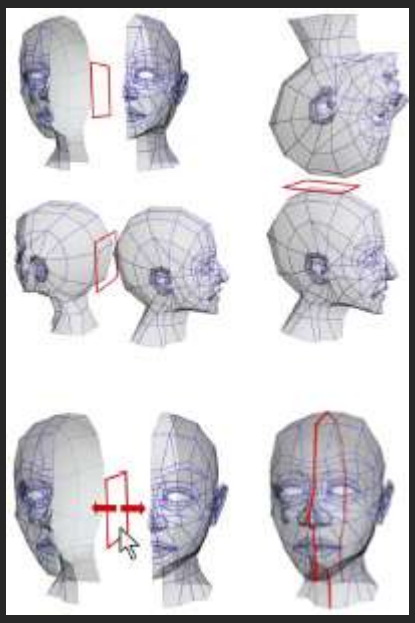
# Symmetry Modifier

The Symmetry modifier is especially useful when modeling characters or building ships or aircrafts.

- Modify panel > Make a selection. > Modifier List > Symmetry
- Default menu: Make a selection. > Modifiers menu > Mesh Editing > Symmetry
- Alt menu: Make a selection. > Modifiers menu > Geometry (Convert to Mesh) > Symmetry

This modifier is unique in that it allows you to perform three common modeling tasks:

- Mirror a mesh about the X, Y, or Z plane.
- Slice a mesh, removing parts if necessary.
- Automatically weld vertices along a common seam.



Examples of using Symmetry with different mirror axes or by moving the mirror gizmo

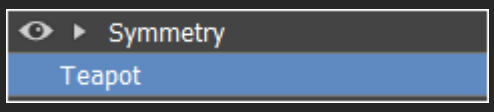
You can apply the Symmetry modifier to any geometry, and you can animate the mirror or slicing effect by animating the modifier's gizmo.

When the Symmetry modifier is applied to a mesh, any edits you make to the original half of the mesh *below the Symmetry modifier in the stack* also occur interactively to the other half. For an example, see the second procedure, below.

**Note:** The Symmetry modifier converts patch and NURBS objects to mesh format in the modifier stack; editable poly and editable mesh objects remain in their original format.

**Interface**

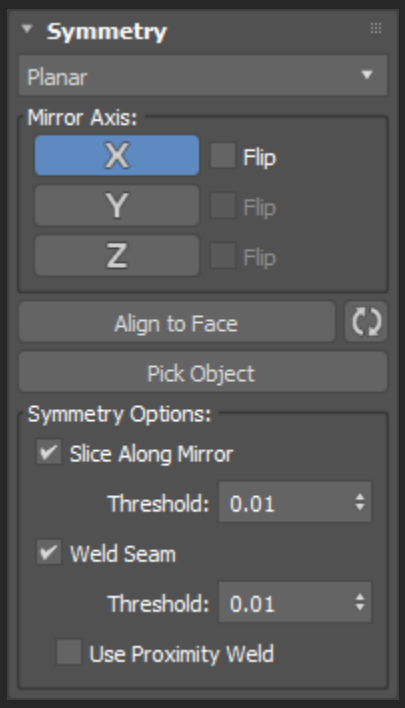
**Modifier Stack**




# Mirror

The placement of the mirror gizmo delegates how the object will be affected by symmetry. You can move or rotate, as well as animate the gizmo.

For more information on the stack display, see Modifier Stack.




## Parameters rollout

### Symmetry Type menu

Choose the slice type, either Planar or Radial.

### Mirror Axis group

### X, Y, Z buttons

Mirrors the object along the selected axes.

**Note:** X is the default axis.

### Flip (Planar type only)

When enabled, flips the symmetry to the opposite side of the corresponding axis.

### Align to Face button

Selects the face used to align the gizmo.

### Reset Gizmo button

Resets the gizmo to its default position.

### Pick Object button

Selects a reference object from the scene.

### Symmetry Options group

### Count (Radial type only)

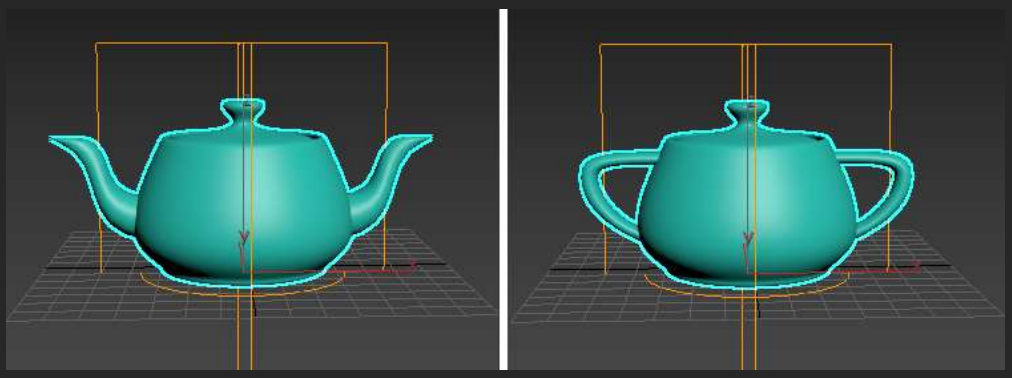
Sets the radius of the symmetry.

### Mirror Symmetry (Radial type only)

When enabled, mirrors half of the geometry to generate clean seams.

### Flip (Radial type only)

When enabled, flips the mirrored half of the geometry to the opposite side.



Mirror Symmetry with Flip option enabled and disabled

**Slice Along Mirror**

When enabled, causes the mirror gizmo to act as a slice plane when located inside the boundaries of a mesh. When the gizmo is outside the boundaries of a mesh, the symmetrical reflection is still treated as part of the originating mesh. If Slice Along Mirror is disabled, the symmetrical reflection is treated as a separate element of the originating mesh. You can also adjust the Threshold spinner to control the distance of the welded vertices when the mesh is cut. Default=on.

**Weld Seam**

When enabled, automatically welds the vertices along the mirror axis. You can also adjust the Threshold spinner to control the distance of the welded vertices when the mesh is cut. Default=on.

**Proximity Weld**

When enabled, Proximity Weld uses the old method of welding vertices based on proximity. Whereas, the new default method keeps track of the vertices that are generated from the cutting and welds them in a smarter way. Default=off.

**Procedures**

**Example: To apply the Symmetry modifier to an object:**

1. In the Perspective viewport, create a teapot.
2.  Apply the Symmetry modifier.

The teapot appears to have two spouts.

3. In the modifier stack, click ⊞ (the plus-sign icon) to expand the Symmetry modifier hierarchy, then highlight Mirror.

The mirror gizmo acts as a slice plane when it is within the boundaries of the object

4. With Mirror Axis set to X, click and drag the mirror gizmo along the X axis.

Dragging right slices more of the teapot until there is nothing visible. Dragging left causes a second teapot to appear. When the mirror gizmo is moved beyond the boundaries of the original mesh, it acts as a mirror plane showing you two complete teapots.

**Example: To perform box modeling with the Symmetry modifier:**

1. In the Perspective viewort, create a box primitive, and then convert it to Editable Poly or apply the Edit Poly modifier.
2. If necessary, press F4 to activate Edged Faces display mode.
3.  Apply the Symmetry modifier.

Other than the new edge loop created by the modifier, the box's appearance doesn't change, because it's already symmetrical.

4. On the modifier stack, click ⊞ (the plus-sign icon) to expand the Symmety modifier hierarchy, and then highlight Mirror.

5. In the Front viewport, with Mirror Axis set to X,  move the Mirror gizmo in either direction on the X axis.

Only the left-hand box moves: This is the copy created by the Symmetry modifier.

6. Position the Mirror near the left side of the original box, so the two copies are merged.

7. On the modifier stack, go to the Edit/Editable Poly level and highlight the Vertex sub-object level.

If you no longer see the Symmetry copy of the box, turn on  (Show End Result On/Off Toggle).

With Show End Result on, you might see an orange wireframe “cage” that shows the edges of original object. This is on by default for editable poly objects, but off by default for the Edit Poly modifier. The Show Cage toggle for editable poly objects is on the Subdivision Surface rollout, and for Edit Poly it's on the Edit Poly Mode rollout.

You can also see that only the vertices of the original object are visible; the vertices of the symmetry object can't be transformed directly.

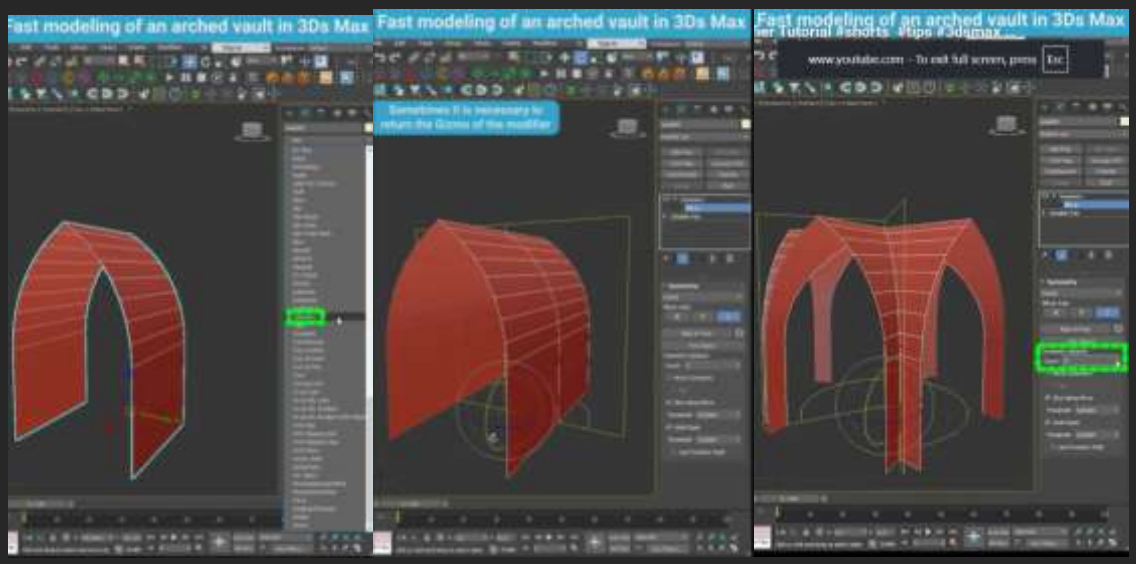
8.  Move one of the visible vertices on the right side of the box.

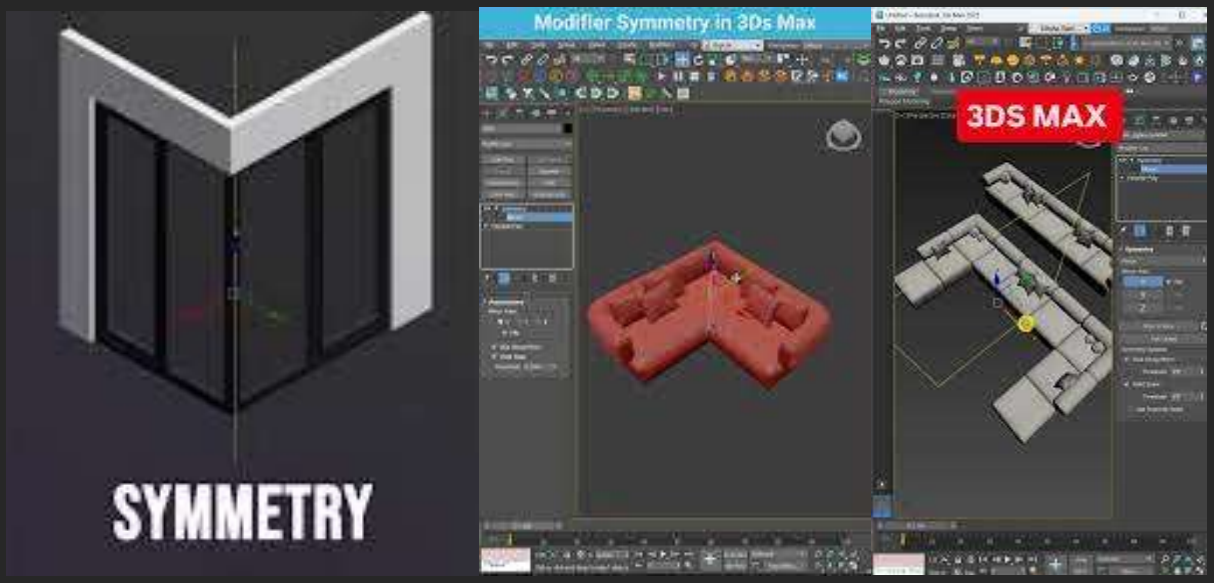
As you do so, its counterpart on the symmetry object moves symmetrically in real time.

As you can see, the Symmetry modifier not only creates a mirror image of an object for you, but also lets you manipulate both sides in tandem in an intuitive way.

9. Now  move one of the vertices on the left side of the box, where it overlaps the symmetry box.

Because you're also moving its counterpart vertex, which is invisible, the apparent result is motion of the corresponding point on the plane of symmetry. This isn't as intuitive as moving a non-overlapping point, so for best results, position the Mirror gizmo so as to cause as little overlap as possible; that way you can edit the center vertices directly on the plane of symmetry.



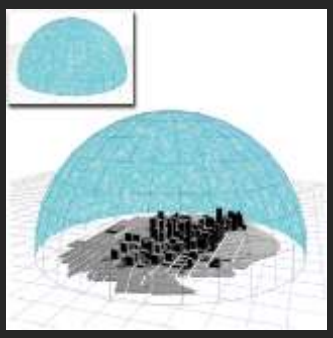




# Normal Modifier

The Normal modifier allows you to unify or flip the normals of an object without applying an Edit Mesh modifier.

- Select an object. > Modify panel > Modifier List > Normal
- Default menu: Select an object. > Modifiers menu > Mesh Editing > Normal Modifier
- Alt menu: Select an object. > Modifiers menu > Geometry (Convert to Mesh) > Normal Modifier

For example, if you wanted to fly inside of a procedural object, such as a sphere or a cylinder, and wanted to retain control over the radius and number of segments, you couldn't collapse the object to an Editable Mesh and maintain the procedural nature of the primitive.



Flipping the normals of a sphere creates a sky dome over a city.

**Tip:** If you are animating the creation of a complex object such as a nested Boolean or a loft, and you think the operation might result in inconsistent faces, apply the Normal modifier to the result and turn on Unify normals.

**Tip:** The Lathe modifier sometimes creates an object with normals pointing inward. Use the Normal modifier with both Unify and Flip turned on to fix "inside-out" lathe objects. The Normal modifier allows whole-object manipulations of normals to be performed quickly without using an Edit Mesh modifier.

**Patches**

Patch objects coming up the modifier stack are not converted to a mesh by this modifier. A patch object input to the Material modifier retains its patch definition.

**Procedures**

**To use the normal modifier:**

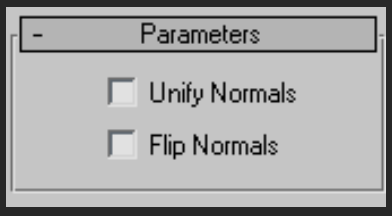
1. Select an object, then on the Modify panel ➤ Modifier List, choose Object-Space Modifiers ➤ Normal.

The object appears to turn inside-out, since Flip Normals is on by default.

2. If the object has some faces pointing inward and others outward, turn on Unify Normals to make all the faces point similarly.

**Tip:** To flip or unify normals on part of an object, convert it to editable mesh and select faces or polygons. On the Surface Properties rollout in the Normals group, use the Flip and Unify buttons.

**Interface**

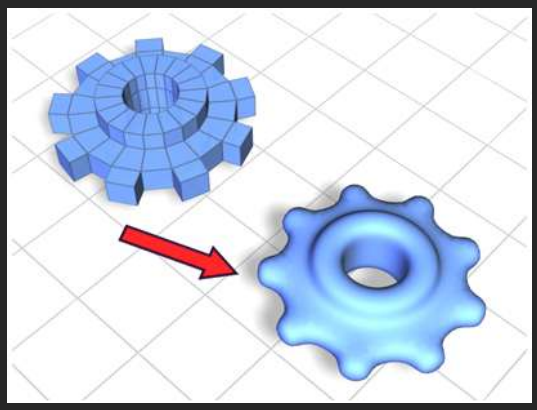



**Unify Normals**

Unifies the normals of an object by flipping the normals so that they all point in the same direction, usually outward. This is useful for restoring an object's faces to their original orientations. Sometimes

normals of scenes that have come into 3ds Max as part of a DWG or DXF file are irregular, depending on the methods used to create the scene. Use this option of the modifier to correct them.

**Note:** Unify Normals does not work on editable poly objects; before applying the Normal modifier, convert the model to editable mesh format or apply a Mesh Selector Turn To Mesh modifier.



**Flip Normals**

Reverses the direction of all the surface normals of the faces of the selected object or objects.

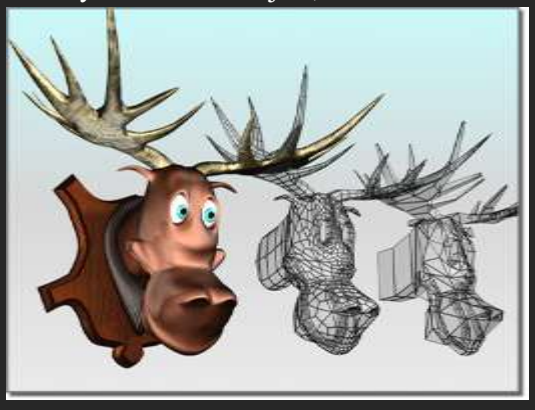
# TurboSmooth Modifier

The TurboSmooth modifier, like MeshSmooth, smoothes geometry in your scene.

- Make a selection. > Modify panel > Modifier List > Object-Space Modifiers > TurboSmooth
- Make a selection. > Modifiers menu > Subdivision Surfaces > TurboSmooth

The differences between the two are as follows:

- TurboSmooth is considerably faster and more memory-efficient than MeshSmooth. TurboSmooth also has an option for Explicit Normals, unavailable in MeshSmooth. See Explicit Normals.
- TurboSmooth provides a limited subset of MeshSmooth functionality. In particular, TurboSmooth uses a single smoothing method (NURMS), can be applied only to an entire object, has no sub-object levels, and outputs a triangle-mesh object.



TurboSmooth lets you subdivide the geometry while interpolating the angles of new faces at corners and edges, and apply a single smoothing group to all faces in the object. The effect of TurboSmooth is to round over corners and edges as if they had been filed or planed smooth. Use TurboSmooth parameters to control the size and number of new faces, and how they affect the surface of the object.

Angular model (shown on the right) changed to a smooth model with TurboSmooth

Smoothing an object modeled with extrusions

# LECTURE 8
# AL-HELLI MARWAH

You use TurboSmooth to produce a Non-Uniform Rational MeshSmooth object (NURMS for short). A NURMS object is similar to a NURBS object in that you can set different weights for each control vertex.

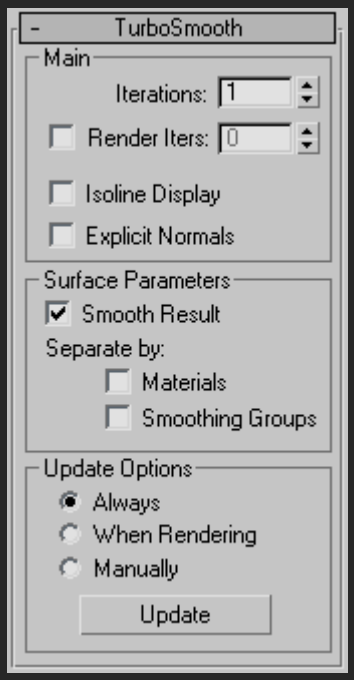
TurboSmooth's effect is most dramatic on sharp corners and least visible on rounded surfaces. Use TurboSmooth on boxes and geometry with crisp angles. Avoid using it on spheres and similar objects.

**Tip:** To better understand TurboSmooth, create a sphere and a cube and apply TurboSmooth to both. The cube's sharp corners become rounded, while the sphere's geometry becomes more complex without changing shape significantly.

**Procedures**

**To apply TurboSmooth to an object:**

1.  Select an angular object.
2.  Apply the TurboSmooth modifier.
3. Set TurboSmooth parameters.




**Example: To compare the speeds of TurboSmooth and MeshSmooth:**

1. Create a Box primitive with Length/Width/Height Segs=**3**. Convert the box to editable poly format.
2.  Apply MeshSmooth.
3. Set Iterations=**5**.

This creates a heavily subdivided mesh.

4. Go to the Editable Poly ➤ Vertex sub-object level, and turn on  (Show End Result On/Off Toggle).
5.  Move one of the corner vertices outward.

There is a significant delay before you see the result of the Move operation.

6. Perform a few more  Move operations on vertices, observe the delays, and then undo (Ctrl+Z) repeatedly until the MeshSmooth modifier goes away.
7. Apply TurboSmooth.
8. Set Iterations=**5**.

This creates a heavily subdivided mesh.

9. Go to the Editable Poly ➤ Vertex sub-object level, and turn on  (Show End Result On/Off Toggle).

10.  Move one of the corner vertices outward.

The response is much faster.

**Interface**

**Main group**

Lets you set the basic parameters for TurboSmooth.

**Iterations**

Sets the number of times the mesh is subdivided. When you increase this value, each new iteration subdivides the mesh by creating smoothly interpolated vertices for every vertex, edge, and face from the iteration before. The modifier then subdivides the faces to use these new vertices. Default=1. Range=0 to 10.

From right to left, effect of increasing the number of iterations

**Note:** Be cautious when increasing the number of iterations. The number of vertices and faces in an object (and thus the calculation time) can increase as much as four times for each iteration. Applying four iterations to even a moderately complex object can take a long time to calculate.

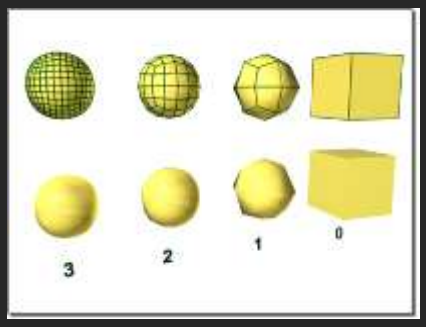
**Render Iter[ation]s**

Lets you choose a different number of smoothing iterations to be applied to the object at render time. Turn on Render Iters, and then use the field to its right to set the number of render iterations.

**Isoline Display**

When on, 3ds Max displays only isolines: the object's original edges, before smoothing. The benefit of using this option is a less cluttered display. When off, 3ds Max displays all faces added by TurboSmooth; thus, higher Iterations values result in a greater number of lines. Default=off.

**Warning:** If you're going to collapse the model or apply further modifiers after the TurboSmooth, you should first turn off Isoline Display. Unlike in MeshSmooth, isoline display is achieved by making all the edges "invisible," joining large groups of faces together in single "polygons." This can be especially problematic if you apply a PolyObject-based modifier afterwards, because all vertices in the interior of these "polygons" will be lost.




**Explicit Normals**

Lets the TurboSmooth modifier compute normals for its output, which is faster than the standard method 3ds Max uses to compute normals from the mesh object's smoothing groups. Default=off.

Consequently, if the TurboSmooth result is used directly for display or rendering, it will generally be faster with this option turned on. Also, the quality of the normals will be slightly higher. However, if you

apply any topology-affecting modifiers, such as Edit Mesh, above the TurboSmooth modifier, these normals will be lost and new ones computed, potentially affecting performance adversely. So it's important to remember to turn on Explicit Normals only if no modifiers change the object topology after TurboSmooth takes effect.

**Surface Parameters group**

Lets you apply smoothing groups to the object and restrict the smoothing effect by surface properties.

**Smooth Result**

Applies the same smoothing group to all faces.

**Separate by Materials**

Prevents the creation of new faces for edges between faces that do not share Material IDs.

**Separate by Smoothing Groups**

Prevents the creation of new faces at edges between faces that don't share at least one smoothing group.

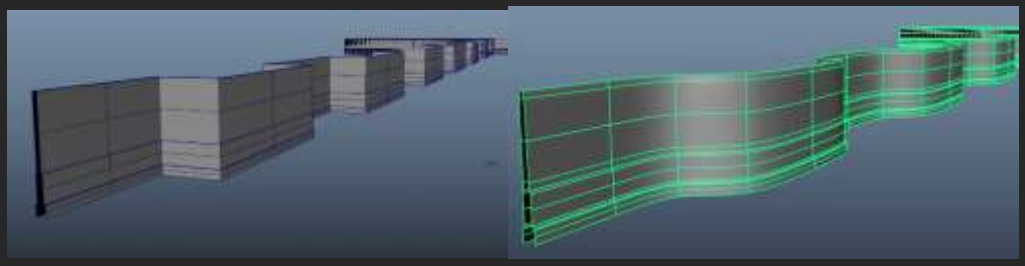
**Update Options group**

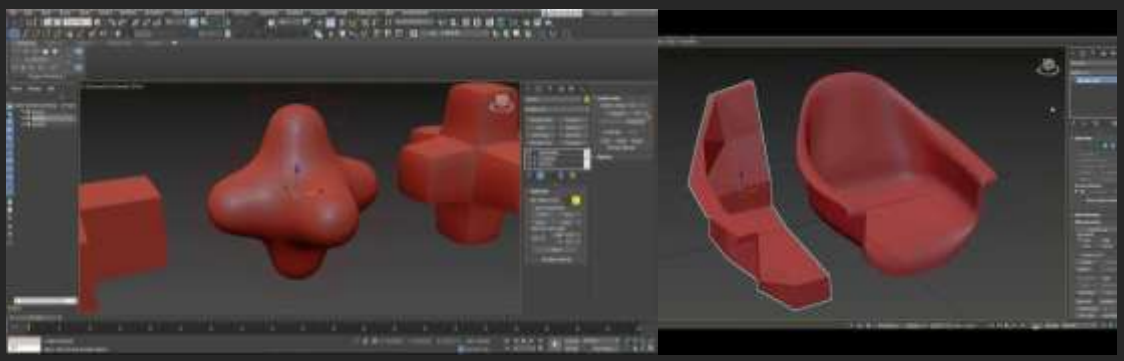
Sets manual or render-time update options, for situations where the complexity of the smoothed object is too high for automatic updates. Note that you can also set a greater degree of smoothing to be applied only at render time, in the Main group.

- **Always** Updates the object automatically whenever you change any TurboSmooth settings.
- **When Rendering** Updates the viewport display of the object only at render time.
- **Manually** Updates the object only when you click Update.

**Update**

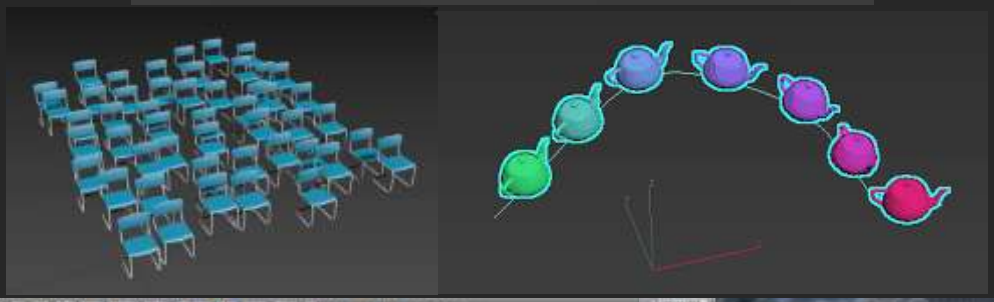
Updates the object in the viewport to match the current TurboSmooth settings. Works only when you choose When Rendering or Manually.

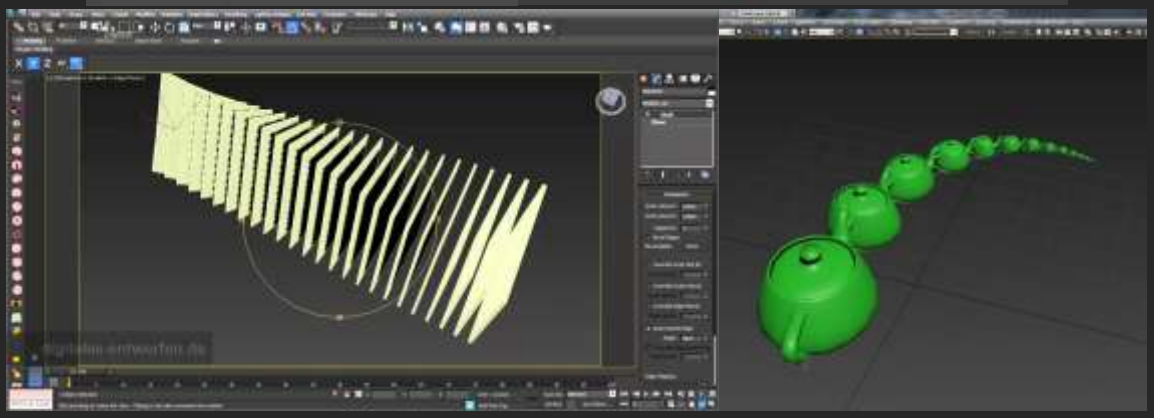




# Array Modifier

-  Modify panel > Make a selection > Modifier List > Object-Space Modifiers > Array
  - Default menu: Make a selection > Modifiers menu > Parametric Deformers > Array





- [Array Modifier Transform Rollout]

-  Modify panel > Make a selection > Modifier List > Object-Space Modifiers > Array > Transform rollout
- Default menu: Make a selection > Modifiers menu > Parametric Deformers > Array > Transform rollout

  The Array modifier's Transform rollout provides controls to adjust the position, rotation, and scale of the clones in the array.

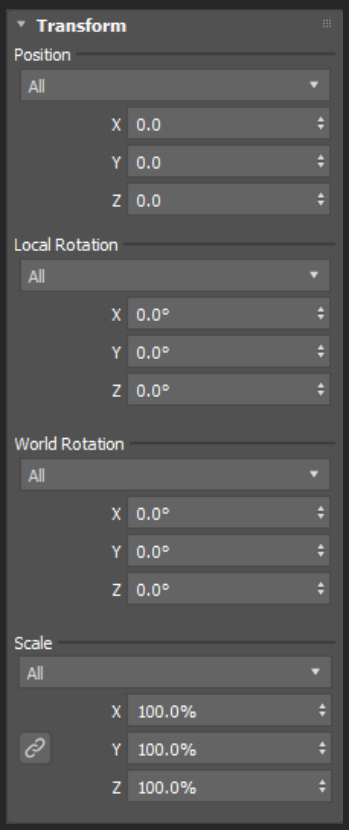
  **Interface**

## Position

The following transform options are only available when the Distribution method is set to Grid. See Array Modifier Distribution Rollout (Grid).

### Position Method

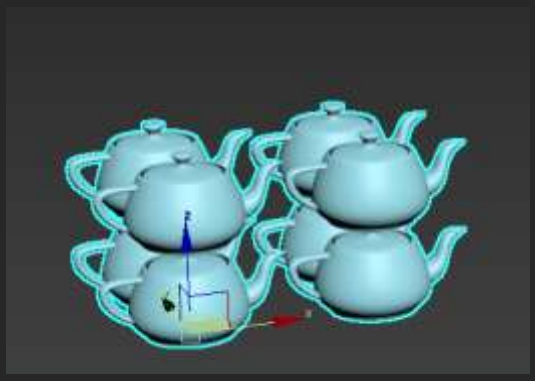
Sets the position transform method.




### All

Moves each clone the distance specified in the X, Y, and/or Z parameters. All clones move outward from the first clone.

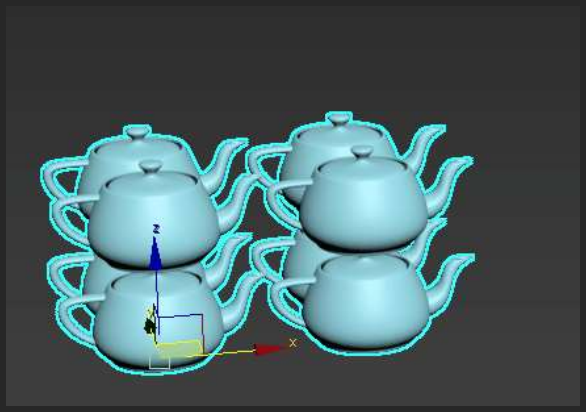
In the following example, the X, Y, and Z parameters are set to 10.



### Incremental

Moves each clone successively the distance specified in the X, Y, and/or Z parameters.

In the following example, the X, Y, and Z parameters are set to 3.



### By Axis

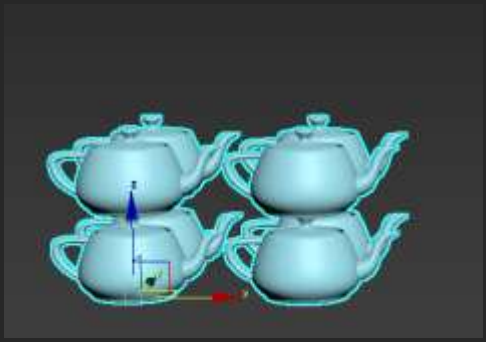
Moves each row by the distance specified in the X, Y, and/or Z parameters.

In the following example, the X, Y, and Z parameters are set to 10.

**Incremental By Axis**

Compounds the offset successively per row in the axis specified.

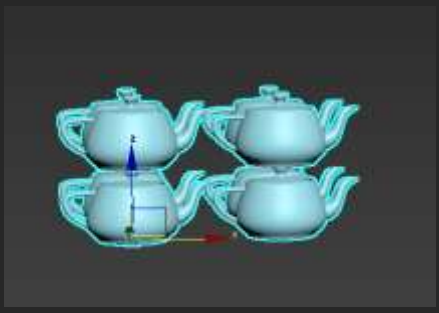
In the following example, the X, Y, and Z parameters are set to 10.



**Alternating**

Moves every other row by the distance specified in the X, Y, and/or Z parameters.

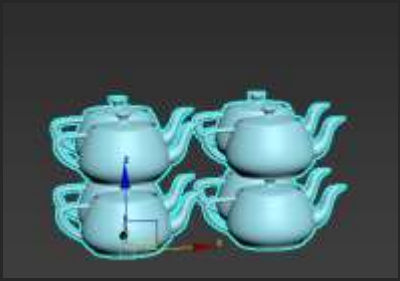
In the following example, the X parameter is set to 10.



**Z By Y**

Moves each successive row in Y by a given amount in Z. This is useful for creating flights of stairs or stadium seating.

In the following example, the Z parameter is set to 10.



**X, Y, Z**

Uses the specified value to modify the array's local position along the X, Y, or Z axis. Values are measured in display units.

**Local Rotation**

**Local Rotation Method**

Sets the local rotation transform method.

**All (Default)**

Rotates each clone about its local axis by the amount specified by the X, Y, Z parameters.

**Incremental**

Rotates each clone about its local axis by an additive incremental amount specified by the X, Y, Z parameters.

**By Axis**

This option is only available when the Distribution Method is set to Grid.

Rotates all the clones in each row by an additive incremental amount specified by the X, Y, Z parameters.

**Incremental By Axis**

This option is only available when the Distribution Method is set to Grid.

Rotates all the clones in each row by an incremental amount specified by the X, Y, Z parameters.

**Alternating**

Rotates every other clone by an amount specified by the X, Y, Z parameters.

**Random Increment**

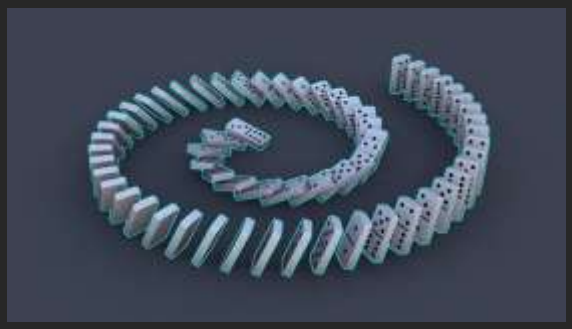
Rotates all the clones multiples of the amount specified by the X, Y, Z parameters.

**Tip:** You can use this parameter to create non periodic tiling or minimize repetition in tiled elements while using the Grid Distribution method.

**Progressive**

This option is only available when the Distribution Method is set to Radial, Spline, or Phyllotaxis.

Rotates clones progressively. For example, a Local Rotation X value of 360 degrees results in no rotation for the first clone, but the last clone is rotated 360 degrees. Every clone in between is rotated incrementally between 0-360 degrees. In the following example, Local Rotation Y is set to -72.0, creating a set of toppling dominos.



**X, Y, Z**

Uses the specified value(s) to modify the array's local rotation about the X, Y, and/or Z axis in degrees. Values are measured in display units.

**World Rotation**

The following transform options are only available when the Distribution method is set to Grid.

**World Rotation Method**

Sets the world rotation transform method.

**All**

Rotates all the clones based on the original object's pivot by an amount specified by the X, Y, Z parameters.

**Incremental**

Rotates all the clones an incremental amount based on the original object's pivot.

**By Axis**

Rotates all clones in a row along the specified axis.

**Incremental By Axis**

Rotates all clones in a row incrementally along the specified axis.

**Bend**

Rotates all clones in a row incrementally along the specified axis, creating a bend.

**X, Y, Z**

Uses the specified value(s) to modify the array's world rotation about the X, Y, and/or Z axis in degrees. Values are measured in display units.

**Scale**

**Scale Method**

Sets the world scale transform method.

**All (Default)**

All clones are scaled in the specified axis.

**Incremental**

All clones are scaled incrementally in the specified axis.

**By Rings**

This option is only available when the Distribution Method is set to Radial.

All clones within a ring are scaled incrementally in the specified axis.

**By Rows**

This option is only available when the Distribution Method is set to Radial.

All clones within a row are scaled incrementally in the specified axis.

**By Axis**

This option is only available when the Distribution Method is set to Grid.

All clones in a row are scaled incrementally in the specified axis.

**Alternating**

This option is only available when the Distribution Method is set to Grid.

Scales every other clone by the specified amount.

**Progressive**

This option is only available when the Distribution Method is set to Radial, Spline, or Phyllotaxis.

Scales clones progressively. For example, a Scale X value of 200% results in no increase in scale for the first clone, but the last clone is scaled 200%. Every clone in between is scaled incrementally between 0-200%.

**X, Y, Z**

Uses the specified percentage(s) to modify the array's scale along the X, Y, and/or Z axis. Values are measured in display units.

**Scale Link ( )**

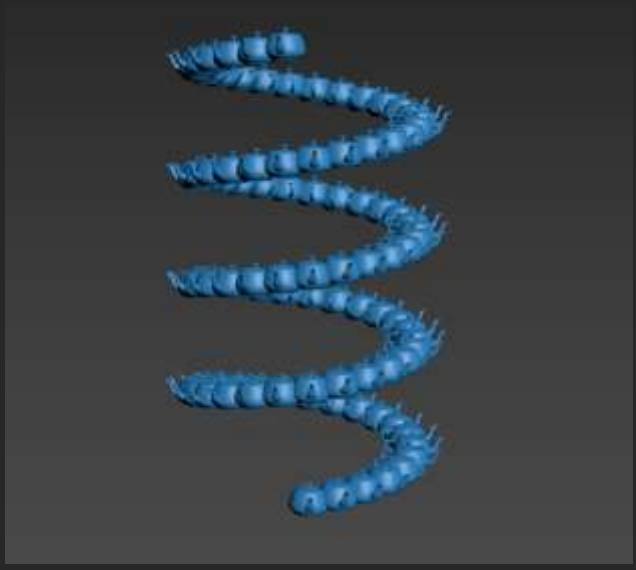
When enabled, all three axes are controlled by the X axis parameter, making it easier to uniformly manipulate the clones.

**Radial**

The following transform options are only available when the Distribution Method is set to Radial.

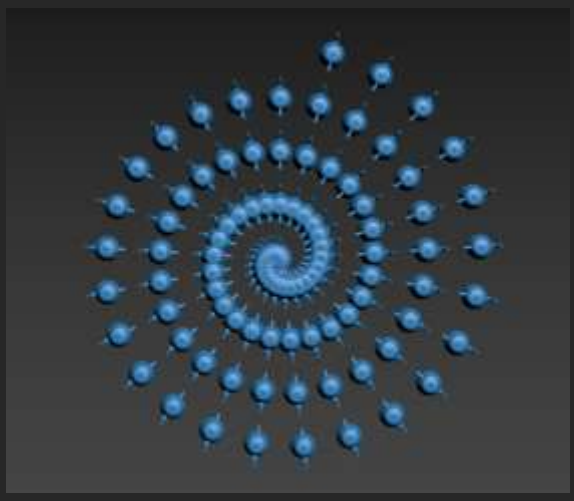
**Axial**

Moves each clone successively along the main axis of the array. This can be used to produce a helical pattern. In the following example, Axial is set to 40.



**Radial**

Moves each clone successively and tangentially (towards/away from) along the main axis of the array. This can be used to produce a spiral pattern. In the following example, Radial is set to 0.3.



**Ring**

Moves each ring of clones successively along the main axis of the array.

**Radial Phase**

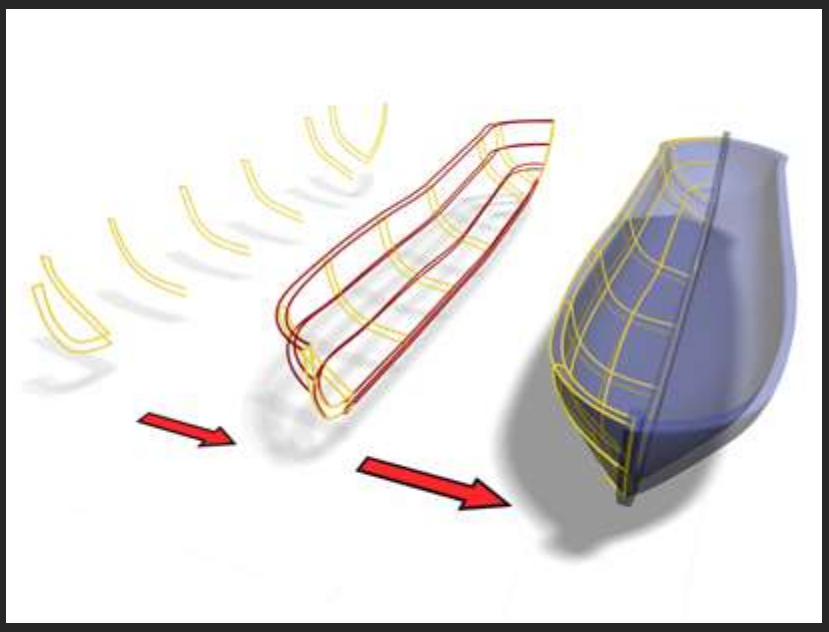
Rotates each ring along the main axis via a phase, where a value of 1.0 equals 360 degrees of rotation.

**Ring Phase**

Rotates each ring in an alternating pattern along the main axis via a phase, where a value of 1.0 equals 360 degrees of rotation.

**CrossSection Modifier**

The CrossSection modifier creates a "skin" across multiple splines. It works by connecting the vertices of 3D splines to form a skin. The resulting object is another spline object that can be used with the Surface modifier to create a patch surface. These two modifiers, when used together, are sometimes referred to collectively as “Surface Tools.”



CrossSection uses splines to create a model of a boat.

CrossSection can build a skin across various-shaped splines with different vertex counts and open/closed status. The more different the splines in vertex count and complexity, the more likely the skin will have discontinuity.

**Note:** Similar functionality is provided by the Editable Spline object. At the Editable Spline ➤ Segment and Spline sub-object levels, you can create a spline cage using Connect Copy and Cross Section. Using this method, you need to region-select the created vertices to transform them. Also, this method lets you define the ordering of the spline more easily than does the CrossSection modifier.

**Procedures**

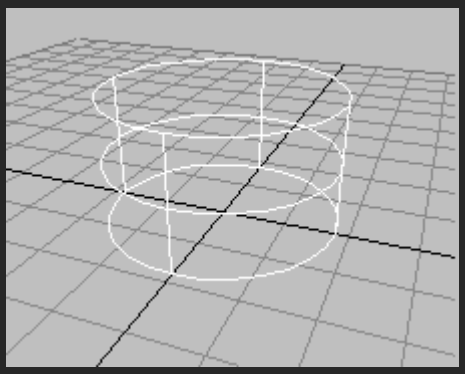
**Example: To explore the CrossSection modifier:**

1. On the  Create panel, click  (Shapes), then click Circle.
2. Drag in the Top viewport to create a circle about **100** units in radius.
3. On the  Modify panel, choose Edit Spline from the Modifier List.
4. In the modifier stack display, turn on Spline sub-object, then  select the circle.
5. In the Front viewport, Shift+Move the spline up to copy it.
6. Shift+Move the copy up to create a third circle.

   **Note:** The order that you attach or clone splines is important: this is the order that CrossSection uses to create the skin.

7. On the  Modify panel, choose CrossSection from the Modifier List.

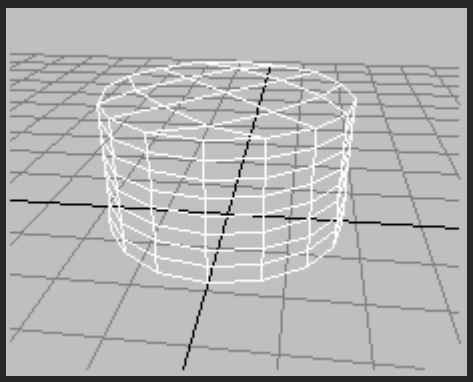
   CrossSection joins the vertices of the three circles. A basic spline cylinder is displayed.



8. On the  Modify panel, on the Modifiers List, choose Surface to add the Surface modifier.

   The spline cylinder is transformed into a patch surface by the Surface modifier.

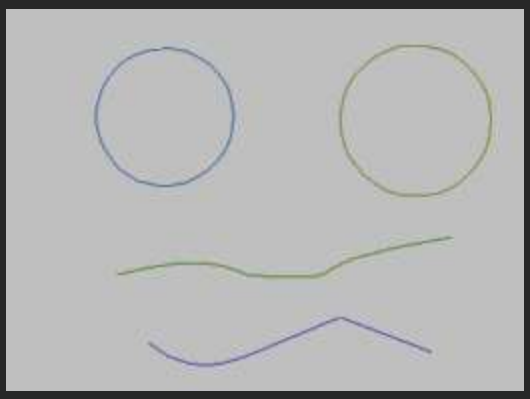
9. To edit the model's surface, change the splines using controls in the Edit Spline modifier. Or, since the output of the Surface modifier is a patch surface, add an Edit Patch modifier and use patch edit controls to change the surface.



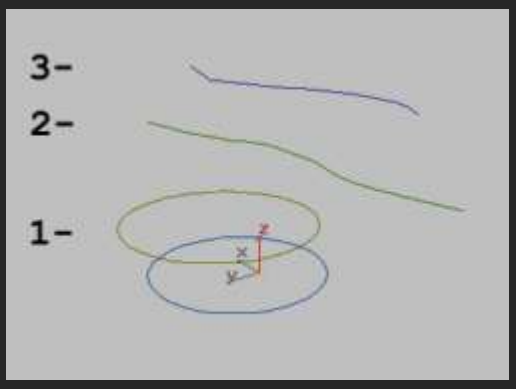
An Edit Patch modifier above the Surface modifier was used to create the image.

**Example: Use the CrossSection modifier to skin several splines with different shapes:**

1. On the  Create panel, click  (Shapes).
2. On the Object Type rollout, turn on Start New Shape, then click NGon.
3. In the Top viewport, create two five-sided circular NGons.
4. On the  Create panel, with  (Shapes) still active, click Line. Create two lines, each with four vertices. Create the vertices left-to-right.



5. On the main toolbar, click  (Select And Move), then move the objects in the viewport to order them along the Z axis, with the NGons at the bottom and the lines above the NGons.




6. Select the bottom NGon.
7. On the Modify panel, choose Edit Spline from the Modifier List.
8. In the Geometry rollout, click Attach.
9. Select the remaining NGon and lines in an ascending order, as numbered in the image.

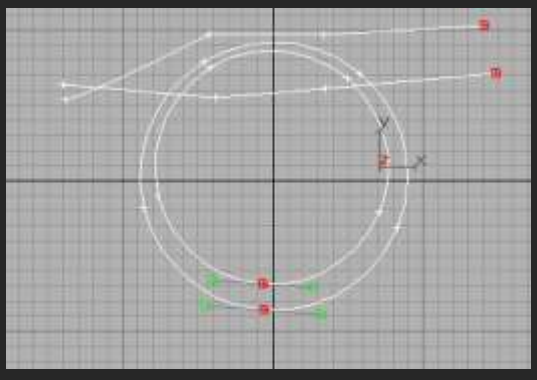
   **Note:** The order of selection is important. The CrossSection modifier uses the selection order to define the skin.

   **Example continued: Lining up the vertices:**

1. On the Modify panel, choose the Vertex sub-object level in the stack display.

   Lining up the first vertex of each spline is important to prevent the surface from twisting.

2. Use Ctrl+click to select the rightmost vertex of each line and the bottommost vertex of each NGon.
3. On the Geometry rollout, click Make First.



   Aligning the first vertex is important. This is where the seam forks, going from a closed to an open spline.

   **Example continued: Use CrossSection and Surface to "skin" the shapes:**

1. On the Modify panel, choose CrossSection from the Modifier List.

The CrossSection modifier connects the splines at the vertices.

2. On the Modifiers List, choose Surface.

   The Surface modifier generates a patch surface based on the splines.

3. In the modifier stack display, choose the CrossSection modifier.

4. On the CrossSection Parameters rollout, toggle between Linear and Smooth. Notice how the splines change.

5. On the Modify panel, toggle  (Show End Result On/Off Toggle) to display the final patch surface. The toggle won't remain on if the CrossSection modifier is current. Drop down to the Editable Patch in the stack and turn on the Show End Result toggle if you like.

   **Tip:** When you use CrossSection, be sure to draw splines in a consistent direction. When you create lines from vertices that are not properly aligned, a twisted surface can result.

**Topics in this section**

- CrossSection Modifier Reference
  Usage of the CrossSection modifier is by and large determined by the arrangement of the splines to which it is applied. For details, see the parent of this topic.