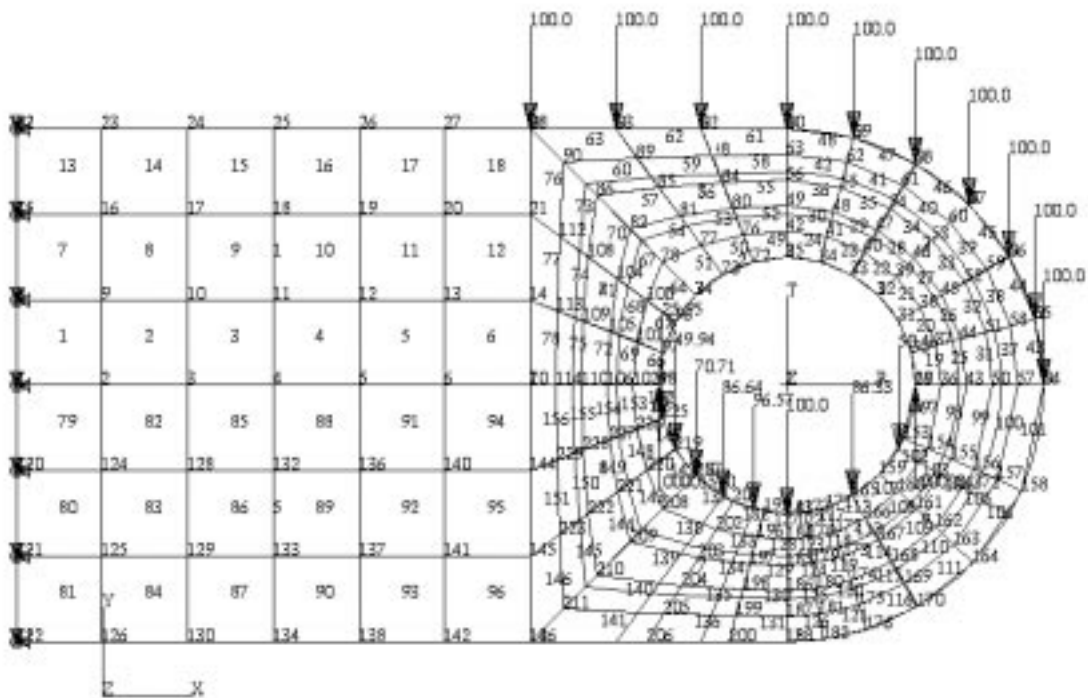


EXERCISE 13

Apply Load Function Graphical User Interface



Objectives:

- Write a class to create the graphical user interface, GUI, for the apply_load_to_selected_nodes function.



Exercise Description:

This exercise, class `load_form`, will create a form that handles the user interface for the loads. It provides two databoxes to enter force and moment vectors in P3/PATRAN's list processor format. Finally the apply button will retrieve the information from the databoxes, and call `apply_load_to_selected_nodes` function created in Exercise 8.

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_13.template</code>	Supplied	A template file that you may use to fill in the missing PCL calls that should be added to the function.
<code>load_form.pcl</code>	Created	The file will be created from the <code>exercise_13.template</code> after the file has been completed with the correct PCL codes.

Exercise Procedure:

1. Edit the PCL function in the file *`exercise_13.template`*. Replace the blanks with the appropriate PCL expressions. Rename the file to *`load_form.pcl`* when you are done.

You should use the pre-defined variables for the widget size and spacing to make the form presentable.

Use MSC/PATRAN list processor calls to parse out the entities inside the Select Nodes databox. As discussed in the lecture, the list processor calls will accept a string as an input and output, and unique ID's of a user specified attribute such as Nodes, Elements, etc.

You may also use the template file your instructor will provide for you to add the missing PCL code. The form that you will be creating should look like the one shown below..

The image shows a dialog box titled "Load Create Form". It has a yellow background and a blue title bar. Inside the dialog, there are several input fields and a checkbox. The "Force Vector" field contains the text "<0.0 0.0 0.0 >". The "Moment Vector" field also contains "<0.0 0.0 0.0 >". Below these is a checkbox labeled "Auto Execute" which is currently unchecked. Underneath the checkbox is a "Select Nodes" field containing the text "Node 24 36 75 99 305". At the bottom of the dialog are two buttons: "Apply" and "Cancel".

2. Pre-Process the function.

```
%cpp -I/patran/patran3/customization load_form.pcl  
load_form.cpp
```

Note: Check the previous exercise for the machine specific commands necessary to run the C Pre-Processor.

Once again the -I switch tells cpp to search in /patran/patran3/customization directory for the include files.

3. When you have completed the code for generating the function, type the command p3pclcomp to compile the function without having to start the full PATRAN program.

```
!!input load_form.cpp
```

The prompt should say that it is compiling and then compiled, if there are no errors in the code that you have just generated.

Apply Load Form

4. Edit the **p3epilog.pcl** file once again so that it contains:

```
!!input load_form.cpp
!!input apply_load_to_selected_nodes.pcl
!!input training.pcl
training.init()
```

5. Start MSC/PATRAN once again making sure that you have these two files in your current working directory:

1. p3epilog.pcl
2. p3prolog.pcl

The p3prolog.pcl file should still contain the values for the global real variables needed for the model of the lug.

6. Open the database, **lug.db**.

File/Open ...

lug.db

Apply

7. Test the function by bringing up the Load Create Form under the exercise 13 menu option..

Training/Exercise 13

Force Vector

<0.0 -100.0 0.0>

Moment Vector

<0.0 0.0 0.0>

Select Nodes

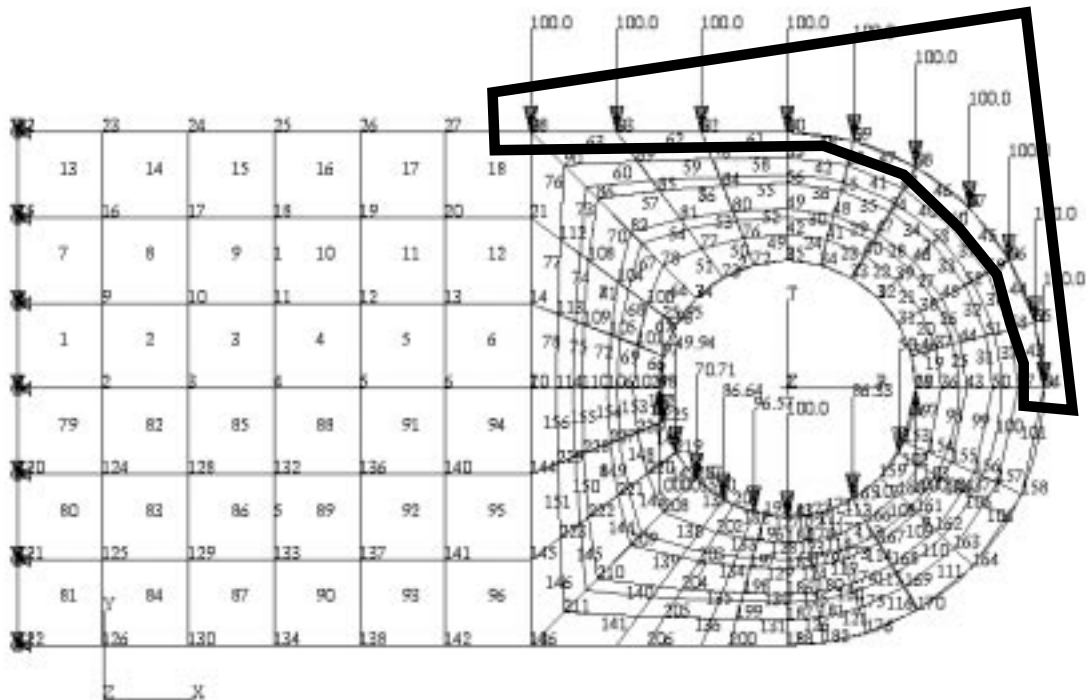
Use the ctrl picking to select the nodes. Refer to next page

Apply



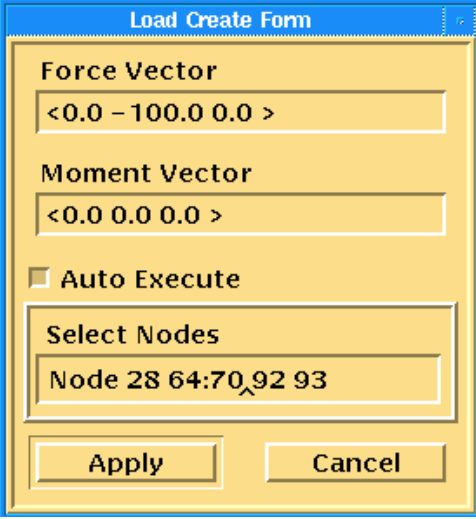
Now, click in the Select Nodes databox and screen select the nodes shown on the next page.

Use the polygon pick to select a series of nodes. To do this, hold the <control> key and click with the left mouse button enclosing the nodes you wish to select. To close the polygon, click on the same vertex in which you started.



Apply Load Form

Your form should appear as shown below.



The screenshot shows a dialog box titled "Load Create Form". It contains the following fields and controls:

- Force Vector:** A text input field containing the value `<0.0 -100.0 0.0 >`.
- Moment Vector:** A text input field containing the value `<0.0 0.0 0.0 >`.
- Auto Execute:** A checkbox that is currently unchecked.
- Select Nodes:** A text input field containing the value `Node 28 64:70, 92 93`.
- Buttons:** Two buttons at the bottom, labeled "Apply" and "Cancel".

Upon clicking Apply, MSC/PATRAN will create a new load set.

Sample Solution:

```
/*$$ Use of PCL in creating customized forms/widgets
*
* Purpose:
*     Create the user interface for the function
*     apply_load_to_selected_nodes function.
*
* Input:
*     <None>
*
* Output:
*     <None>
*
* Log:
*
* Notes:
*
*/

*****1***** Insert the include files discussed in lecture
*****1*****

CLASS load_form

    /* Variable initialization */

    CLASSWISE          widget form_id,          @
                        dbx1_id,                @
                        dbx2_id,                @
                        apply_button,           @
                        cancel_button,          @
                        select_frame,           @
                        select_data_box1

    FUNCTION INIT()

        REAL            y_loc, select_data_box_width

        /*
        * Create the form
        */

        /* Add all the parameters to the next ui_form_create command. */

        form_id=ui_form_create (                @
            /* callback */      ****2****      @
            /* x */             ****2****      @
            /* y */             ****2****      @
            /* position */      ****2****      @
            /* width */         ****2****      @
            /* height */        ****2****      @
            /* label */         ****2****      @
            /* iconname */      ****2****      )

        y_loc = ****3**** Keep a running tab on the y location in this form

        /*
        * Create the "Force Vector Databox"
        */
```



```

dbx1_id= ui_databox_create(                                     @
/*  parent  */      form_id,                                  @
/*  callback */      "",                                       @
/*    x     */      UNFRAMED_L_MARGIN,                         @
/*    y     */      y_loc,                                       @
/* label_length */   0.0,                                       @
/* box_length */   DBOX_WID_SINGLE,                             @
/*  label    */      "Force Vector",                             @
/*  value    */      "<0.0 0.0 0.0 >",                           @
/* label_above */   TRUE,                                       @
/* datatype  */      "STRING",                                   @
/* num_vals  */      1                                           )

y_loc += ****4****

/*
 * Create the "Enter the Moment Vector Databox"
 */

dbx2_id= ui_data_box_create(                                   @
/*  parent  */      form_id,                                  @
/*  callback */      "",                                       @
/*    x     */      ****5****                                    @
/*    y     */      ****5****                                    @
/* label_length */   ****5****                                    @
/* box_length */   ****5****                                    @
/*  label    */      "Moment Vector",                             @
/*  value    */      "<0.0 0.0 0.0 >",                           @
/* label_above */   TRUE,                                       @
/* datatype  */      "STRING",                                   @
/* num_vals  */      1                                           )

y_loc += *****6*****

/*
 * select data frame for element selection
 */

select_frame = ui_selectframe_create (                       @
/*  parent  */      ****7****                                    @
/*  callback */      ****7****                                    @
/* Left_margin */   ****7****                                    @
/*    y     */      ****7****                                    @
/* col_width */   SFRAME_WID_SINGLE,                             @
/* height    */   SFRAME_1SDB_HGT_LABOVE,                       @
/* always_one */   "Auto Execute",                               @
/* recycle   */   TRUE                                           )

select_data_box_width= SDBOX_WID_SINGLE - SFRAME_R_MARGIN

```

```

/*
 * Create the data boxes to be placed inside the dataframe
 */

select_data_box1 = ui_selectdatabox_create (      @
/*parent_frame */      ****8****      @
/* callback */      ****8****      @
/* Left_margin */      ****8****      @
/* y */      ****8****      @
/*label_length */      ****8****      @
/* Box_length */      ****8****      @
/* label */      " Select Nodes",      @
/*default_value */      " " ,      @
/* label_above */      TRUE,      @
/* data_type */      "NODE",      @
/* prompt */      "select Node filter"      )

y_loc += SFRAME_LABEL_HGT + @
        SFRAME_1SDB_HGT_LABOVE + @
        SFRAME_2EDGE + @
        INTER_WIDGET_SPACE

/*
 * Create the "Apply" button
 */

apply_button = ui_button_create(      @
/* parent */      ****9****      @
/* callback */      ****9****      @
/* x */      BUTTON_HALF_X_LOC1,      @
/* y */      y_loc,      @
/* width */      BUTTON_WID_HALF,      @
/* height */      0.0,      @
/* label */      "Apply",      @
/* labelinside */      TRUE,      @
/* highlight */      TRUE      )

/*
 * Create the "Cancel" button
 */

cancel_button = ui_button_create(      @
/* parent */      ****10****      @
/* callback */      ****10****      @
/* x */      BUTTON_HALF_X_LOC2,      @
/* y */      y_loc,      @
/* width */      BUTTON_WID_HALF,      @
/* height */      0.0,      @
/* label */      "Cancel",      @
/* labelinside */      TRUE,      @
/* highlight */      FALSE      )

y_loc += BUTTON_DEFAULT_HGT + FORM_B_MARGIN

/* what is the value of the "parm" in the next call. refer to your course notes and
the on line help */

ui_wid_set( form_id, ****11****, y_loc )

END FUNCTION /* init */

FUNCTION display

    ui_form_display( "load_form" )

```

```

END FUNCTION /* display */

FUNCTION apply_cb

    STRING load_value_string [40] (2)
    STRING load_set_name [40]

    STRING node_list [VIRTUAL]

    INTEGER status, num_of_nodes
    INTEGER node_array (VIRTUAL)

    *****12***** Get the values of the Force from the databox
    *****12***** Get the vaules of the Moment
    *****12***** Get the nodes in list processor format

! The value of the force vector entered is `load_value_string (1)`
! The value of the moment vector entered is `load_value_string (2)`

    /*
    * parse the node list
    */

    /* Enter all the parameters for the fem_u_count_id_list call */

    num_of_nodes = fem_u_count_id_list( *****13***** )

    IF (status != 0 ) THEN
        msg_to_form( status, 4, 14000000, 1, 1., "" )
        RETURN status
    END IF

! the number_of_nodes selected= `num_of_nodes`

    IF ( num_of_nodes == 0 ) THEN
        ui_write ("You must select nodes before "// @
                "clicking on apply !!! " )
        RETURN 1
    END IF

    /*
    * allocate the size of the node integer array
    */

    sys_allocate_array( node_array, 1, num_of_nodes )

    /* Enter all the parameters for the fem_u_get_id_list call */

    status = fem_u_get_id_list(*****14*****)

    /*
    * calling the apply_load_to_selected_nodes function
    */

    status = apply_load_to_selected_nodes ( @
/* input--array of the nodes selected */           node_array,           @
/* input--number of nodes */                       num_of_nodes,         @
/* input--load value string array*/                 load_value_string,    @
/* output--name of the load set created*/           load_set_name         )

```

```

*1* #include "appforms.p"
*1* #include "lpenums.i"
*2* "", FORM_X_LOC, FORM_Y_LOC, "UL", FORM_WID_SML,
    FORM_HGT_FULLL, "Load Create Form", ""
*3* FORM_T_MARGIN
*4* DBOX_HGT_LABOVE + INTER_WIDGET_SPACE
*5* UNFRAMED_L_MARGIN, y_loc, 0.0, DBOX_WID_SINGLE,
*6* DBOX_HGT_LABOVE + INTER_WIDGET_SPACE
*7* form_id, "apply_cb", FORM_L_MARGIN, y_loc,
*8* select_frame, "", SFRAME_L_MARGIN, SDBOX_Y_LOC1_LABOVE,
    0.0, select_data_box_width,
*9* form_id, "apply_cb",
*10* form_id, "cancel_cb",
*11* "HEIGHT"
*12* ui_wid_get( dbx1_id, "VALUE", load_value_string (1) )
    ui_wid_get( dbx2_id, "VALUE", load_value_string (2) )
    ui_wid_get_vstring( select_data_box1, "VALUE", node_list)
*13* LP_SUBLIST_NODE, node_list, TRUE, status
*14* LP_SUBLIST_NODE, node_list, num_of_nodes, TRUE, node_array

```

```

END CLASS /* load_form */
END FUNCTION /* cancel_cb */
select_focus.exit()
ui_form_hide("load_form")
FUNCTION cancel_cb
END FUNCTION /* apply_cb */
END IF
WRITE ( "error creating the requested load set !!!" )
ELSE
WRITE ( "load set has been created properly " )
IF ( status == 0 ) THEN

```
