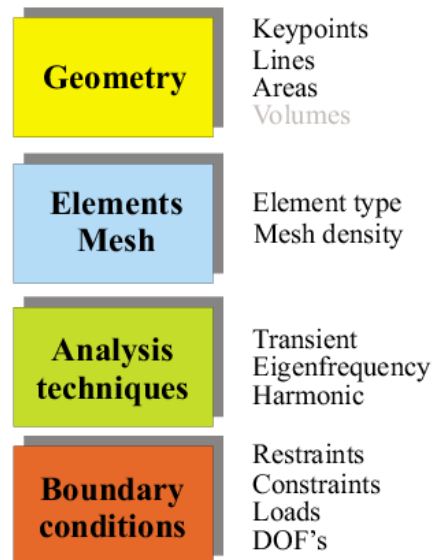


# 1 Basic Steps of Modeling in ANSYS

In finite element simulations, the following items have to be defined. They will be discussed on the next pages.



## 1.1 General Preparative Commands

The starting section of every ANSYS input file has to include some general definitions, like the reset to a well defined status in ANSYS, the setting of the filename, the call of the preprocessor and the initialization of the mesh-interface:

```
! STARTING =====
!
! exits normally from a (maybe running) processor =====
fini

! clearing the database =====
/clear

! definition of output filename =====
/filename,cube,1
```

```

! starting the preprocessor =====
/prep7

! initialize .mesh-interface =====
init

! end of general section =====

```

In order to guarantee reproducibility the geometry is not generated interactively. We are using a special macro language offered by *ANSYS* for the model generation - the so-called *PREP7* command language.

## 1.2 Setup of the Geometry

The setup of the geometry can't be totally decoupled from the mesh generation process. Certain conditions have to be considered during the geometry generation.

### 1.2.1 Keypoints

After the general heading section the batch oriented input script starts with the input of the keypoints, which represent the boundary nodes of each geometric structure.

It is very useful to perform the modeling fully parameter oriented. Therewith, modifications of the model become easily possible. You should start your keypoint generation section by defining appropriate parameters.

In 2D- and axisymmetric modeling all geometries are defined in the x-y-plane and all data have to be entered in SI-units.

```

! parameter definition section
! x coordinates
center_rad    = 0.0e-3
thelem_rad    = 0.4e-3
...
! y coordinates
tip           = 0.0e-3
tip_plate     = 0.5e-3
...
! end of parameter definition section
! keypoint generation
k,,center_rad,thelem_rad
...
k,,thelem_rad,tip_plate
...

```

The keypoints are entered using the **k** command. The points are numbered automatically, but explicit numbering is also possible.

### **K, NPT, X, Y, Z**

- **NPT** Reference number for keypoint. If zero, the lowest available number is assigned.
- **X, Y, Z** Keypoint location in the active coordinate system (may be  $R, \theta, Z$  or  $R, \theta, \Phi$ ).

### **1.2.2 Lines**

The explicit generation of lines is normally not necessary. The according lines are typically generated as a subentity of an area.

But for the generation of complex area structures it is sometimes useful to generate the according lines first.

This can be done by the command

### **L, P1, P2**

with **P1, P2** defining the end keypoints of the line. Additionally, spacing ratio, number of elements, etc. may be defined. Please refer to the ANSYS-Help.

### **1.2.3 Areas**

Areas are generated with the **a**-command followed by a list of keypoints.

### **A,P1,P2,P2,P3,...**

- **P1,P2,...** keypoints

In order to generate areas without self intersection and to guarantee the proper functionality of the FE-program the keypoint list must be defined in counterclockwise direction. Therewith, the normal direction is fixed.

```
! area definition section
a,1,2,6,5      ! copper plate
```

```
a,2,3,7,6          ! copper plate
...
! end of area definition section
```

```

13. .14    15. .16
9. .10     11. .12
5. .6      7. .8
. .        . .
1  2       3  4
```

9	8	5
6	7	4
1	2	3

## 1.3 Entity Selection

Very often, specific entities have to be selected before an operation may be done on these entities.

### 1.3.1 Direct Selection of an Entity

For instance, specific lines have to be selected by the *ANSYS* selection command **LSEL, Type, Item, Comp, VMIN, VMAX, VINCO**

- **Type:** Label identifying the type of selection:
  - S Select a new set (default)
  - R Reselect a set from the current set.
  - A Additionally select a set and extend the current set.
  - U Unselect a set from the current set.
  - ALL Restore the full set.
  - NONE Unselect the full set.
  - INVE Invert the current set (selected becomes unselected and vice versa).
  - STAT Display the current select status.
- **Item** Label identifying data. The most important valid item labels are shown in the table below. Some items also require a component label. Defaults to LINE.
- **Comp** Component of the item (if required). Valid component labels are shown in the table below.

- **VMIN, VMAX, VINC** Minimum, maximum and increment of item range (e.g. startIndex, stopIndex, incIndex with Item=LINE)

Item	Comp	Description
LINE		Line number.
LOC	X,Y,Z	X,Y, or Z center location in the active coordinate system

In a similar way, the command

### **ASEL, Type, Item, Comp,VMIN, VMAX, VINC**

selects the concerning area(s).

**ATTENTION:** For all selection commands with coordinates as item (**loc**), a default tolerance of  $10^{-6}$  is used!! Therefore, always provide the coordinates as “range”: e.g.

*Isel,s,loc,x,xPos-epsilon,xPos+epsilon*

with *epsilon* a sufficient small value  $\leq 10^{-6}$ .

In the same way as **Isel** and **asel**, the command **nsel** may be used for selecting nodes:

### **NSEL, Type, Item, Comp,VMIN, VMAX, VINC.**

## **1.3.2 Selection with the Aid of another Entity**

Another possibility to select an item is the selection with the aid of another item with higher dimension.

The command

### **NSLL, Type, NKEY**

selects those nodes associated with the selected lines. The parameter **Type** is the same as in the previous selection commands.

- **NKEY** specifies whether only interior line nodes are to be selected:
  - 0 Select only nodes interior to selected lines.
  - 1 Select all nodes (interior to line and at keypoints) associated with the selected lines

In the same way operate the commands

## ESLA, Type

which selects the elements of the selected areas and

## LSLA, Type

which selects the lines of the selected areas. Both commands do not need a **NKEY** parameter.

## 1.4 Meshing

### 1.4.1 Selecting Element Types

Before any mesh command, the element type for meshing has to be specified with the **setelems** command. The **setelems** command is a specific command of the *mesh-interface*. Therewith, you can select the element type (2D, 3D element), and the order of the element (linear or quadratic).

The chosen element type is valid until the next **setelems** command.

Syntax:

$$\text{setelems, } \left\{ \begin{array}{l} \text{'2d-line'} \\ \text{'3d-line'} \\ \text{'triangle'} \\ \text{'quadr'} \\ \text{'tetra'} \\ \text{'brick'} \end{array} \right\} [, \text{'quad'}]$$

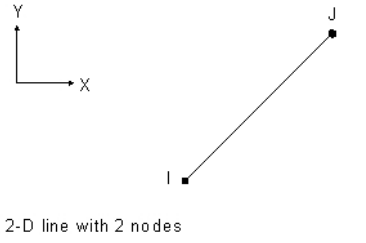
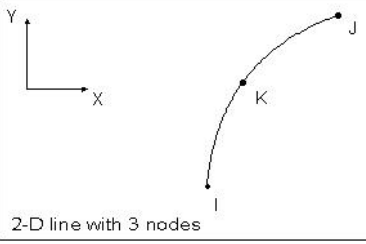
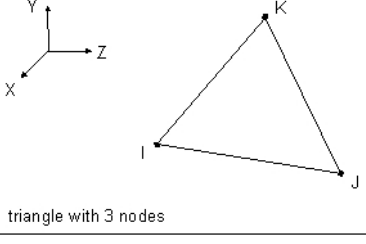
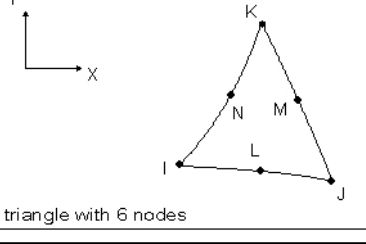
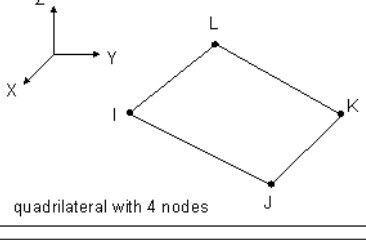
The first argument defines the element type, whereas the optional second argument 'quad' switches to second order elements.

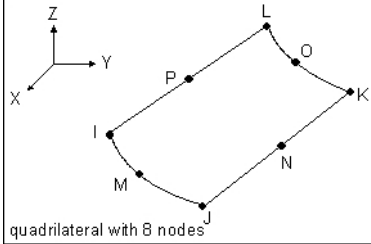
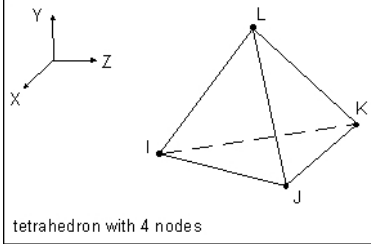
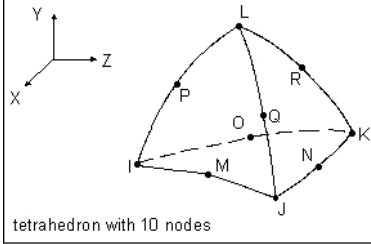
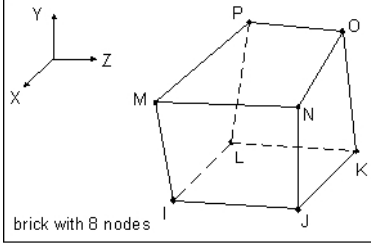
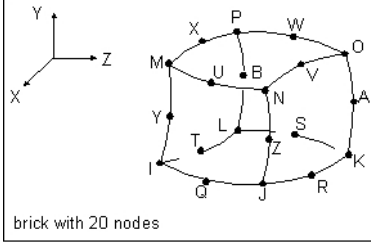
**ATTENTION:** The use of the apostrophes in the command is obligatory!

```
! element type selection =====  
setelems,'quadr','quad'
```

The same results can be achieved by using the **type** command accompanied by the proper element type number of the elements:

The following table shows the mapping between the element setting commands and *element type numbers*:

Element setting command	Element type number	Meshing Facet
<code>setelems,'2d-line'</code>	100	 <p>2-D line with 2 nodes</p>
<code>setelems,'2d-line','quad'</code>	1	 <p>2-D line with 3 nodes</p>
<code>setelems,'triangle'</code>	4	 <p>triangle with 3 nodes</p>
<code>setelems,'triangle','quad'</code>	5	 <p>triangle with 6 nodes</p>
<code>setelems,'quadr'</code>	6	 <p>quadrilateral with 4 nodes</p>

setelems,'quadr','quad'	7	 <p>quadrilateral with 8 nodes</p>
setelems,'tetra'	8	 <p>tetrahedron with 4 nodes</p>
setelems,'tetra','quad'	9	 <p>tetrahedron with 10 nodes</p>
setelems,'brick'	10	 <p>brick with 8 nodes</p>
setelems,'brick','quad'	11	 <p>brick with 20 nodes</p>

### 1.4.2 Setting the Line Element Size

There are various methods for controlled mesh generation. One possibility to define a mesh with varying densities is to set the number of elements at the surrounding lines. After selecting a line, a specific number of sampling points is assigned to this line by the



**lesize** command.

### **LESIZE, NL1, SIZE, ANGSIZ, NDIV, SPACE**

- **NL1** Number of the line to be modified. If **ALL**, all selected lines are modified. A component name may also be substituted for **NL1**.
- **SIZE** If **NDIV** is blank, **SIZE** is the division (element edge) length. The number of divisions is automatically calculated from the line length (rounded upwards to the next integer). If **SIZE** is zero (or blank), use **ANGSIZ** or **NDIV**.
- **ANGSIZ** The division arc (in degrees) spanned by the element edge (except for straight lines, which always results in one division). The number of divisions is automatically calculated from the line length (rounded upwards to the next integer).
- **NDIV** If positive, **NDIV** is the number of element divisions per line.
- **SPACE** Spacing ratio. If positive, nominal ratio of last division size to first division size (if  $> 1.0$ , sizes increase, if  $< 1.0$ , sizes decrease). If negative, **SPACE** is the nominal ratio of center division(s) size to end divisions size. Ratio defaults to 1.0 (uniform spacing).

### **1.4.3 Meshing of Areas**

The command

#### **MSHKEY, KEY**

indicates the type of meshing. **KEY** may be one of

- **0** Use free meshing (the default)
- **1** Use mapped meshing
- **2** Use mapped meshing if possible; otherwise, use free meshing.  
If you specify **MSHKEY,2**, SmartSizing will be inactive even while free meshing non-map-meshable areas.

The command

#### **AMESH, NA1, NA2, NINC**

meshes areas from number **NA1** to **NA2** in steps of **NINC** (defaults to 1). If **NA1 = ALL**, **NA2** and **NINC** are ignored and all selected areas are meshed. A component name may also be substituted for **NA1** (**NA2** and **NINC** are ignored).

For a more detailed control about meshing, further commands are needed.

The very powerful command

### **SMRTSIZE, SIZLVL, FAC, EXPND, TRANS, ...**

specifies meshing parameters for automatic (smart) element sizing.

- **SIZLVL** Overall element size level for meshing. The level value controls the fineness of the mesh. Valid inputs range from 1 (fine mesh) to 10 (coarse mesh).
- **FAC** Scaling factor applied to the computed default mesh sizing
- **EXPND** Mesh expansion (or contraction) factor. EXPND is used to size internal elements in an area based on the size of the elements on the area's boundaries.
- **TRANS** Mesh transition factor. TRANS is used to control how rapidly elements are permitted to change in size from the boundary to the interior of an area.

A further command to influence the meshing strategy is

### **MOPT,LAB,VAL**

- **LAB** Meshing option to be specified (dictates the meaning of Value). Many options are similar to those in **SMARTSIZE**, e.g. EXPAND, TRANS. For further details, see ANSYS-HELP.
- **VAL** Depends on **LAB**.

### **AOVLAP, NA1, NA2, NA3, NA4, NA5, NA6, NA7, NA8, NA9**

generates new areas which encompass the geometry of all the input areas. The new areas are defined by the regions of intersection of the input areas, and by the complementary (non-intersecting) regions.

- **NAx** Number of overlap areas. Instead of NA1, **all** can be used.

This command works not properly, if an area lies entirely inside another one. In this case, the command

### **ASBA, NA1, NA2, SEPO, KEEP1, KEEP2**

must be used, which subtracts areas from areas.

- **NA1, NA2** define the areas to subtract from and which has to be subtracted. Both may be substituted by **ALL**.
- **SEPO** Behavior if the intersection of the areas is a line or lines:

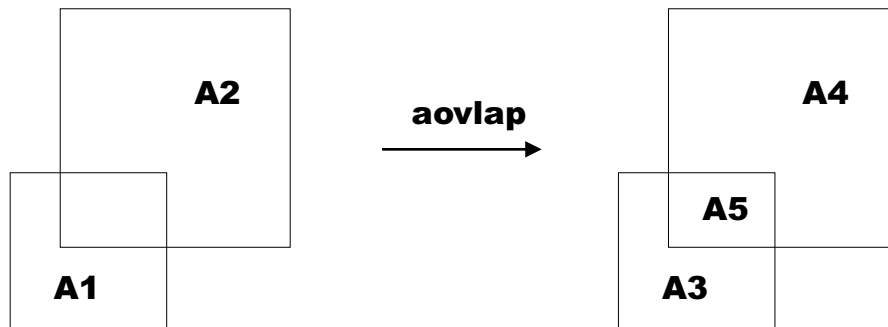


Figure 1.1: Explanation of the **AOVLAP** command

- (blank) The resulting areas will share line(s) where they touch.
- SEPO The resulting areas will have separate, but coincident line(s) where they touch.
- KEEP1 Specifies whether the areas of NA1 are to be deleted:
  - (blank) Use the setting of KEEP on the BOPTN command.
  - DELETE Delete the areas of NA1 after ASBA operation
  - KEEP Keep the areas of NA1 after ASBA operation
- KEEP2 Same as KEEP1, just for areas of NA2.

In connection with the commands **AOVLAP** and **ASBA**, the command

### **ASEL, INVE**

is very important. It inverts the current set (selected items becomes unselected and vice versa). Therewith, it is much easier to do the necessary reselection.

If possible, the whole model should be meshed in one step. Therewith, no duplicate nodes are generated and you need not to specify the number of elements at every line at the intersection of two media.

```
lssel,s,loc,x,center_rad
lssel,r,loc,y,tip,tip_plate
lesize,all,,20
amesh,all
```

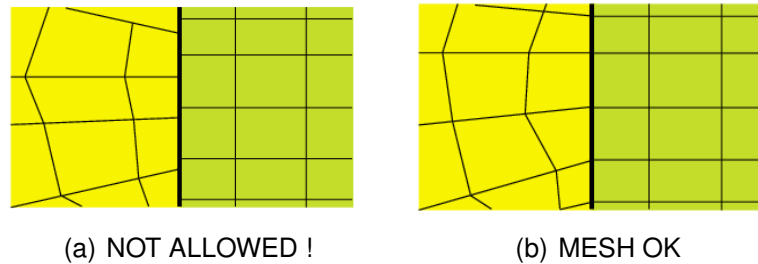


Figure 1.2: Hanging Nodes: A typical problem of mesh generation

Due to the fact that the whole area is meshed at once there is also only one element type assigned to this mesh. Therefore, we have to modify the elements in order to distinguish between different element or material regions.

The command

### **EMODIF,ALL**

assigns the previously defined element type to all selected elements:

```
asel,s,,,7,10
esla
type,2
emodif,all
```

The elements to be modified can be easily selected via the associated areas using the **asel** command. The elements within this region are then selected by an **esla** command. With the **type** command the element group that should be assigned to this region is selected and by the **emodif** command this element group is assigned to the selected elements.

**NUMMRG,NODE**  
**NUMMRG,ELEM**  
**NUMCMP,NODE**  
**NUMCMP,ELEM**

In order to avoid hanging nodes, the **NUMMRG** and **NUMCMP** commands may be applied for the case of individual area meshing. The **NUMMRG** command merges nodes, elements, keypoints with identical coordinates. Due to the merging of nodes and elements a inconsistent numbering occurs. By applying a **NUMCMP** command, the numbering is compacted, which is necessary for CFS++.

Additionally, to check which elements violate element shape limits you can use

**CHECK,esel,warn**

Please read the ANSYS-HELP for more details.

## 1.5 Using Components

A nice way of modeling in ANSYS is to define components. The command

**CM, Cname, Entity**

groups geometry items into a component. The argument **Cname** is a self defined name and may be chosen arbitrarily, **Entity** identifies the type of geometry and must be one of

- **VOLU** Volumes
- **AREA** Areas
- **LINE** Lines
- **KP** Keypoints
- **ELEM** Elements
- **NODE** Nodes

Due to the fact that you have to define the geometrical type while assigning a component name, you have to pay attention by reusing this component with the command

**CMSEL, Type, Name**

which selects a subset of components and assemblies: You always get only the geometric entities you have selected (and defined) first!

- **Type** may be one of the selection parameters (**s,r,a,u,all,none**) - see the section *Entity Selection* in chapter 1.3) and
- **Name** is the name you have assigned with a previous **cm** command.

Therefore: The easiest way to work with component names is to assign names to the appropriate areas and select the elements of that area by

**esla**

which returns all elements of the selected area.

## 1.6 Coloring the Model

To get a good overlook over your model, always color it. Areas belonging together shall have - consequently - the same color!

### **/COLOR, Lab, Color, N1, N2, NINC**

- **Lab** Defines a geometric item (similar to the cmsel command above). For this command, there are much more parameters, but you normally won't need them...
- **Color** selected color
- **N1, N2, NINC** Apply color to Lab items numbered N1 to N2 (defaults to N1) in steps of NINC (defaults to 1). If N1 is blank, apply color to entire selected range. If Lab is CM, use component name for N1 and ignore N2 and NINC.

As color, one of

BLAC (0)	Black
MRED (1)	Magenta-Red
MAGE (2)	Magenta
BMAG (3)	Blue-Magenta
BLUE (4)	Blue
CBLU (5)	Cyan-Blue
CYAN (6)	Cyan
GCYA ((7)	Green-Cyan
GREE (8)	Green
YGRE (9)	Yellow-Green
Yell (10)	Yellow
ORAN (11)	Orange
RED (12)	Red
DGRA (13)	Dark Gray
LGRA (14)	Light Gray
WHIT (15)	White

may be used.

## 1.7 Writing the Mesh-File

Up to now, we are able to establish a mesh in our preprocessor ANSYS. To be able to do simulations with this mesh, we need an interface to our finite-element kernel. The easiest way (and the fastest, if parameter studies have to be done) is the use of a

mesh-file.

This file is written by the

### **mkmesh**

command. Before you are able to establish this file, several steps have to be performed:

- Initialize the mesh-extension with the **init** command.
- Select the element type(s) with the **setelems** command.
- Mesh the different components of the model.
- Select the appropriate elements and write them with the desired level attribute (**welems** command).
- Select the appropriate nodes and write them with the desired level attribute (**wnodbc** command).
- Select all nodes belonging to the model and write them with the **wnodes** command.
- Select the appropriate nodes, for which results should be saved and write them with the desired level attribute with the **wsavnod** command.
- Select the appropriate elements, for which results should be saved and write them with the desired level attribute with the **wsavelem** command.
- Write the .mesh file with the **mkmesh** command.

The commands **welems**, **wnodbc**, **wsavnod** and **wsavelem** need an additional parameter: a name (enclosed in apostrophes). An examples is given below:

```
! select nodes at y=upper and write nodal boundary conditions with
! level 'upper'
nsel,s,loc,y,upper-eps,upper+eps
wnodbc,'upper'
```

Below you can find all commands needed for writing the mesh-file (as a summary). Please keep in mind: These are not standard commands from ANSYS!

MESHFILE COMMANDS	DESCRIPTION
<b>init</b>	Initializes the mesh-extension
<b>setelems</b> , <i>type</i> ,[ <i>order</i> ]	Select element type for meshing
<b>welems</b> , <i>name</i>	Write selected elements with name attribute <i>name</i>
<b>wnodes</b>	Write selected nodes
<b>wnodbc</b> , <i>name</i>	Write selected nodal boundary conditions with name attribute <i>name</i>
<b>wsavnod</b> , <i>name</i>	Write selected nodes for saving nodal results with attribute <i>name</i>
<b>wsavelem</b> , <i>name</i>	Write selected elements for saving element results with attribute <i>name</i>
<b>mkmesh</b>	Write .mesh file