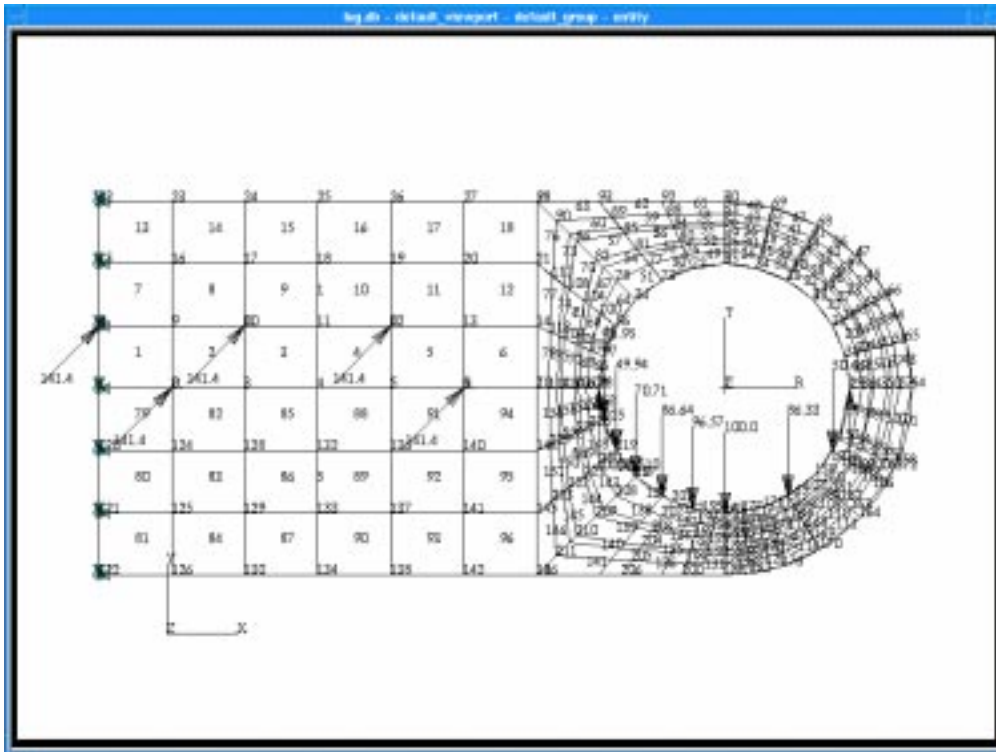


EXERCISE 8

Apply Load to Selected Nodes



Objectives:

- Write a function to apply loads to an array of nodes.



Exercise Description:

In the exercise, `apply_load_to_selected_nodes (nodes, numnod, loads, lname)`, you will pass an array of nodes, the length of the array, and a string array containing the loads to be applied to each node. The function will return the load set name in the variable `lname`. The function should have appropriate error handling. This function will be called by PCL functions you will create in the subsequent exercises.

Files:

All the files that used in this exercise are listed below. Each list includes the file, where it originated, and a summary of information of how it relates to the exercise.

File	Supplied/Created	Description
<code>exercise_08.template</code>	Supplied	A template file that needs to be corrected with the proper PCL syntax.
<code>apply_loads_to_selected_nodes.pcl</code>	Created	Created from the exercise template. This file should be able to be compiled and run in PATRAN with no problems occurring.

Exercise Procedure:

1. Enter the vi editor and edit the PCL function in a file called `exercise_08.template`. Substitute the appropriate PCL syntax for the lines which contain:

```
***** # *****
```

Afer filling in the blanks, save the function with the file name `apply_load_to_selected_nodes.pcl`.

2. Compile the function.

Use the PATRAN compiler by typing `p3pclcomp` in your xterm window.

Enter the command:

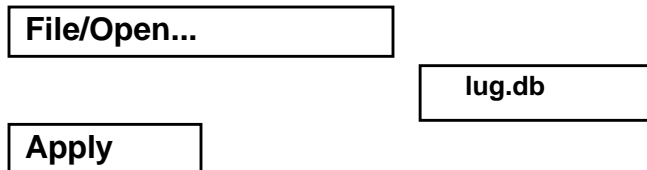
```
!!input apply_load_to_selected_nodes.pcl
```

All the error messages and diagnostics will be written in the lines below.

If no problems arise the messages in the succeeding lines should say compiling, and then compiled.

Exit from the compiler.

3. Start PATRAN and open up the database lug.db.



4. Add a string variable for the name of the load that you are going to assign to the nodes.

```
STRING load_name[ 32 ]
```

5. Since we created the geometry, mesh, etc. earlier we are going to just apply the loads to a few select nodes on the lug.

6. Input the pcl file:

```
!!input apply_load_to_selected_nodes.pcl
```

Type in the following command in the PATRAN command window:

```
apply_load_to_selected_nodes([ 2, 6, 8, 10, 12], 5, @  
[ "<100 100 0>", "< 0 0 0 >"], load_name)
```

The parameters that need to be passed to the function are the following:

The nodes that are going to be selected:

```
([2, 6, 8, 10, 12]
```

The second parameter gives the number of nodes that are going to be selected:

```
5,
```

The third parameter gives the translational forces, the direction of the load, and the rotational forces:

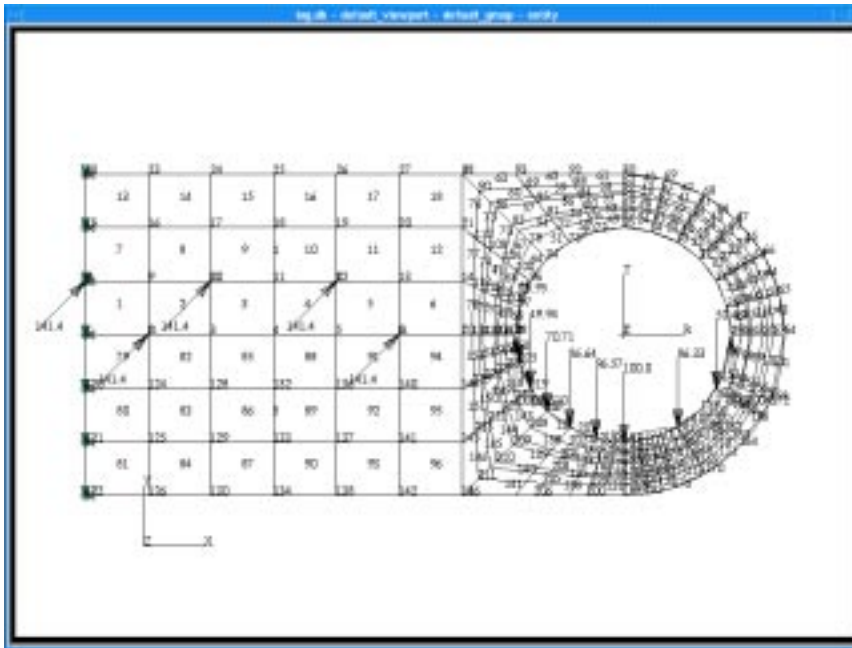
```
[ "<100 100 0>", "<0 0 0>" ]
```

Apply Load to Selected Nodes

The fourth parameter gives the name of the load that is given to those properties.

load_name)

Your model should appear as shown.



Sample Solution:

```
FUNCTION apply_load_to_selected_nodes( nodes, numnod, loads, lname )
/* Purpose:   This function accepts an array of nodes and two load
*             vectors as input and creates a load set.
*
* Input:  nodes      I      integer array of node ids
*         numnod     I      number of nodes in array
*         loads      S(2)   load vector in list processor fmt
*
* Output: lname      S      load set name created
*
* ERROR: The function returns zero for success and 1 for failure.
*/

STRING loads[ ]( ), lname[ ]
INTEGER nodes( ), numnod, i, increment = 100

STRING temp_string[12], nodelist[VIRTUAL](1)
INTEGER status, string_count, string_size

/*
* allocate memory for the virtual string
*/

/*Fill in the parameter list for the following function call*/

status = sys_allocate_string(*****1***** )

IF( status != 0 ) THEN
    ui_write( " Could not allocate memory for virtual string.")
    RETURN status
END IF

string_size = increment
nodelist(1) = "Node"
string_count = 5

/*
* Check to see that a database is open
*/

IF ( db_is_open() ) THEN

/*
* Build the node list
*/

FOR ( i = 1 TO numnod )
    temp_string = str_from_integer( nodes(i) )
    string_count += str_length( temp_string ) + 1
    IF( string_count > string_size ) THEN
        string_size += increment
        sys_reallocate_string(*****2***** )
    END IF
    nodelist(1) = *****3***** Give the value to be assigned to
    ***** the nodelist string for each pass
END FOR

/*
* Build a load set name based on node id of first node in the array
* Execute the P3 Force directive.
*/
```

```
*/

lname = "f_n"//str_from_integer ( nodes(1) )
loadsbc_create ( lname, "Force", "Nodal", "", @
                "Static", ****4***, "FEM", "Coord 0", @
                1., loads, ["", ""])
ELSE
  ui_write("Must have a database open.")
END IF
  sys_free_string( ****5**** )

END FUNCTION
```

```
*1* nodelist, increment  
*2* nodelist, string_size  
*3* nodelist(1)/" "//temp_string  
*4* nodelist  
*5* nodelist
```

