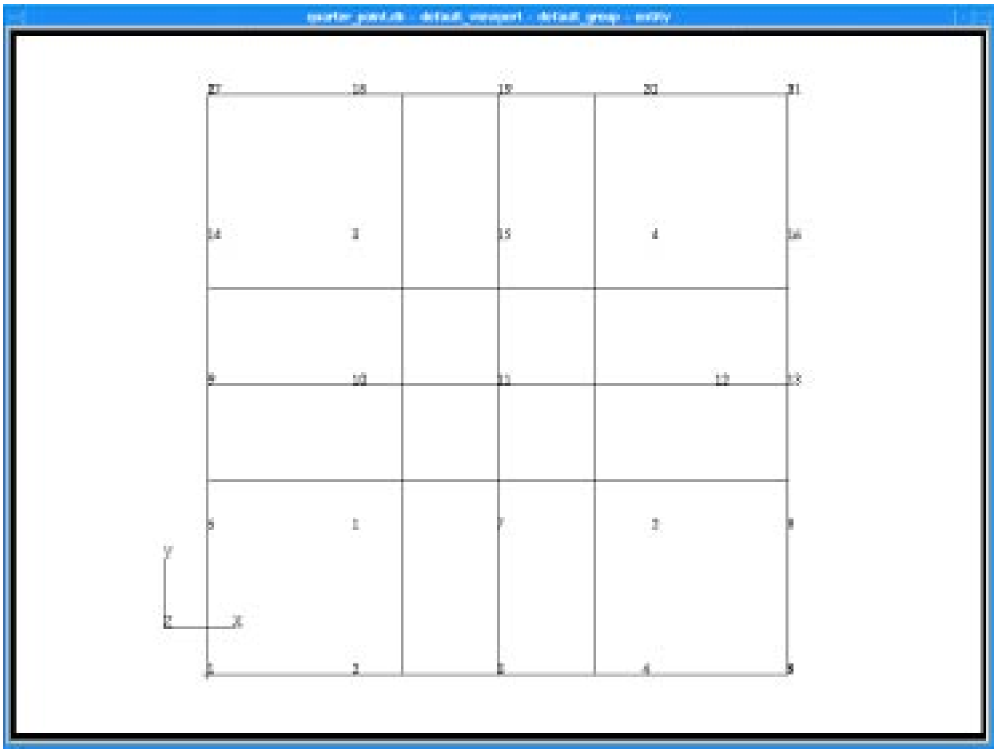
EXERCISE 9

Move Mid-side Node to the Quarter Point


quarter_point.db - default_viewport - default_group - entity
17
18
19
20
21
14
3
15
4
16
9
10
11
12
13
6
1
7
2
8
y
Z
X
1
2
3
4
5

Objectives:

- Write a function to move a node from the midpoint of an edge to the first quarter point of that edge toward a user specified corner node. This function is useful for modeling crack tip elements in some analysis codes.

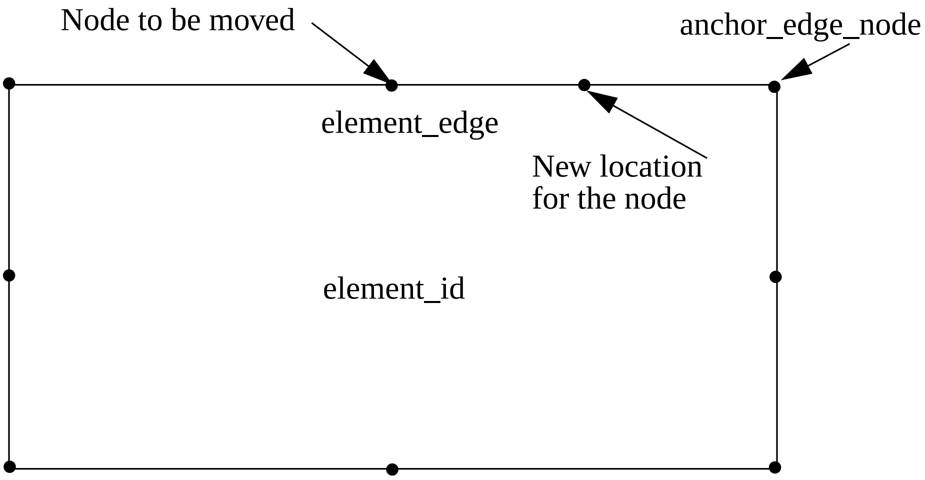
EXERCISE 9

Midside to Quarter

Exercise Description:

This exercise, move_midside_to_quarter(element_id, element_edge, anchor_edge_node), moves a mid-side node on an edge, *element_edge*, of a given element, *element_id*, halfway between the midside node and the corner anchor node, *anchor_edge_node*. The element type should be QUAD8 or TRI6.

QUAD8



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
exercise_09.template	Supplied	A template that is used in order to fill in the missing PCL commands in the code.
move_midside_to _quarter.pcl	Created	Created from exercise_09.template. The completed version of the exercise template.

Exercise Procedure:

1. Edit the PCL function in the file *exercise_09.template*. Replace the blanks with the appropriate PCL expressions. Rename the file to *move_midside_to_quarter.pcl* when you are done.

2. Compile the function.

Type p3pclcomp in your xterm window to make sure that the file runs without any errors.

If there are no error messages written to the screen then type **“exit”** at the p3pclcomp prompt.

3. Start MSC⁄PATRAN by typing p3 in your xterm window.

Enter the command:

!!input move_midside_to_quarter.pcl

into the MSC⁄PATRAN command window.

4. Create the model in which you are going to count the type and number of elements.

File/New...

Group Name	**quarter_point.db**

Apply

In the window that opens up after the database has completed loading click OK

OK

5. Create a surface.

◆ **Geometry**

Action:	**Create**
Object:	**Surface**
Method:	**XYZ**

Apply

6. Create the mesh on the surface.

◆ **Finite Elements**

Action:	**Create**
Object:	**Mesh**
Type:	**Surface**
Global Edge Length	**0.5**
Element Topology	**Quad8**

◆ **IsoMesh**

Surface List	**Surface 1**

Apply

Upon completion of creating the mesh you should have four elements with two nodes per one side of the element.

7. Turn on the display lines and show the labels.

Show Labels

Display Lines

8. Next you are going to move one of the middle nodes half the distance to the edge of the model.

Type in the command:

```
!!input move_midside_to_quarter.pcl
```

The file should once again compile without any problems.

9. Execute the function.

At the command window type:

```
move_midside_to_quarter( 4, 1, 13)
```

The following lines should appear in the history window of PATRAN.

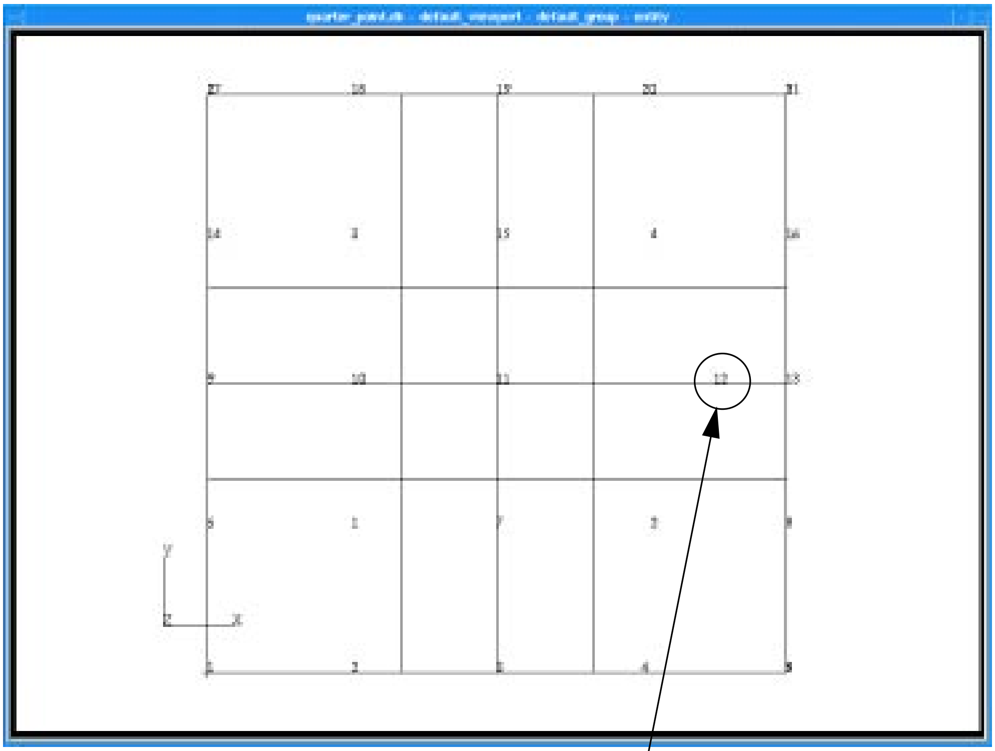
```
$# 1 Line created: Line 1
$# 1 Pointcreated: Point 7
$# === 1 node modified.  ID = 1.
$# 1 Point Deleted: Point 7
$# 1 Curve Deleted: Curve 1
```

10. Click on the screen repaint icon in the top menu bar.

The Screen Repaint Icon

You should notice that the right node has moved closer to the right side of the model.

Your QUAD8 elements should appear as shown.


quarter_point.db - default_viewport - default_group - entity
17
18
19
20
21
14
3
15
4
16
9
10
11
12
13
6
1
7
2
8
Y
Z
X
1
2
3
4
5

Notice that Node 12 is shifted

Sample Solution:

```
 FUNCTION move_midside_to_quarter( element_id, element_edge, @
                                         anchor_edge_node)

/* Purpose:    This function moves the midside node of a user specified
 *             element edge to the quarter point toward the anchor
 *             edge node.
 *             Currently supports quad/8 and tri/6 only.
 *
 * Input:  element_id            I      Element id
 *         element_edge          I      element edge id in P3 conventions
 *         anchor_edge_node      I      node id
 *
 * Output: none
 *
 */

    INTEGER element_edge, element_id
    STRING asm_line_3point_created_ids[VIRTUAL]
    STRING asm_grid_interp_cur_created_ids[VIRTUAL]
    STRING fem_modify_node__nodes_modified[VIRTUAL]
    STRING asm_delete_point_deleted_ids[VIRTUAL]
    STRING asm_delete_curve_deleted_ids[VIRTUAL]

    /* Indices in node list array for an element edge */

    INTEGER n1_q(4) = 1, 2, 3, 4
    INTEGER n2_q(4) = 2, 3, 4, 1
    INTEGER mn_q(4) = 5, 6, 7, 8

    INTEGER n1_t(3) = 1, 2, 3
    INTEGER n2_t(3) = 2, 3, 1
    INTEGER mn_t(3) = 4, 5, 6

    INTEGER anchor_edge_node, status, eid(1)
    INTEGER tria6, quad8, etop(1)
    INTEGER node(8)
    INTEGER node1, node2, midnd
    REAL location
    LOGICAL tri

      /*
       * Parse element edge string to get element id and edge id
       */

       eid(1) = element_id

       /*
       * Check to see if element is a quad or tri with midside nodes
       */

       status = db_get_elem_etop( 1, eid, etop )
       IF( status != 0 ) THEN
           msg_to_form( status, 4, 13000000, 1, 1., "" )
           RETURN status
       END IF
       ************* 1 ****************
       ************* 2 ****************
```

```
/*
* Ensure element is a tri or quad with midside nodes
*/

IF( etop(1) == tria6 ) THEN
    tri = TRUE
ELSE IF ( etop(1) == quad8 ) THEN
    tri = FALSE
ELSE
    ui_write(“Element is not a Tria6 or Quad8. Aborted”)
    RETURN 1
END IF

/*
* Get the element node list
*/

******************* 3 *******************
IF( status != 0 ) THEN
    msg_to_form( status, 4, 13000000, 1, 1., ““ )
    RETURN status
END IF

/*
* Do a database commit so that the undo command only
* undoes the node modification
*/

uil_db_commit( “Move midside node”)

/*
* Get node two end nodes and midside node on specified edge
*/

IF( tri ) THEN
    node1 = node( n1_t(element_edge) )
    node2 = node( n2_t(element_edge) )
    midnd = node( mn_t(element_edge) )
ELSE
    node1 = node( n1_q(element_edge) )
    node2 = node( n2_q(element_edge) )
    midnd = node( mn_q(element_edge) )
END IF

/*
* Create a line thru these three nodes
*/

asm_const_line_3point( “#”, “Node “************* 4 ************
                          1, 0.5, asm_line_3point_created_ids )

/* Create an interior grid on this line at the quarter point. The
* node is to be moved toward the specified anchor_node. If the
* anchor node is at node1, then the distance ratio (L2/L1),
* (midside node - node2)/(node1 - midside node), is 3 to 1. If the
* anchor node is node2, the the ratio is 1 to 3.
*/

IF( anchor_edge_node == node1 ) THEN
    location = 3.0
ELSE
    location = 0.333
END IF
```

```
   asm_const_grid_interp_curve( “#”, “Curve #”, location, 1, @
                             asm_grid_interp_cur_created_ids )/* Now move the midside
       node to the new grid location */

   fem_modify_nodes(                                                     @
   /*    Node list    */    “Node “//STR_FROM_INTEGER( midnd ),            @
   /*    newnid_list  */    “ “,                                           @
   /*      rcid       */    “ “,                                           @
   /*      acid       */    ******* 5 *******,                             @
   /*    point_list   */    [0,0,0,0,1,0,0,0,0,0],                         @
   /* Modified nodes  */    fem_modify_node__nodes_modified )

   /* Delete the construction grid and line */

   asm_delete_point( ************ 6 ************,        @
                     asm_delete_point_deleted_ids )
   asm_delete_curve( *********** 7 *************,    @
                     asm_delete_curve_deleted_ids )

   RETURN 0

END FUNCTION
```

```
*1* fem_get_patran25_etop( 3, 6, tria6 )
*2* fem_get_patran25_etop( 4, 8, quad8 )
*3* status = db_get_nodes_for_elems( 1, 8, eid, node )
*4* //STR_FROM_INTEGER( node1 ), @
    “Node “//STR_FROM_INTEGER( midnd ), @
    “Node “//STR_FROM_INTEGER( node2 ), @
*5* “Point #”
*6* “Point #”
*7* “Curve #”
```