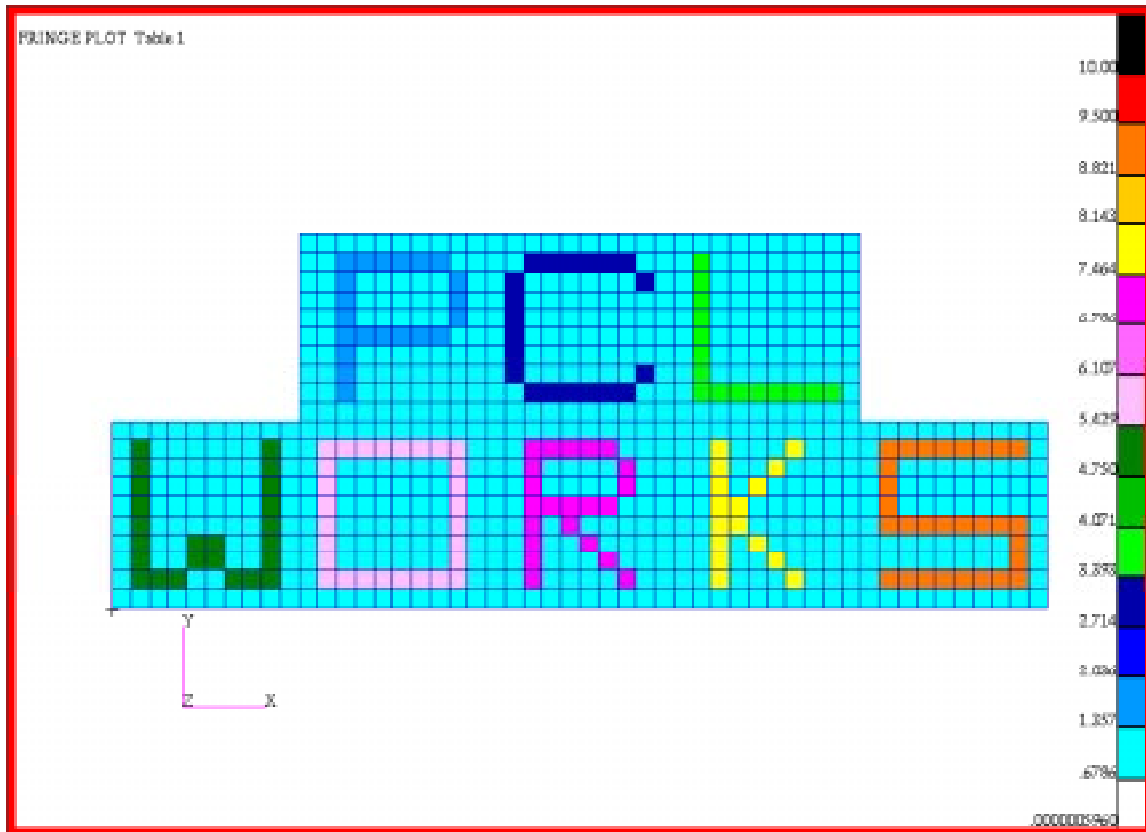


EXERCISE 10

Element Scalar Data Tables



Objectives:

- Understand the Element Scalar Utilities
- Make a simple Element Scalar Plot.



Exercise Description:

In this exercise it is necessary to create a set of elements, and a set of values which are mapped onto these elements and are interpreted as colors. After doing this, create an element scalar table and plot it to the screen.

Exercise Procedure:

1. Change your directory to `ex10`. Enter the vi editor and edit the PCL function in a file called `exercise_10.template`. Substitute the appropriate PCL syntax or create new function for the lines which contain:

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

After filling in the blanks, save the function and name the file `elem_res.pcl`.

2. Compile the function. Run the C Preprocessor by typing at the shell prompt:

```
/usr/lib/cpp -l/patran/patran3/customization plot.pcl plot_erase.cpp
```

Refer to the previous exercises for the machine specific commands.

Start MSC/PATRAN by typing `p3` in your xterm window.

Enter the command `!!input plot_erase.cpp` into the MSC/PATRAN command window.

All the error messages and diagnostics will be written to the MSC/PATRAN command window.

3. Run the session file called `create.ses` to create surfaces and finite elements. Test the function yourself by typing in the command window:

```
create_table()
```

and you will see the display as shown on the cover of this exercise.

Exercise Template:

```
FUNCTION create_table()

INTEGER num_elems, status
INTEGER elem_ids(VIRTUAL), num_values(VIRTUAL), id
REAL values(VIRTUAL)

sys_allocate_array(elem_ids, 1, 800)
sys_allocate_array(num_values, 1, 800)
sys_allocate_array(values, 1, 800)

num_elems = 800

oi_get_results("P", 501, elem_ids, num_values, values)
oi_get_results("C", 601, elem_ids, num_values, values)
oi_get_results("L", 701, elem_ids, num_values, values)
oi_get_results("W", 001, elem_ids, num_values, values)
oi_get_results("O", 101, elem_ids, num_values, values)
oi_get_results("R", 201, elem_ids, num_values, values)
oi_get_results("K", 301, elem_ids, num_values, values)
oi_get_results("S", 401, elem_ids, num_values, values)
dump num_elems
status = *****]*****

IF ( status != 0 ) THEN
    msg_to_form(status, 2, 0, 0, 0., "")
END IF

fem_verify_display(*****2*****)

END FUNCTION /* create_table */

FUNCTION oi_get_results(char, start, eids, num_val, values)

STRING char[]
INTEGER start, eids(), num_val(), i
REAL values(), temp_values(100)

SWITCH (char)

    CASE ( "P" )

        temp_values = [ 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., @
            1., 1., 2., 1., 1., 1., 1., 1., 1., 1., 1., @
            1., 1., 2., 1., 1., 1., 1., 1., 1., 1., 1., @
            1., 1., 2., 1., 1., 1., 1., 1., 1., 1., 1., @
            1., 1., 2., 2., 2., 2., 2., 2., 1., 1., @
            1., 1., 2., 1., 1., 1., 1., 1., 2., 1., @
            1., 1., 2., 1., 1., 1., 1., 1., 2., 1., @
            1., 1., 2., 1., 1., 1., 1., 1., 2., 1., @
            1., 1., 2., 2., 2., 2., 2., 2., 1., 1., @
            1., 1., 1., 1., 1., 1., 1., 1., 1., 1. ]

    CASE ( "C" )

        temp_values = [ 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., @
            1., 1., 3., 3., 3., 3., 3., 3., 1., 1., @
            1., 3., 1., 1., 1., 1., 1., 1., 3., 1., @
            1., 3., 1., 1., 1., 1., 1., 1., 1., 1., @
            1., 3., 1., 1., 1., 1., 1., 1., 1., 1., @
            1., 3., 1., 1., 1., 1., 1., 1., 1., 1., @
            1., 3., 1., 1., 1., 1., 1., 1., 1., 1., @
            1., 3., 1., 1., 1., 1., 1., 1., 3., 1., @
```

```

1., 1., 3., 3., 3., 3., 3., 3., 1., 1., @
1., 1., 1., 1., 1., 1., 1., 1., 1., 1 ]

```

```

CASE ( "L" )

```

```

temp_values = [ 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., @
1., 4., 4., 4., 4., 4., 4., 4., 4., 1., @
1., 4., 1., 1., 1., 1., 1., 1., 1., 1., @
1., 4., 1., 1., 1., 1., 1., 1., 1., 1., @
1., 4., 1., 1., 1., 1., 1., 1., 1., 1., @
1., 4., 1., 1., 1., 1., 1., 1., 1., 1., @
1., 4., 1., 1., 1., 1., 1., 1., 1., 1., @
1., 4., 1., 1., 1., 1., 1., 1., 1., 1., @
1., 4., 1., 1., 1., 1., 1., 1., 1., 1., @
1., 1., 1., 1., 1., 1., 1., 1., 1., 1 ]

```

```

CASE ( "W" )

```

```

temp_values = [ 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., @
1., 5., 5., 5., 1., 1., 5., 5., 5., 1., @
1., 5., 1., 1., 5., 5., 1., 1., 5., 1., @
1., 5., 1., 1., 5., 5., 1., 1., 5., 1., @
1., 5., 1., 1., 1., 1., 1., 1., 5., 1., @
1., 5., 1., 1., 1., 1., 1., 1., 5., 1., @
1., 5., 1., 1., 1., 1., 1., 1., 5., 1., @
1., 5., 1., 1., 1., 1., 1., 1., 5., 1., @
1., 5., 1., 1., 1., 1., 1., 1., 5., 1., @
1., 1., 1., 1., 1., 1., 1., 1., 1., 1 ]

```

```

CASE ( "O" )

```

```

temp_values = [ 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., @
1., 6., 6., 6., 6., 6., 6., 6., 6., 1., @
1., 6., 1., 1., 1., 1., 1., 1., 6., 1., @
1., 6., 1., 1., 1., 1., 1., 1., 6., 1., @
1., 6., 1., 1., 1., 1., 1., 1., 6., 1., @
1., 6., 1., 1., 1., 1., 1., 1., 6., 1., @
1., 6., 1., 1., 1., 1., 1., 1., 6., 1., @
1., 6., 1., 1., 1., 1., 1., 1., 6., 1., @
1., 6., 6., 6., 6., 6., 6., 6., 6., 1., @
1., 1., 1., 1., 1., 1., 1., 1., 1., 1 ]

```

```

CASE ( "R" )

```

```

temp_values = [ 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., @
1., 1., 7., 1., 1., 1., 1., 7., 1., 1., @
1., 1., 7., 1., 1., 1., 7., 1., 1., 1., @
1., 1., 7., 1., 1., 7., 1., 1., 1., 1., @
1., 1., 7., 1., 7., 1., 1., 1., 1., 1., @
1., 1., 7., 7., 7., 7., 7., 1., 1., 1., @
1., 1., 7., 1., 1., 1., 1., 7., 1., 1., @
1., 1., 7., 1., 1., 1., 1., 7., 1., 1., @
1., 1., 7., 7., 7., 7., 7., 1., 1., 1., @
1., 1., 1., 1., 1., 1., 1., 1., 1., 1 ]

```

```

CASE ( "K" )

```

```

temp_values = [ 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., @
1., 1., 8., 1., 1., 1., 8., 1., 1., 1., @
1., 1., 8., 1., 1., 8., 1., 1., 1., 1., @
1., 1., 8., 1., 8., 1., 1., 1., 1., 1., @

```

```

1., 1., 8., 8., 1., 1., 1., 1., 1., 1., @
1., 1., 8., 8., 1., 1., 1., 1., 1., 1., @
1., 1., 8., 1., 8., 1., 1., 1., 1., 1., @
1., 1., 8., 1., 1., 8., 1., 1., 1., 1., @
1., 1., 8., 1., 1., 1., 8., 1., 1., 1., @
1., 1., 1., 1., 1., 1., 1., 1., 1., 1 ]

```

```

CASE ( "S" )

```

```

temp_values = [ 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., @
1., 9., 9., 9., 9., 9., 9., 9., 9., 1., @
1., 1., 1., 1., 1., 1., 1., 1., 9., 1., @
1., 1., 1., 1., 1., 1., 1., 1., 9., 1., @
1., 9., 9., 9., 9., 9., 9., 9., 9., 1., @
1., 9., 1., 1., 1., 1., 1., 1., 1., 1., @
1., 9., 1., 1., 1., 1., 1., 1., 1., 1., @
1., 9., 1., 1., 1., 1., 1., 1., 1., 1., @
1., 9., 9., 9., 9., 9., 9., 9., 9., 1., @
1., 1., 1., 1., 1., 1., 1., 1., 1., 1 ]

```

```

END SWITCH

```

```

FOR ( i = start TO start + 99)

```

```

*****3*****
*****4*****
*****5*****

```

```

END FOR

```

```

END FUNCTION /* oi_get_results */

```

```

*1*ga_elem_scalar_single_create("Table 1", num_elems, elem_ids, @
num_values, values, id)
*2* default_group", 1, 0., 10., 9.5, "ELEMENT"
*3* ids(i) = i
*4* num_val(i) = 4
*5* values(i) = temp_values(i - start + 1)

```