

**\*ALE\_STRUCTURED\_MESH\_TRIM**

Purpose: This keyword does trim/untrim operations on a structured ALE mesh generated by \*ALE\_STRUCTURED\_MESH card. It can have multiple cards and each card represents one mesh trim/untrim operation. These operations are done one by one, in the order of their cards' appearance. Please see Example 1.

| Card 1   | 1     | 2      | 3    | 4    | 5      | 6      | 7      | 8      |
|----------|-------|--------|------|------|--------|--------|--------|--------|
| Variable | MSHID | OPTION | OPER | FLIP | E1     | E2     | E3     | E4     |
| Type     | I     | A      | I    | I    | I or F | I or F | I or F | I or F |
| Default  | 0     | none   | 0    | 0    | None   | none   | none   | None   |

| <b>VARIABLE</b> | <b>DESCRIPTION</b>   |
|-----------------|--|
| MSHID           | S-ALE Mesh ID. A unique number must be specified.  |
| OPTION          | There are six available options. They are: trim by PARTSET, SEGSET, PLANE, CYLINDER, BOXCOR, BOXCPT and SPHERE.  |
| OPER            | To trim or un-trim, i.e. to delete the picked elements or keep them. Default value is 0 which means to trim; 1 to keep.  |
| FLIP            | There tells to pick which elements, "inside" or "outside". For PARTSET and SEGSET options, "outside" is defined as the region to which the segment normal point. Default value is 0 which means "outside"; 1 "inside". |
| E1, E2, E3, E4  | Those values have different means for different options. Please see the table below.   |

The "OPTION" column in the table below enumerates the allowed values for the "OPTION" variable. Same for the variables E1, ..., E4. Each of the following operations accepts up to 4 arguments, but they may take fewer. Values of "En" left unspecified are ignored.

| <b>OPTION</b> | <b>DESCRIPTION</b>   |
|---------------|--|
| PARTSET       | Trim by PARTSET. E1 is the shell part set ID. E2 is the distance. Elements farther away than the distance are deleted. Please note, only element on one side will be deleted. To delete both sides, repeat the card with FLIP value reversed.  |
| SEGSET        | Trim by SEGMENT SET. E1 is the segment set ID. E2 is the distance. Elements farther away than the distance are deleted. Please note, only element on one side will be deleted. To delete both sides, repeat the card with FLIP value reversed. |
| PLANE         | Trim by PLANE. E1 is the node ID of a node on the plane. E2 is another node ID off the plane. And vector (E2-E1) is normal to the plane.   |
| CYLINDER      | Trim by CYLINDER. E1, E2 are node IDs of the center nodes at two ends. E3, E4 are the radius at those two ends.  |
| BOXCOR        | Trim by BOX. The box is defined using coordinates. E1 is BOX ID. Please refer to *DEFINE_BOX for details on setting up a box in global coordinate system or *DEFINE_BOX_LOCAL in local coordinate system.                                      |
| BOXCPT        | Trim by BOX. The box is defined using S-ALE control points (CPT). E1 is BOX ID. Please refer to *DEFINE_BOX for details on setting up a box.   |
| SPHERE        | Trim by SPHERE. E1 is the node ID of the sphere center node. E2 is the radius of the sphere.   |

**Example:**

1. This example shows how to trim a mesh generated by \*ALE\_STRUCTURED\_MESH card. We use the same mesh in example 1 in the \*ALE\_STRUCTURED\_MESH card. But now we trim the mesh so any element outside of a sphere centered at (0,0,0) with a radius of 0.1 are deleted.

```

*ALE_STRUCTURED_MESH
$  mshid      pid      nbid      ebid
   1          1      200001    200001
$  nptx      npty      nptz      nid0      lcsid
   1001      1001      1001         1         234
*ALE_STRUCTURED_MESH_TRIM
$  mshid      option      oper      flip      nid      radius
   1          SPHERE      oper         1         1         0.10
*NODE
  1  0.0000000e+00  0.0000000e+00  0.0000000e+00
  2  0.0000000e+00  0.0000000e+00  0.0000000e+00
  3  0.1000000e+00  0.0000000e+00  0.0000000e+00
  4  0.0000000e+00  0.1000000e+00  0.0000000e+00

```

\*END

If instead we want to trim the elements inside the sphere, we simply change flip to 1 as follows.

```
*ALE_STRUCTURED_MESH
$  mshid      pid      nbid      ebid
    1          1      200001    200001
$  nptx      npty      nptz      nid0      lcsid
    1001      1001      1001        1        234
*ALE_STRUCTURED_MESH_TRIM
$  mshid      option      oper      flip      nid      radius
    1          SPHERE      oper      1          1        0.10
*END
```

In case we want to trim the elements between two spheres which both centered as (0,0,0) and have a radius of 0.05 and 0.1 respectively. What we could do is to use two cards: first delete all elements inside the 0.1 sphere and then un-delete the elements inside the 0.05 sphere.

```
*ALE_STRUCTURED_MESH
$  mshid      pid      nbid      ebid
    1          1      200001    200001
$  nptx      npty      nptz      nid0      lcsid
    1001      1001      1001        1        234
*ALE_STRUCTURED_MESH_TRIM
$  mshid      option      oper      flip      nid      radius
    1          SPHERE      oper      1          1        0.10
    1          SPHERE      1          1          1        0.00
*END
```

Or we could first delete all the elements outside the 0.05 sphere and then un-delete the elements outside the 0.1 sphere.

```
*ALE_STRUCTURED_MESH
$  mshid      pid      nbid      ebid
    1          1      200001    200001
$  nptx      npty      nptz      nid0      lcsid
    1001      1001      1001        1        234
*ALE_STRUCTURED_MESH_TRIM
$  mshid      option      oper      flip      nid      radius
    1          SPHERE      oper      0          1        0.05
    1          SPHERE      1          0          1        0.10
*END
```

To use BOXCPT, we define a box using S-ALE control point numbers. The example below deletes all elements outside of a box with two endpoints at (8,8,8) and (15,15,15) in S-ALE control points.

```
*ALE_STRUCTURED_MESH
$  mshid      pid      nbid      ebid
    1          1      200001    200001
$  nptx      npty      nptz      nid0      lcsid
    1001      1001      1001        1        234
*ALE_STRUCTURED_MESH_TRIM
$  mshid      option      oper      flip      boxid
    1          BOXCPT      oper      0          1
```

# \*ALE

# \*ALE\_STRUCTURED\_MESH\_TRIM

---

```
*DEFINE_BOX
$   boxid      xmn      xmx      ymn      ymx      zmn      zmx
      1         8        15         8        15         8        15
*END
```