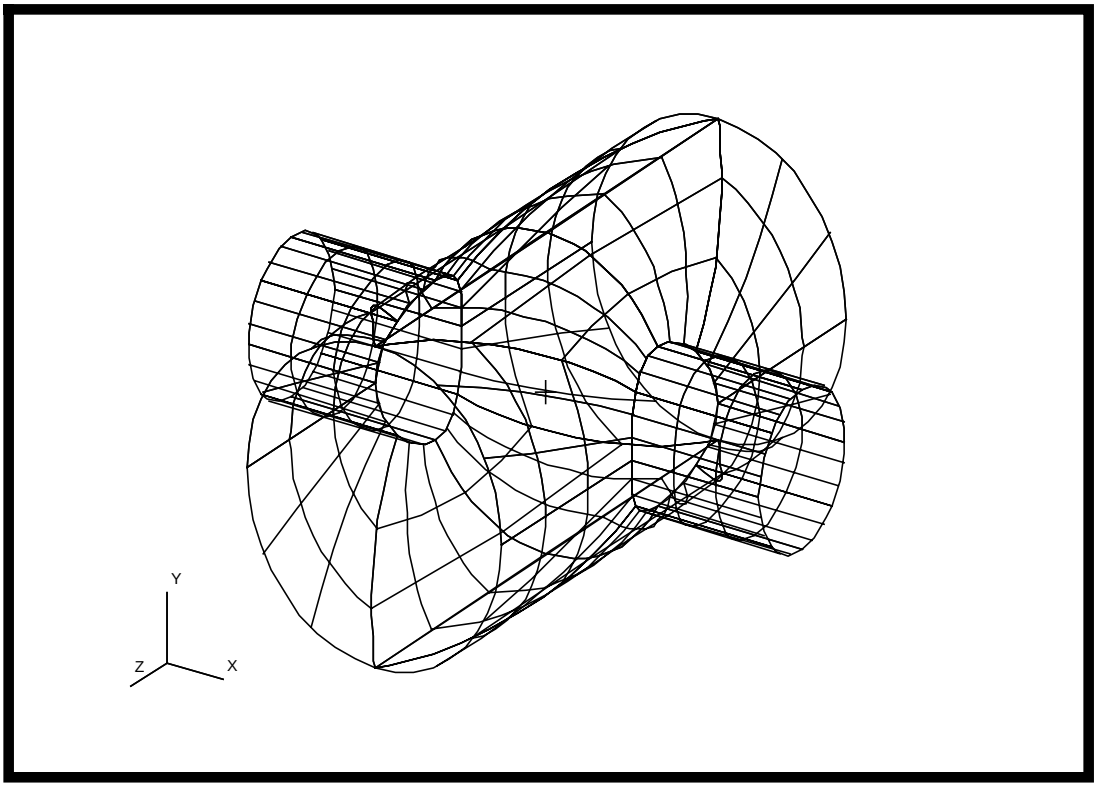
LESSON 18

Parameterized Pipe Pipe Intersection



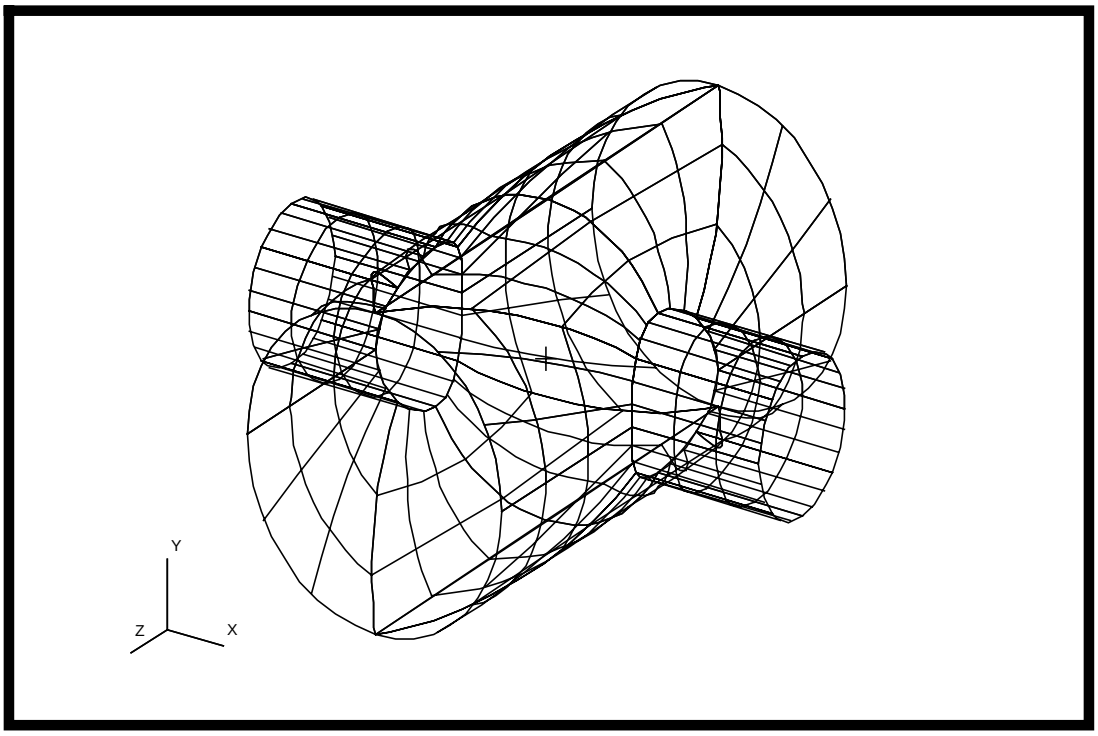
Objectives:

- Model a pipe pipe intersection.
- Create a PCL Function from the resulting session file.

Model Description:

In this exercise you will model a cylinder (pipe) of one radius intersecting a cylinder of a different radius. The resulting session file will be used to create a PCL function which will allow you to vary the two radii to solve a set of modeling problems.

Shown below is the resulting pipe intersection model.



Suggested Exercise Steps:

- Start PATRAN.
- Open a database (pipe.db).
- Define two real variables, rad_1 and rad_2.
- Model a pipe pipe intersection using those two variables.
- Exit PATRAN and edit the session file to make a PCL function.
- Change the values of the variables rad_1 and rad_2 and execute the function to verify that it works.

Files:

All the files used in this exercise are listed below. Each listing includes the file, where it originated, its format (text/binary) and summary information as to how it relates to this exercise.

File	Supplied/Create	Description
pipe.db	Created	This is a P3/PATRAN database (binary) that is created in this exercise. The geometry for the model is created in this exercise using parametric variables rad_1 and rad_2.
patran.ses	Created	This is a session file (text) that is created when the pipe.db is made. This file contains the basic instructions for making the pipe-pipe intersection. This file will be copied to pipe.pcl and modified.
pipe.pcl	Created	This file is a modified version of the patran.ses file (above). Additional logic is added to check for one pipe being larger than the other.

Exercise Procedure:

Model the Pipe Pipe Intersection

1. In your xterm window type **p3** to start P3/PATRAN. After that, create a new database named **pipe.db.**

File/New ...

New Database Name: **pipe**

OK

In the *New Model Preference* form set the following:

Tolerance: ◆ **Default**

Analysis Code: **MSC/PatranFEA**

Analysis Type: **Structural**

OK

Enter the following in the command window:

REAL rad_1 = 1., rad_2 = .5

Click on **Geometry** in the *Main Window*, and then perform the following:

◆ **Geometry**

Action:	**Create**
Object:	**Point**
Method:	**XYZ**
Point Coordinates List	**[‘rad_1‘ 0 0]**

Apply

A point should show up on the screen.

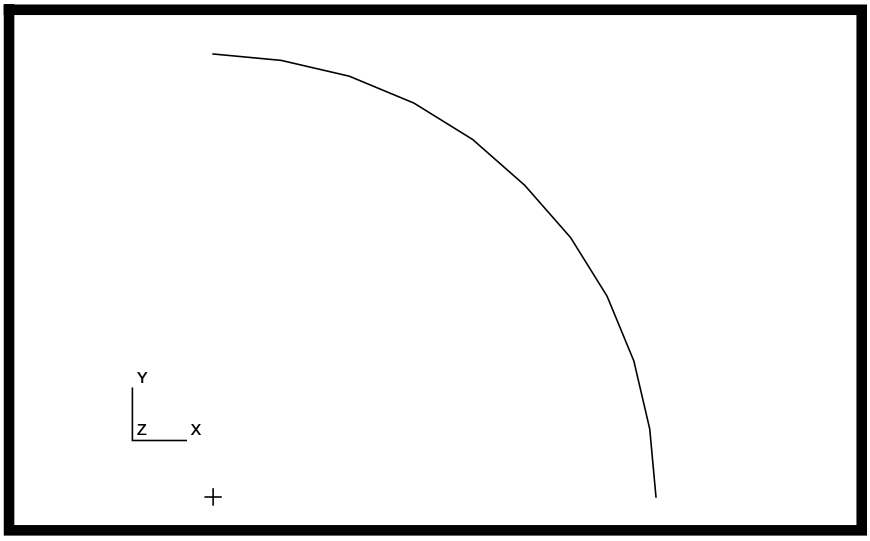
2. Secondly, create *Curve 1* as follows:

◆ **Geometry**

Action:	**Create**
Object:	**Curve**
Method:	**Revolve**
Axis	**{[0 0 0][0 0 1]}**
Total Angle	**90.0**
Point List	**Point 1**

Apply

Curve 1 should appear as follows:

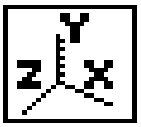


3. Thirdly, create a surface as follows:

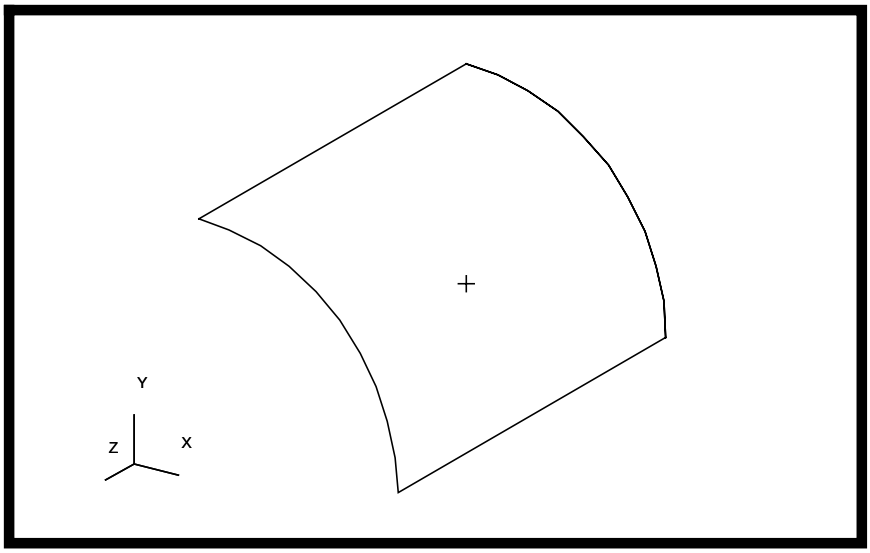
◆ **Geometry**

Action:	**Create**
Object:	**Surface**
Method:	**Extrude**
Translation Vector	**<0 0 ‘2. * rad_1‘>**
Curve List	**Curve 1**

Use the **Iso 1 View** icon to change the model to an isometric display.



Surface 1 should appear as follows:



4. Now perform the following to create *Point 5*:

◆ **Geometry**

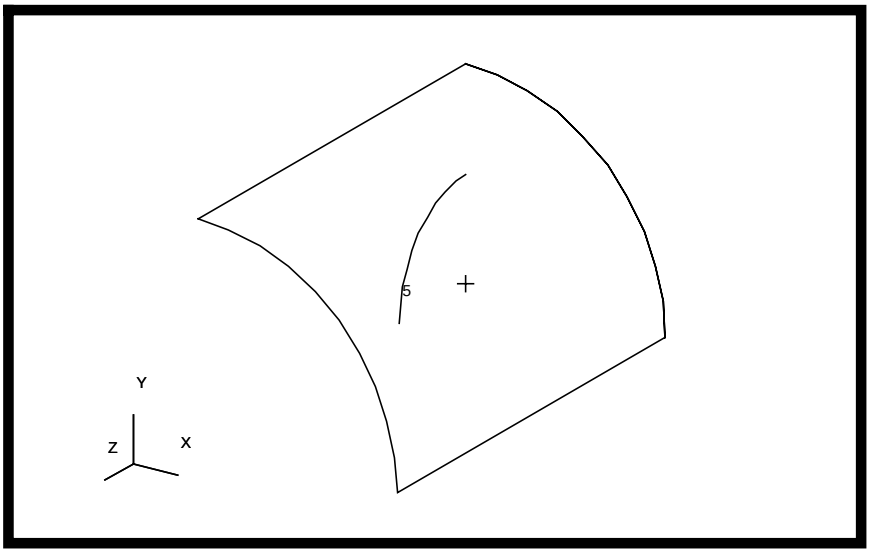
Action:	**Create**
Object:	**Point**
Method:	**XYZ**
Point Coordinates List	**[0 0 'rad_2']**

5. Fifthly, create *Curve 2* as follows:

◆ **Geometry**

Action:	**Create**
Object:	**Curve**
Method:	**Revolve**
Axis	**{[0 0 0][1 0 0]}**
Total Angle	**-90.0**
Point List	**Point 5**

Your model should appear as shown below.

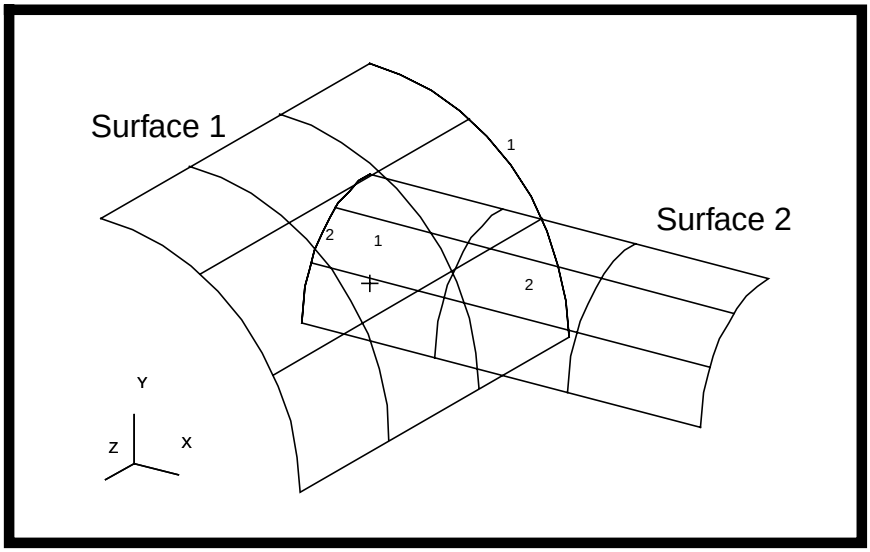


6. Next, create another surface as follows:

◆ **Geometry**

Action:	**Create**
Object:	**Surface**
Method:	**Extrude**
Translation Vector	**<‘2. * rad_1‘ 0 0>**
Curve List	**Curve 2**

Two surfaces should appear in your viewport as follows:

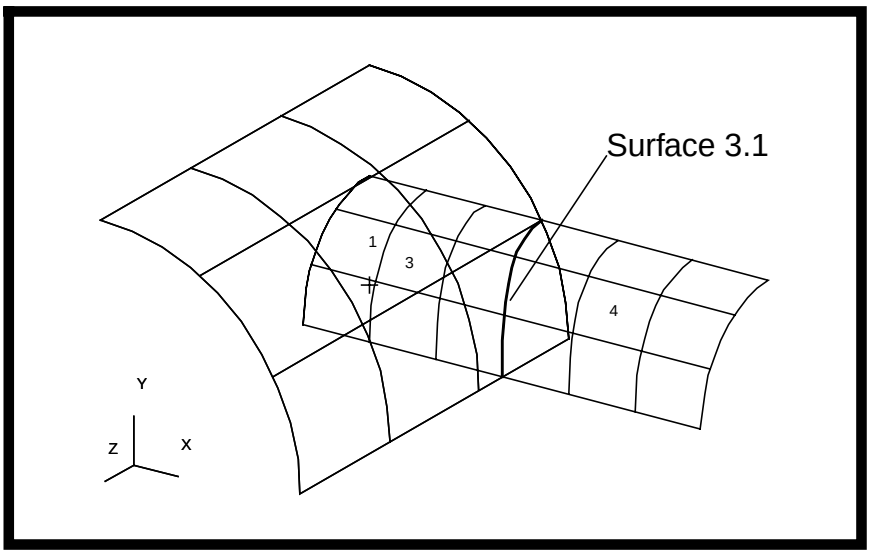


7. Break up *Surface 2* into two pieces at the intersection of Surface 1 and Surface 2:

◆ **Geometry**

Action:	**Edit**
Object:	**Surface**
Method:	**Break**
Option:	**Surface**
Surface List	**Surface 2**
Break Surface List	**Surface 1**

Answer **Yes** to the prompt to delete the original surface. Your model should look like the following:



Break the Edge

8. Now create *Curve 3* and *4* by breaking *Surface 3.1* as follows:

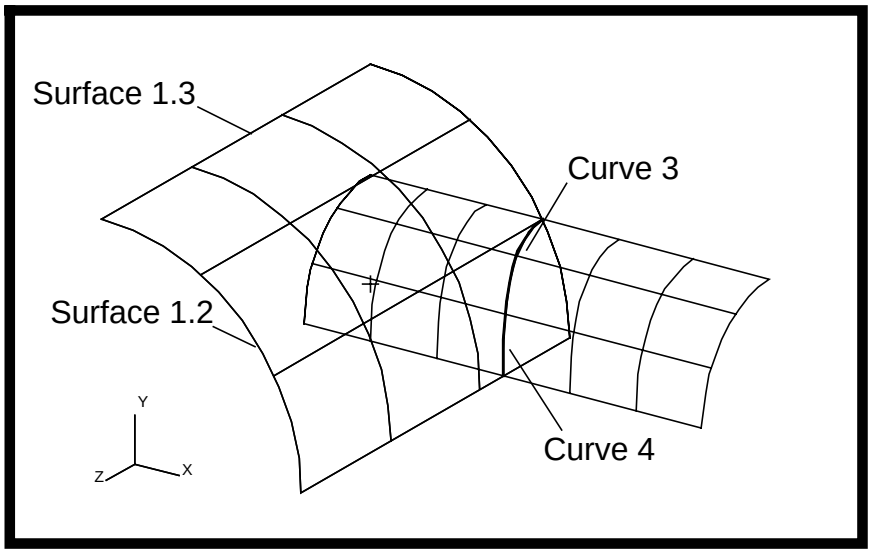
◆ **Geometry**

Action:	**Edit**
Object:	**Curve**
Method:	**Break**
Option:	**Parametric**
u Parametric Value	**0.5**

■ **Delete Original Curves**

Curve List	**Surface 3.1**

Your model should appear as follows:



9. Next, create more surfaces as follows (use the above picture for reference).:

◆ **Geometry**

Action:	**Create**
Object:	**Surface**
Method:	**Curve**
Option:	**2 Curve**

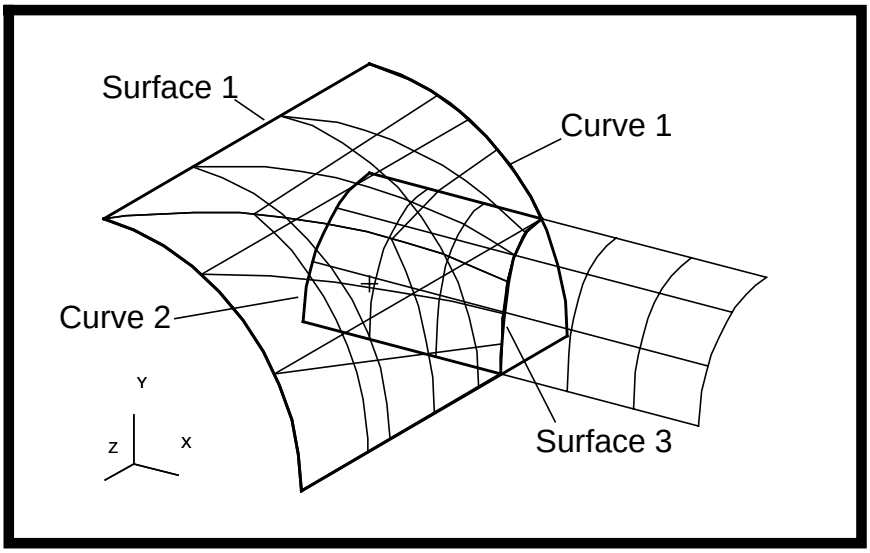
■ **Manifold**

❑ **Autoexecute**

Manifold Surface	**Surface 1**
Start Curve List	**Curve 3 4**
Ending Curve List	**Surface 1.3 1.2**

Apply

The following should appear in your viewport.



10. Delete some of the unnecessary entities as follows:

◆ **Geometry**

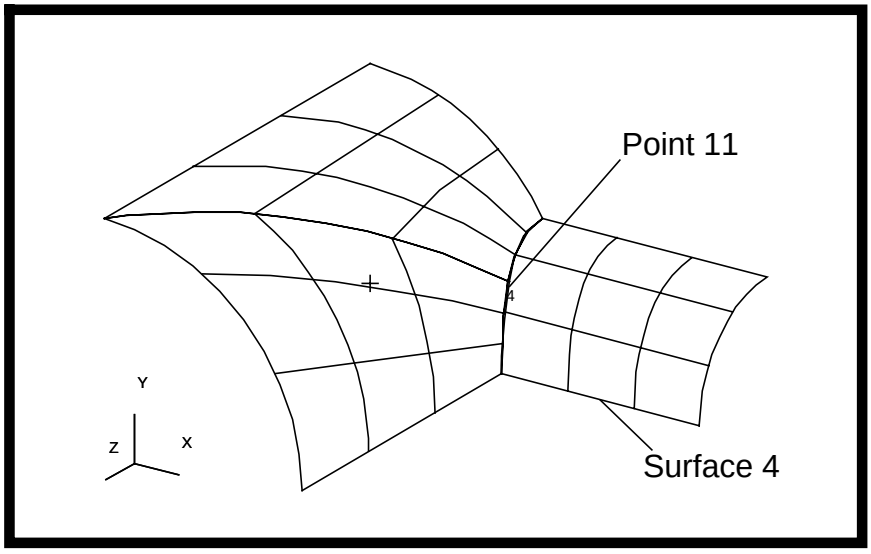
Action: **Delete**

Object: **Any**

Geometric Entity List **Surface 1 3 Curve 1 2**

Apply

Your model should appear as follows:



11. Break up *Surface 4* at Point 11 as follows:

Break the Surface

◆ **Geometry**

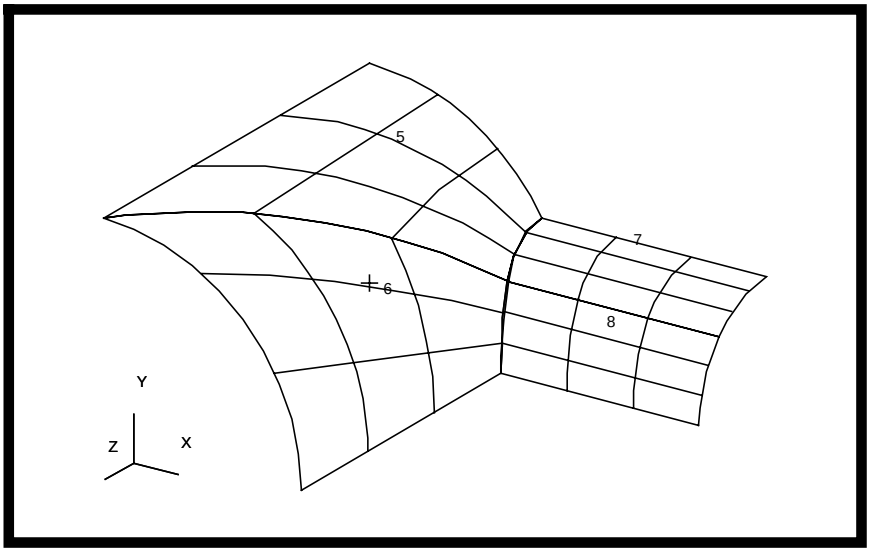
Action:	**Edit**
Object:	**Surface**
Method:	**Break**
Option:	**Point**

■ **Delete Original Surfaces**

Surface List	**Surface 4**
Break Point List	**Point 11**

Apply

Respond **Yes** to the prompt to delete *Surface 4*. The following should appear in your viewport.



12. Lastly, transform your model as follows:

Mirror the surfaces

First, mirror the surfaces in the z-direction as follows:

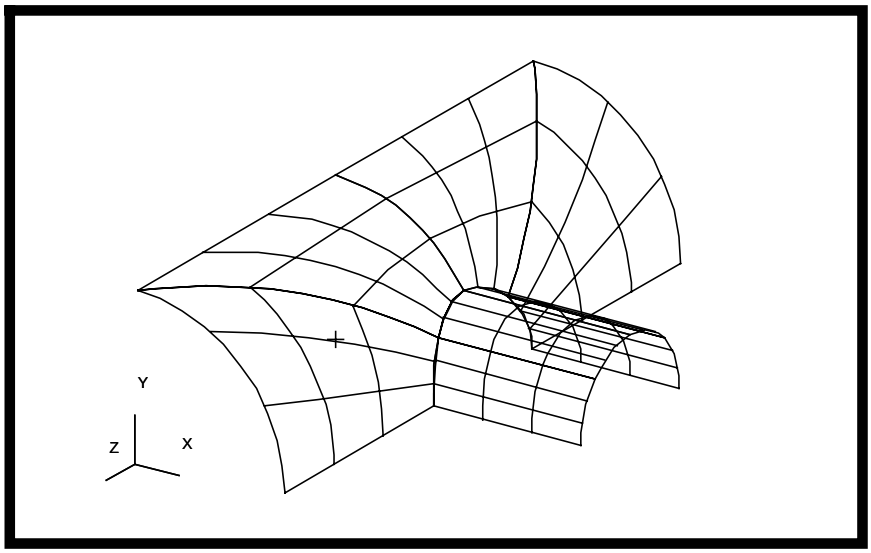
◆ **Geometry**

Action:	**Transform**
Object:	**Surface**

Method:	**Mirror**
Define Mirror Plane Normal	**{[0 0 0][0 0 1]}**
Offset Parameters	**0.0**
Surface List	**Surface 5:8 (Select all surfaces on screen)**

Apply

Your model should appear as follows:



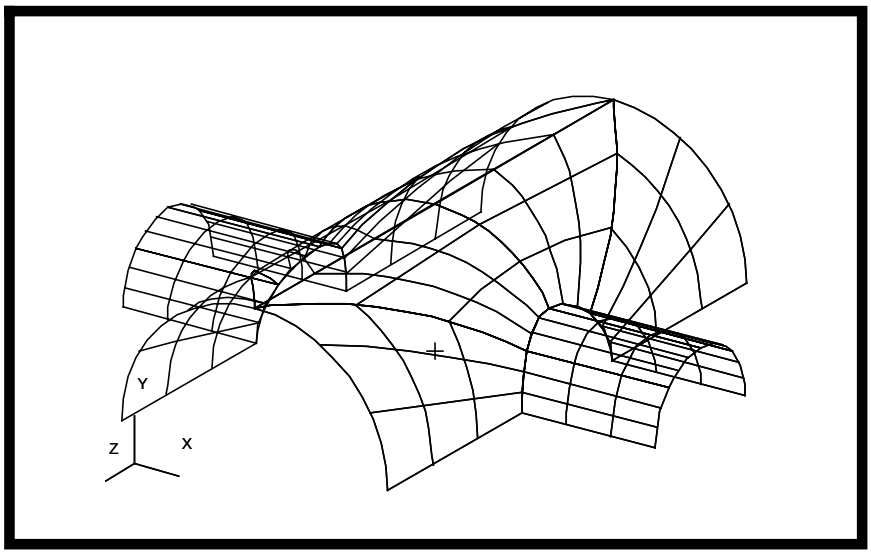
Second, mirror the model in the x-direction as follows:

◆ **Geometry**

Action:	**Transform**
Object:	**Surface**
Method:	**Mirror**
Define Mirror Plane Normal	**{[0 0 0][1 0 0]}**
Offset Parameters	**0.0**
Surface List	**Surface 5:12 (Select all surfaces on screen)**

Apply

The following should appear in your viewport



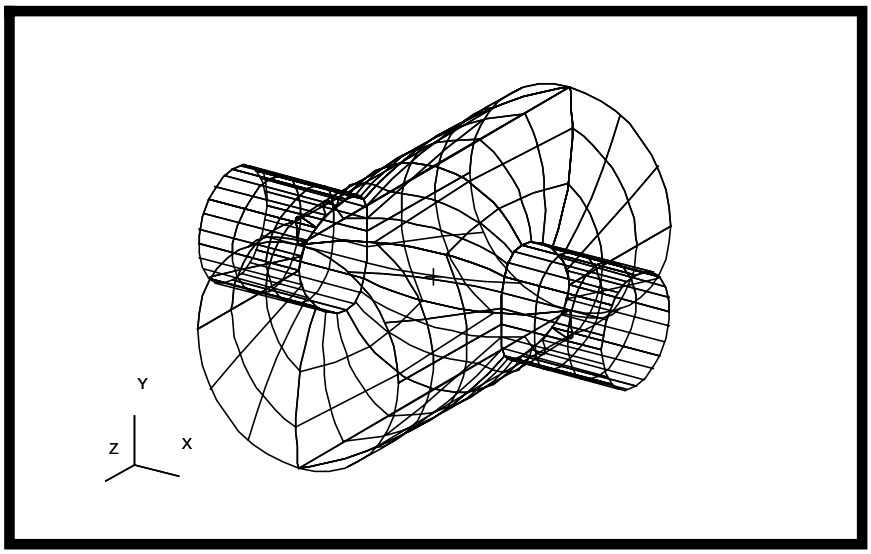
Finally, mirror the model in y-direction as follows:

◆ **Geometry**

Action:	**Transform**
Object:	**Surface**
Method:	**Mirror**
Define Mirror Plane Normal	**{[0 0 0][0 1 0]}**
Offset Parameters	**0.0**
Surface List	**Surface 5:20 (Select all surfaces on screen)**

Apply

Your model should appear as shown below.



Create a PCL Function from the session file

13. Exit PATRAN. Copy the latest version of the patran session file (patran.ses.*) to **pipe.pcl** as follows:

```
cp  patran.ses.*  pipe.pcl
```

Edit the file by deleting all the lines before the lines shown below (but not including):

```
STRING asm_create_grid_xyz_created_ids[VIRTUAL]
asm_const_grid_xyz( "1", "[`rad_1` 0 0]", "Coord 0",  @
asm_create_grid_xyz_created_ids )
.
.
.
```

14. Remove the following lines from the end of the file:

```
uil_file_close.goquit( )
$# Journal file stopped recording at 21-Mar-95 17:10:52
$# P3/PATRAN 1.3-2 has released 68 license(s) to NetLS at 21-Mar-95 17:10:55.
$# Session file patran.ses.08 stopped recording at 21-Mar-95 17:10:55
```

15. Add the following lines to the beginning of the file:

```
FUNCTION pipe()

  GLOBAL REAL rad_1, rad_2
  REAL temp
  IF( rad_2 > rad_1 ) THEN
    temp = rad_1
    rad_1 = rad_2
    rad_2 = temp
  END IF

  ui_override_message( 38000219, "YES" )
```

16. At the end of the file add the line:

```
END FUNCTION
```

17. To debug and verify the pipe pcl function, start PATRAN and open a new database (test.db). After that, type in the command shown below:

```
!!input pipe.pcl
```

At this point, the following lines appear in the command window:

```
$# Compiling: pipe
$# (PCL) Duplicate name defined: ASM_CREATE_GRID_XYZ_CREATED_IDS
$# File:pipe.pcl, Line 25
$# Line is "string asm_create_grid_xyz_created_ids[virtual]"
$# Compilation aborted
```

When PATRAN tried to compile our pipe function, it encountered a problem. Notice that PATRAN tells us that it is on line 25, which is "string asm_create ..." The compiler has found a line where the previous session file re-defined a variable which had already been defined. When a surface is first extruded from a curve, PATRAN version 5.0 somehow "forgets" if it has already defined a variable needed when points are first created. Thus, it re-defines the variable. This is particularly bad when the session file is compiled, since the

function cannot define the variable twice. To remedy the situation, simply ***delete the second line in which the variable is defined.*** (remove line 25 of the file pipe.pcl)

Once again, start a test database and type in the command window:

!!input pipe.pcl

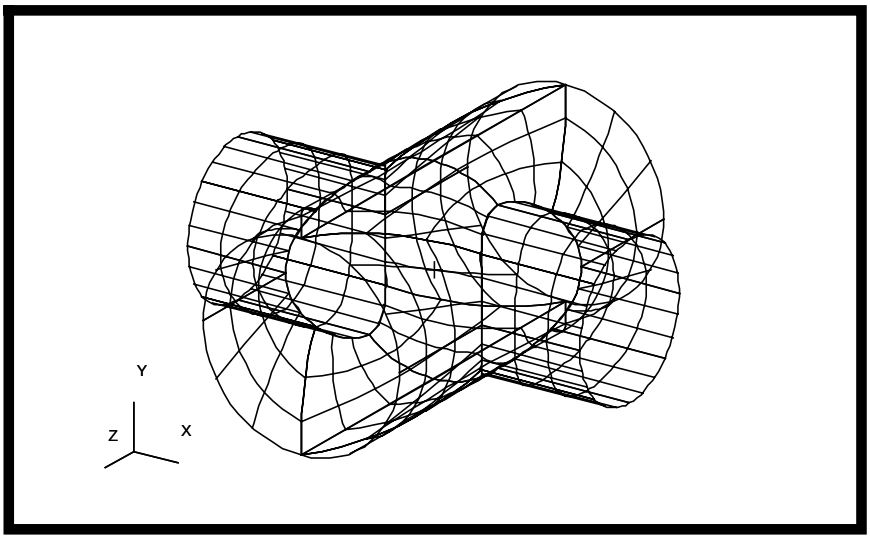
and

REAL rad_1=3,rad_2=4

and finally

pipe()

in the command line. The model should be rebuilt with the new values as shown below.



Close the database to complete this exercise.