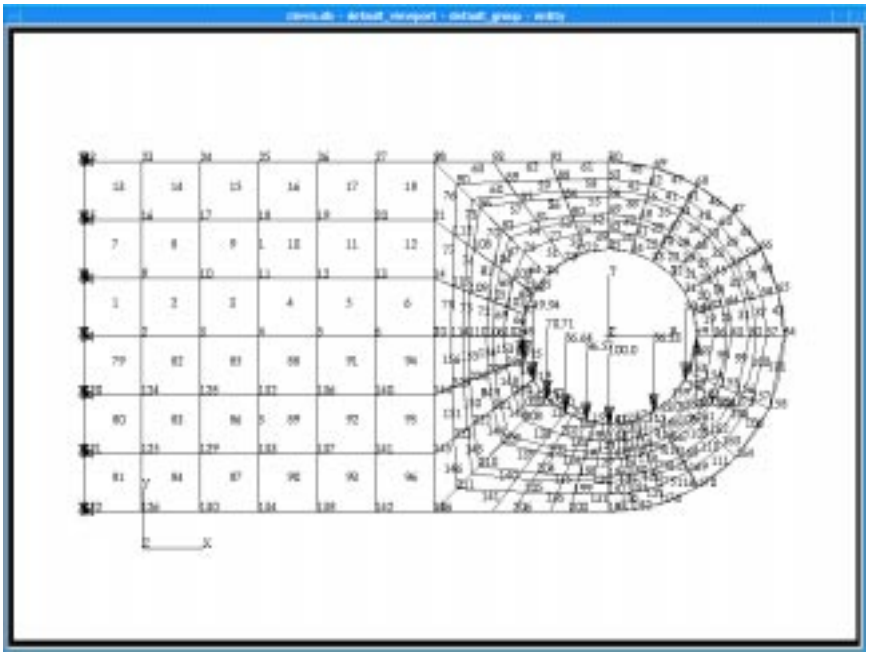
EXERCISE 6

Load Lug Model



Objective:

- Write a function to apply the loads and element properties to the finite element mesh of the lug.

Load Lug Geometry

Exercise Description:

This exercise, lug_load ( ), applies the loads and boundary conditions, material and element properties to the lug model. When complete, the model will have material properties, element properties, a fixed end, and a spatially varying distributed load in the lower half of the pin hole.

Exercise Procedure:

1. Open the database that you have been using for the previous two exercises if it is not already open.

2. Start recording a new session file called **lug_load.ses**.

File/Session/Record ...

lug_load.ses

Apply

3. Create the materials that you are going to put on the model.

◆ **Materials**

Action:	**Create**
Object:	**Isotropic**
Method:	**Manual Input**
Material Name:	**steel**

Input Properties...

Elastic Modulus	**30E6**
Poisson Ratio	**0.3**

Apply

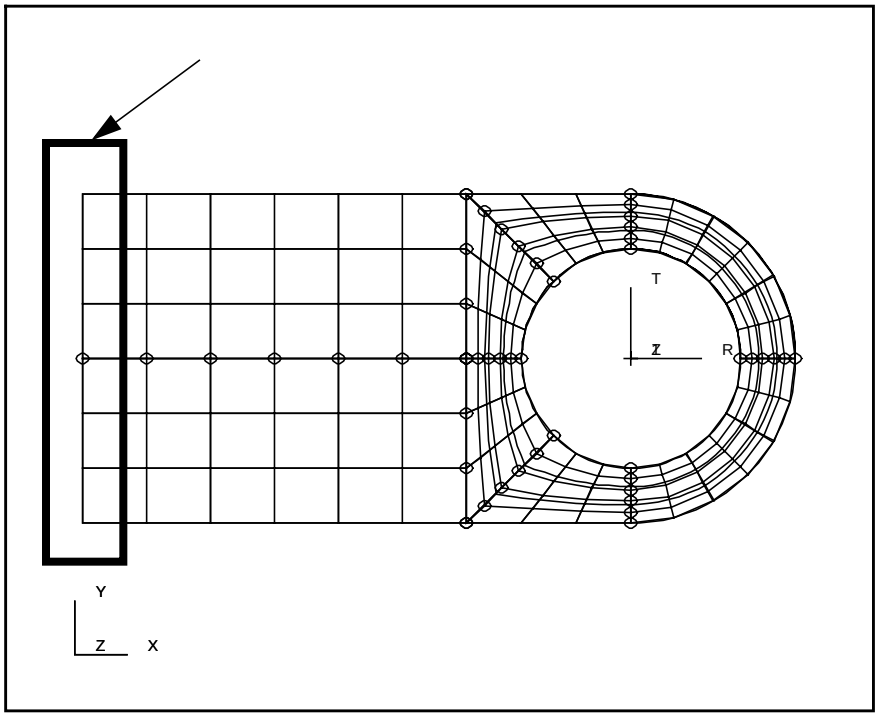
Cancel

4. Create the properties for the steel material.

◆ **Properties**

Action:	**Create**
Dimension:	**2D**
Type:	**Shell**
Property Set Name	**eighth_in_steel_plate**
Input Properties...	
Material Name	**m:steel**
Thickness	**'thickness'**
OK	
Select Members	Select all on screen
Add	
Apply	

5. Secure the left side of the lug.



◆ **Loads/BCs**

Action:	**Create**
Object:	**Displacement**
Type:	**Nodal**
New Set Name	**solid_wall**

Input Data...

Translations	**<0 0 0>**
Analysis Coordinate Frame	**Coord 0**

OK

Select Application Region...

◆ **Geometry**

Curve or Edge Icon

Select Geometry Entities	See the graphic shown above

Add

OK

Apply

6. Create the fields on the model

◆ **Fields**

Action:	**Create**
Object:	**Spatial**
Method:	**PCL Function**
Field Name	**lug_force**
Field Type	

◆ **Vector**

Coordinate System Type

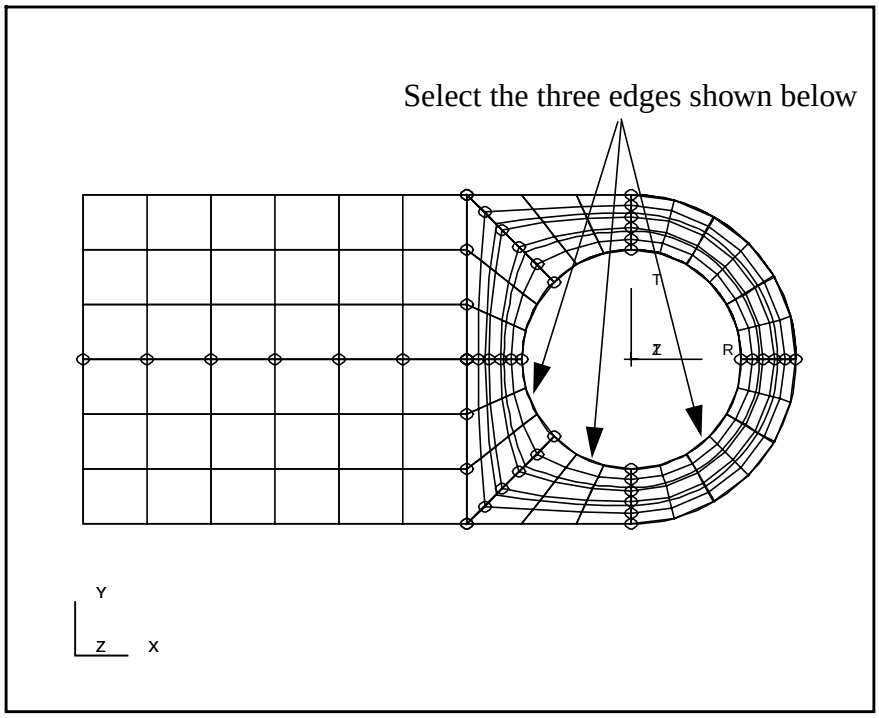
◆ **Real**

Coordinate System	**Coord 1**
Vector Function (’R, ’T, ’Z)	
Second Component	**amplitude*sinr(’T)**

Apply

Load Lug Geometry

7. Apply a force to the inner radius of the lug.



◆ **Loads/BCs**

Action:	**Create**
Object:	**Force**
Type:	**Nodal**
New Set Name	**lug_force**

Input Data...

Spatial Fields	**lug_force**
Force	**f:lug_force**
Analysis Coordinate Frame	**Coord 0**

OK

Select Application Region...

◆ **Geometry**

Select Geometry Entities	See the graphic shown above

Add

OK

Apply

8. Stop recording the session file.

File/Session/Record ...

lug_load.ses

Stop

Cancel

9. Use a vi editor or jot to create a PCL function from the session file. Name the function **lug_load()**. Make sure to end the function.

10. Rename the session file **lug_load.pcl**.

11. Instead of deleting all the Loads/BCs, Materials, Fields, and Properties close the database and quit PATRAN.

File/Quit

12. Create a new file call **p3epilog.pcl**. Enter the following into the file:

```
!!input lug_create.pcl
!!input lug_mesh.pcl
!!input lug_load.pcl
```

Make sure that the p3epilog.pcl file is in your current working directory.

13. Start PATRAN again by typing p3 in your xterm window.

14. Open an new database.

File/New...

completed_lug.db

OK

Click **OK** when the *New Model Preferences Form* appears.

15. Enter the following command:

```
lug_create()
```

After commiting this command the geometry for your lug should be created.

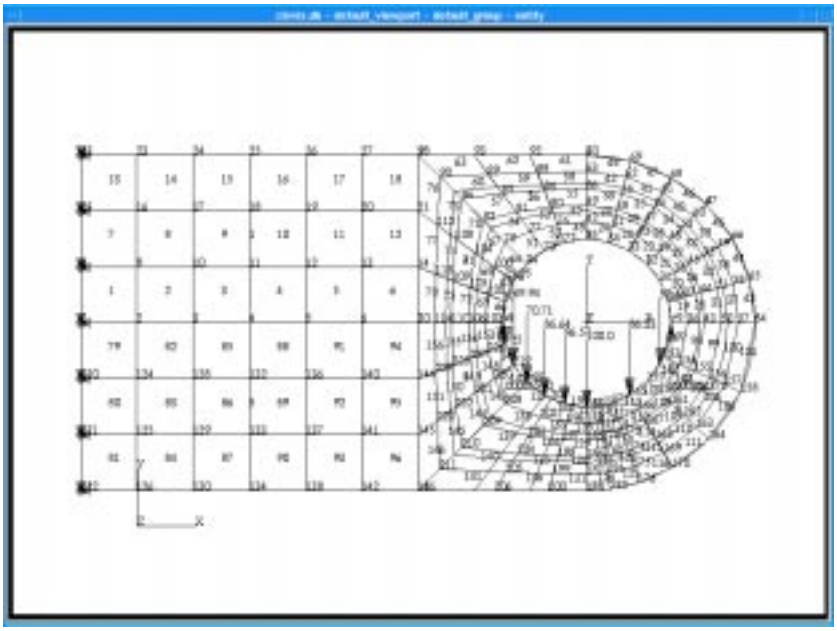
16. Enter the name of the function that should create the mesh seeds and mesh for the lug.

```
lug_mesh()
```

17. After executing this last command the lug model should appear the same as when you created it by hand

```
lug_load()
```

Your model should appear as shown.



18. Quit MSC/PATRAN.

File/Quit

Sample Solution:

```
FUNCTION lug_load( )

/*
 * Purpose:
 *         Load a previously created 2-dimensional model of a lug with
 *          parametric cubic patches.
 *
 * INPUT:
 *         lug_geometry STRING[VIRTUAL] Picklist of the geometry contained
 *                                      in the lug.
 * OUTPUT:
 *         none
 *
 * Side effects:
 *         A 2D lug is loaded from the specified dimensions
 */

       STRING     lug_geometry[], surface_1[], surface_5[]
       STRING     surface_6[], surface_7[], surface_8[]
       STRING     temp[VIRTUAL](VIRTUAL)

material.create( “Analysis code ID”, 1, “Analysis type ID”, @
                  1, “steel”, 0, @
                  “Date: 16-Mar-93 Time: 11:32:24”, @
                  “Isotropic”, 1, @
                  “Directionality”, 1, “Linearity”, 1, @
                  “Homogeneous”, 0, “Linear Elastic”, 1,@
                  “Model Options & IDs”, [“”,””,””,””,””], @
                  [0,0,0,0,0], “Active Flag”, 1, @
                  “Create”, 10, “External Flag”, FALSE, @
                  “Property IDs”, [“Elastic Modulus”,@
                  “Poisson Ratio”], [2,5,0], “Property Values”, @
                  [“30e+6”,”.3”,””] )

elementprops_create( “eighth_in_steel_plate”, 57, 25, 19, 1, 1, @
                  20, [13,20,@
                  36,4037], [5,9,1,1], @
                  [“m:steel”,””,”‘thickness‘”,””], @
                  lug_geometry )

loadsbcs_create( “solid_wall”, “Displacement”, “Nodal”, “”, @
                  “Static”, temp, @
                  “Geometry”, “Coord 0”, 1., @
                  [“< 0 0 0 >”,”< >”], [“”,””] )

fields_create( “lug_force”, “Spatial”, 1, “Vector”, “Real”, @
                  “Coord 1”, “”, “Function”, 1, “”, “T”, “”, “”, @
                  “amplitude * sinr( ’T )”, @
                  “”, FALSE, [0.], @
                  [0.], [0.], [[[0.]]] )

sys_reallocate_string( temp, str_length( surface_6//”.1 “//surface_7//@
                      “.1 “//surface_8//”.1 “))
sys_reallocate_array( temp, 1, 1 )
temp(1) = surface_6//”.1 “//surface_7//”.1 “//surface_8//”.1 “
loadsbcs_create( “lug_force”, “Force”, “Nodal”, “”, “Static”, @
                  temp, “Geometry”, “Coord 0”, 1., @
                  [“f:lug_force”,”< >”], [“”,””] )

END FUNCTION
```

Load Lug Geometry