The .D3O file format

The .D3O exchange file format specifications

Castalia srl
Via Pinturicchio, 24 - 20133 Milan Italy
www.castaliaweb.com
www.steelchecks.com
staff@castaliaweb.com

revision 7.0 April, 19, 2021

1 Introduction

The .D3O exchange file format wishes to be an easy format to be used in order to transfer component information to and from CSE software program, and to and from other CAD/CAE software programs.

The aim of the file is to fully describe the geometry of connections of steel members. So, the file describes a list of MEMBERS, a list of CLEATS (plates, angles, cross-section trunks, et cetera), a list of WELDLAYOUTS and a list of BOLTLAYOUTS.

.D3O file is a simple ASCII file. It uses fixed internal units, Newton, millimeters, °C and derived units.

The objects are placed in space according to how they are modeled in CSE or in an external program. The object position in space is the key criterion to identify and attribute components to "Renodes" within CSE, and to extract them from Renodes, in CSE.

There are several ways CSE can create a .D3O file: it may refer to all the components of a structure (members, cleats, weldlayouts and boltlayouts) or it may refer to the components of a single Renode, and so on. This is more specifically explained in CSE guide.

What is important, however, is that the .D3O file will embed a list of objects of the kind specified, and each program using .D3O file will be able to re-create and place correctly these components in space, with their proper sizes, position and orientation.

.D3O file also manages work processes like bevels, cuts, trimmings, face rotations and so on.

The .D3O file has been designed to allow easy data exchange to and from CSE and from and to external CAD CAE programs.

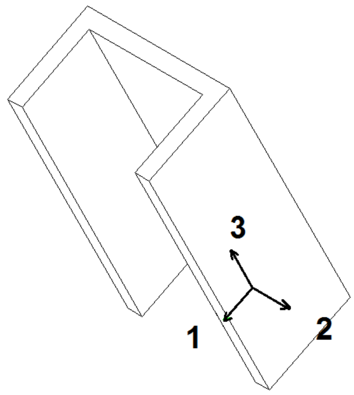
The .D3O file does only list geometrical information and material information. It does not list the choices made about the checks to be done, the bolt working mode, or more generally all the typical information which are set inside CSE to finely tune the checks.

The idea, is that by using an external program the geometry of the connections may be defined and then exported to CSE in a seamless way. Also, using CSE, once the checks are done, by using .D3O it is possible to directly send the details of the geometry of all the connections (or of part of the connections) to a CAD CAE program.

In the following sections, some specific issues will be addressed as preliminary tools to deal with .D3O file.

2 Orientation of components

Components orientation is defined via a set of three local object axes, axes 1, 2, and 3. They use right hand rule. A matrix of nine doubles must be defined.



v1	unit module vector pointing in axis 1 direction	(coo[0], coo[1], coo[2])
v2	unit module vector pointing in axis 2 direction	(coo[3], coo[4], coo[5])
v3	unit module vector pointing in axis 3 direction	(coo[6], coo[7], coo[8])

3 Placement of components

Components are defined into their local cs. Also their work processes are defined into their local cs. Once the object is ready, that is it has been created locally with all its work processes, it is put in place by using the insertion point and the orientation.

The insertion point is generally defined by a couple of 3D points, a point properly said and a 3D shift from that point. So the origin of local cs is made coincident with 3D point:

(point + mv)

The object is then rigidly rotated so that its local axes coincide with the *orientation* specified for that object.

4 Components

4.1 Members

Members are 3D objects got extruding a cross-sections, they map members in true structure.

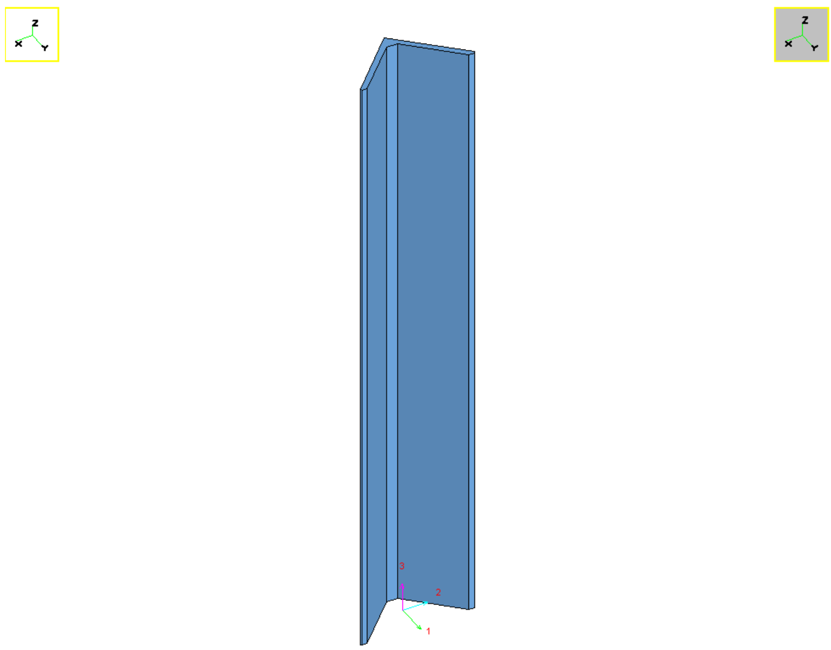
Members in the global 3D model (the *scene*) are defined by the following data:

- Insertion point, usually the first extremity position (before elongation or shortening is applied). However, shifts can be applied relative to original FEM model member position. If this holds true, then insertion point is no longer the first extremity original position.

- Possible shift from insertion point (usually is a null vector)
- Orientation vectors (**v1**, **v2**, **v3**). **v1** and **v2** are the principal axes of the cross section at first extremity. **v3** is the member axis, from 1st to 2nd extremity.
- Original first extremity position (keeps into account fem model offset if any), a 3D point.
- Original second extremity position, a 3D point (keeps into account fem model offset if any).
- Cross-section number, identifying cross-section in the cross-section array. If the member is tapered, two cross-sections numbers are needed.
- Elongation (positive) or shortening (negative), at first and second extremity. A member can join a RENODE in its first or second extremity.
- Material number
- Possible work processes applied to the member.

The member is referred to a coordinate system (1,2,3) where 3 is the axis going from first to second extremity, and 1 and 2 are the cross-section principal axes. Globally the origin of (1,2,3) is placed in insertion point. The insertion point is defined by a 3D point and a 3D shift

point+mv



A member in the scene with its cs (coordinate system)

Locally the origin of the member is placed in its first-extremity un-cut un-modified cross section centroid, no matter if the joint is at first or second extremity of member.

4.2 *Bolt Layouts*

Bolt Layouts are sets of identical bolts, all drilling the same objects or sub-objects (e.g. composed members are made by two or more sub components). The current model used is that the bolts drill a number of parallel faces, up to 10 different thicknesses. If air strata exist, these can also be managed. However, it is necessary to describe the "thickness" of the air strata between the bolted thicknesses.

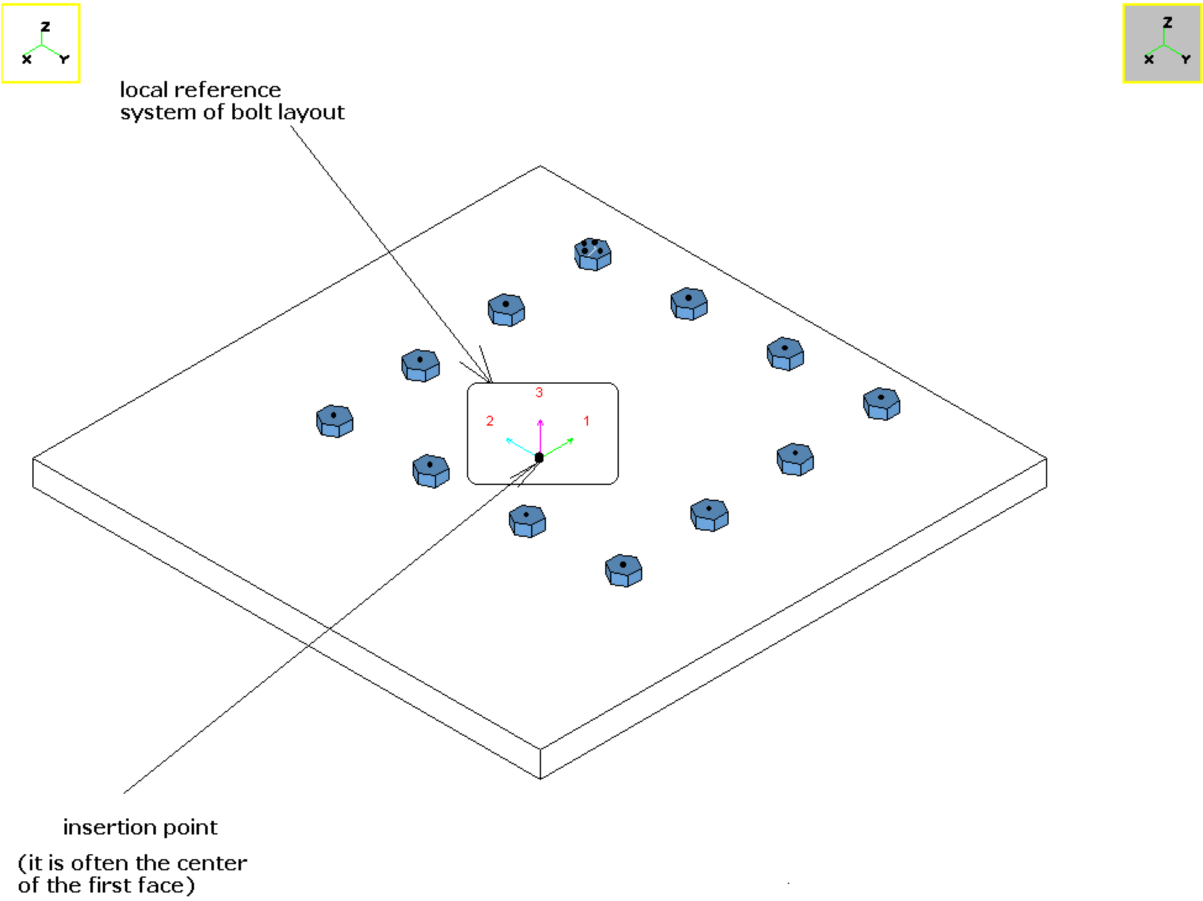
Bolt layouts are placed using two different cs (coordinate systems).

The first (1,2,3) has its origin in the insertion point. The insertion point is placed in:

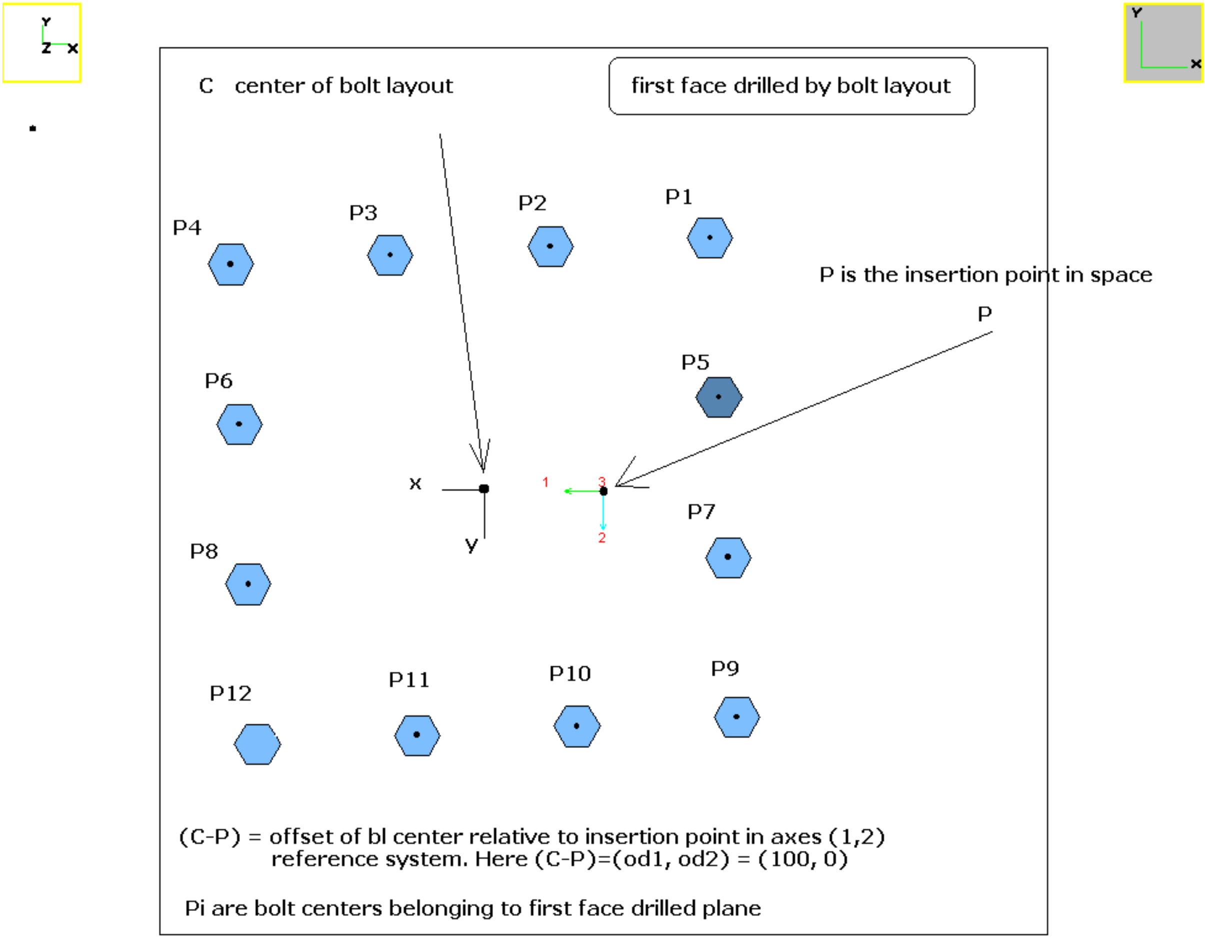
point + mv

a 3D point + a 3D shift.

The insertion point must lay over the first face drilled by bolts. Axis 3 is perpendicular to the first face, and so is parallel to first face normal.

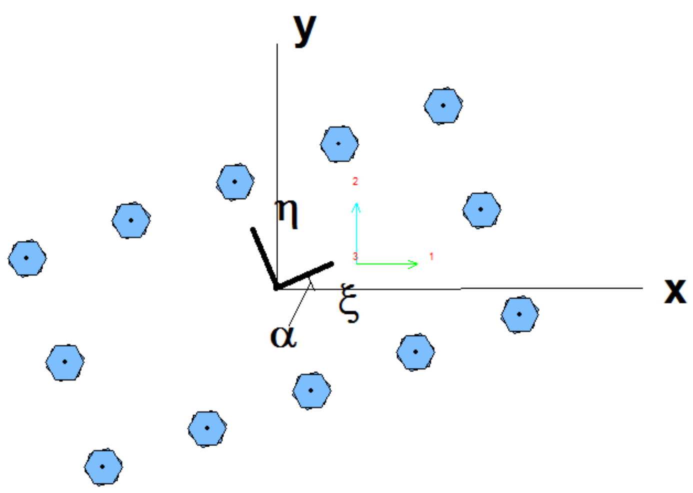


Axis 2 and 1 are free, usually axis 1 is parallel to one of the side of the first face drilled.



The boltlayout centroid is the simple average of the bolt positions, considering the centers of all the bolt holes, over the first drilling plane.

The second cs uses axes (x,y). Its origin is in the bolt layout centroid, over the first drilled plane. Axis x is parallel to axis 1 and axis y is parallel to axis 2. (x,y) is placed in the bolt layout centroid over the first drilling plane. The center can be offset from the insertion point (point+mv) by an offset (od1, od2), i.e. by a planar shift over the first drilled face plane. The user clicks the first drilled face and initially the BL center is in the center of the face clicked. Then by using proper controls, the user displaces the BL over the clicked face so as to position it properly.



A third cs (ξ, η) is that using direction of rows and columns of a regular grid. This cs may be useful to specify boltlayouts which are regular grids, but when rows and columns are not oriented as (x, y) axes (i.e. parallel to 1 and 2 axes). A rotation angle a can be specified. The angle α is the angle that the rows of the grid form with axis x.

A fourth cs is here only cited, as it is used only during computations. It is the principal axes cs (u,v, w) i.e. the set of the three principal axes of the BL. This cs has its origin in BL centroid.

Once defined position, mv,and offset (od1, od2), the position of the bolts is referred to cs (x,y), i.e. it is a vector of 2D points which give the position of single bolts relative to centroid, in (x,y) c.s..

The bolt layout can be of the following types:

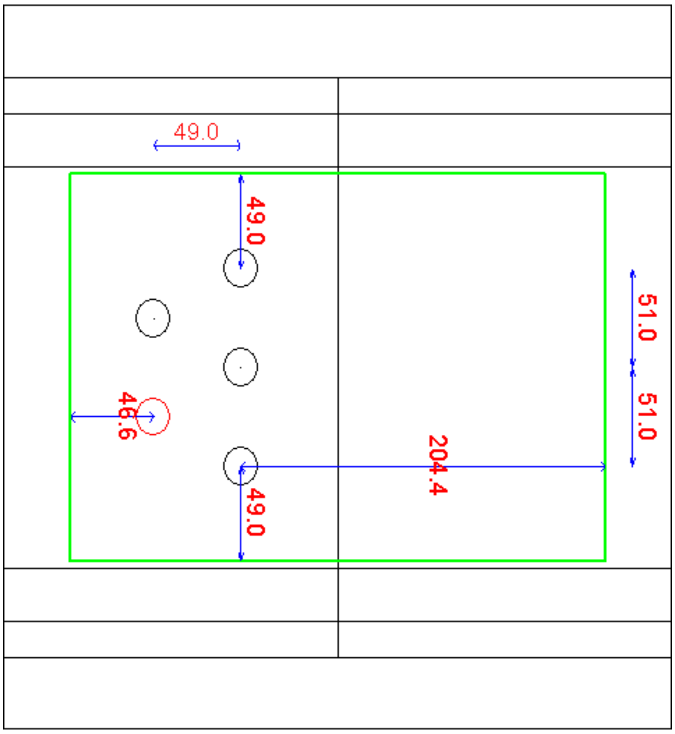
GRID

Here the set of points is reconstructed by using the following parameters:

- Number of rows
- Number of columns
- Distance between rows
- Distance between columns
- Angle of rows relative to axis x, positive if counterclockwise

Grid layouts can be made "empty inside" thus leaving only the perimeter. This is of course meaningful only if nrows > 2 or ncols > 2.

GRID_OFFESET1



Bolts are staggered.

CIRCULAR

Here the parameters to be given are:

- The number of circles (number of "rows");
- The number of bolts over a circle (number of "columns");
- The difference in circles radii (distance between rows)
- The distance between the bolts in the first, smaller circle ("distance between columns").

FREESET

Here the vector of bolt point must be explicitly given in (x, y) cs.

Summing up the following axes can be defined:

1,2,3

x1,x2,x3 — Local 3D axes of BOLT LAYOUT OBJECTS. Origin is insertion point, usually the center of the first drilled face.

x, y

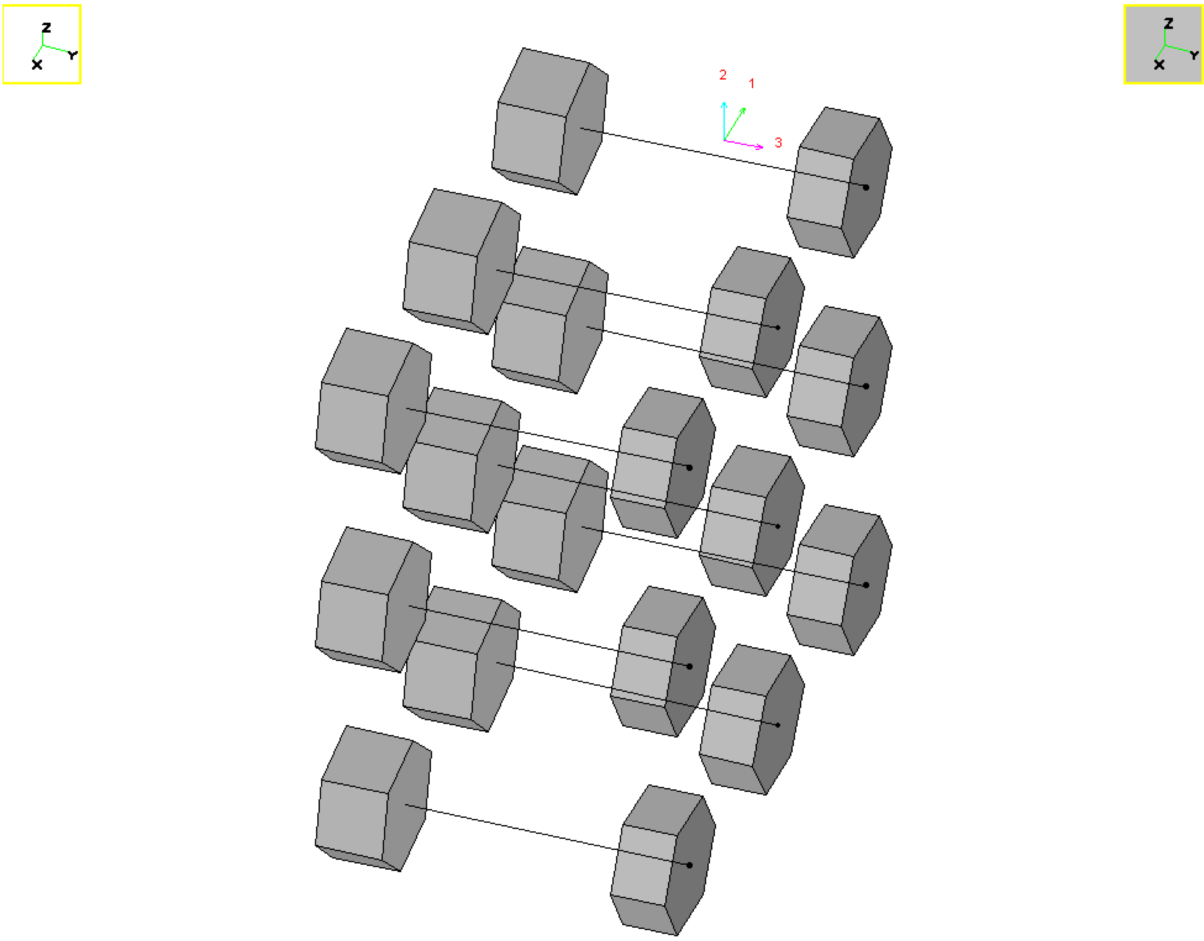
x, y — Local 2D axes of bolts relative to bolt centroid. Origin is bolt center, over the first drilled face. Axis “x” is parallel to axis “1”. Axis “y” is parallel to axis 2.

u,v,w

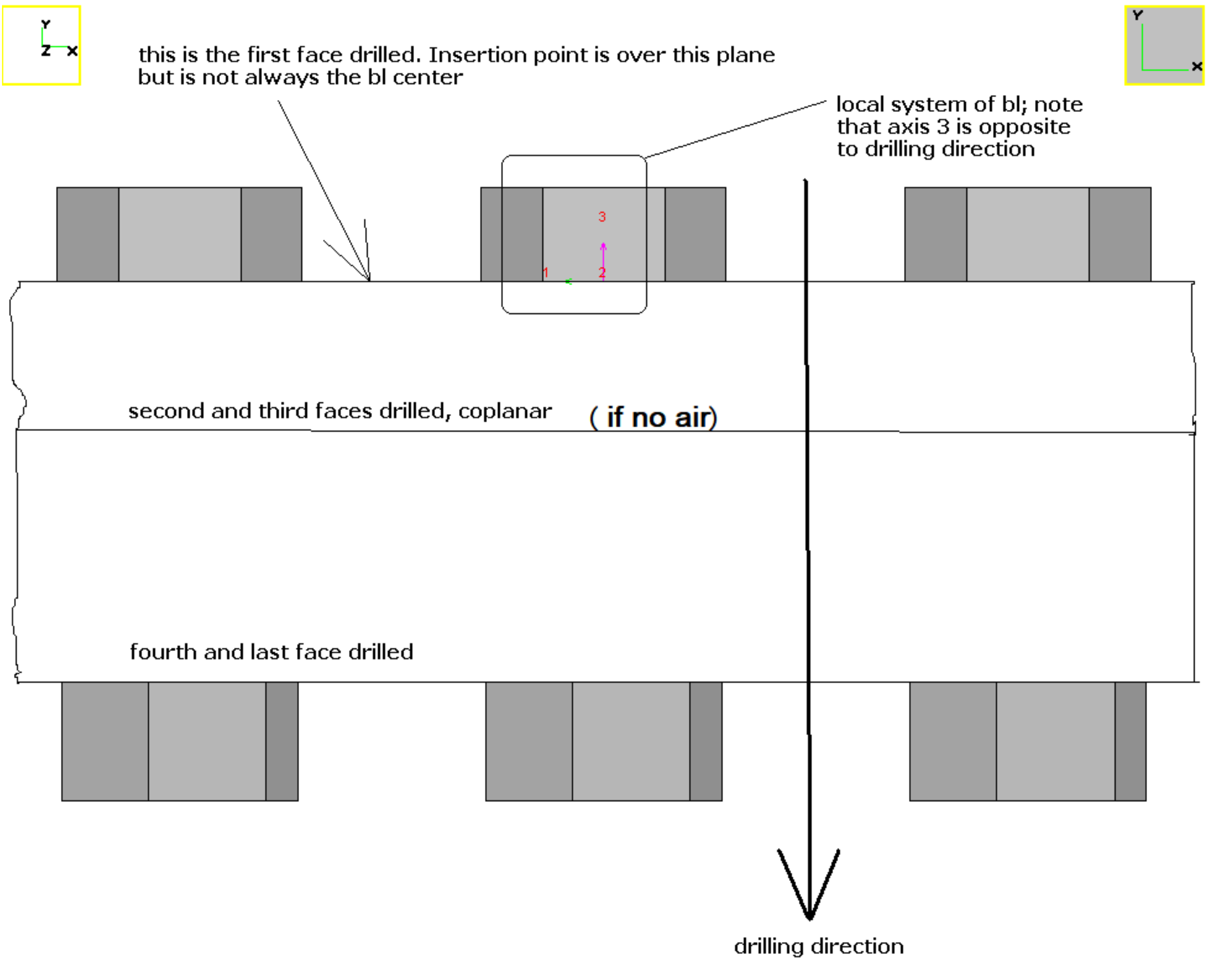
u,v,w — Principal axes of bolt layout. Origin is in bolt layout center. The angle formed by axis “u” to axis “x” is angle β.Axis w is always parallel to axis “3”.

ξ, η

ξ, η — Grid axis for regular bolt layout. Origin is in bolt layout center. The angle formed by “ξ” axis to “η” axis is α.



A BL viewed in 3d view



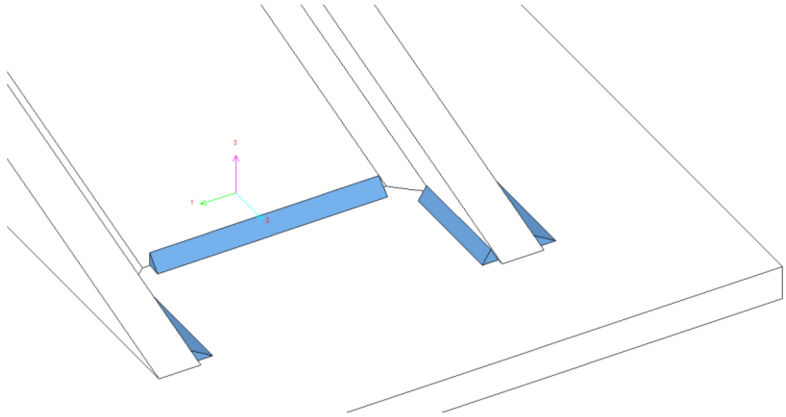
The same BL of previous image in front view

4.3 *Weld Layouts*

Weld layouts are sets of weld seams all welding the same two objects or sub-objects. Weld layouts need two coplanar faces, having opposite normal vector. Usually, one face is contained inside the other, but they can also overlap.

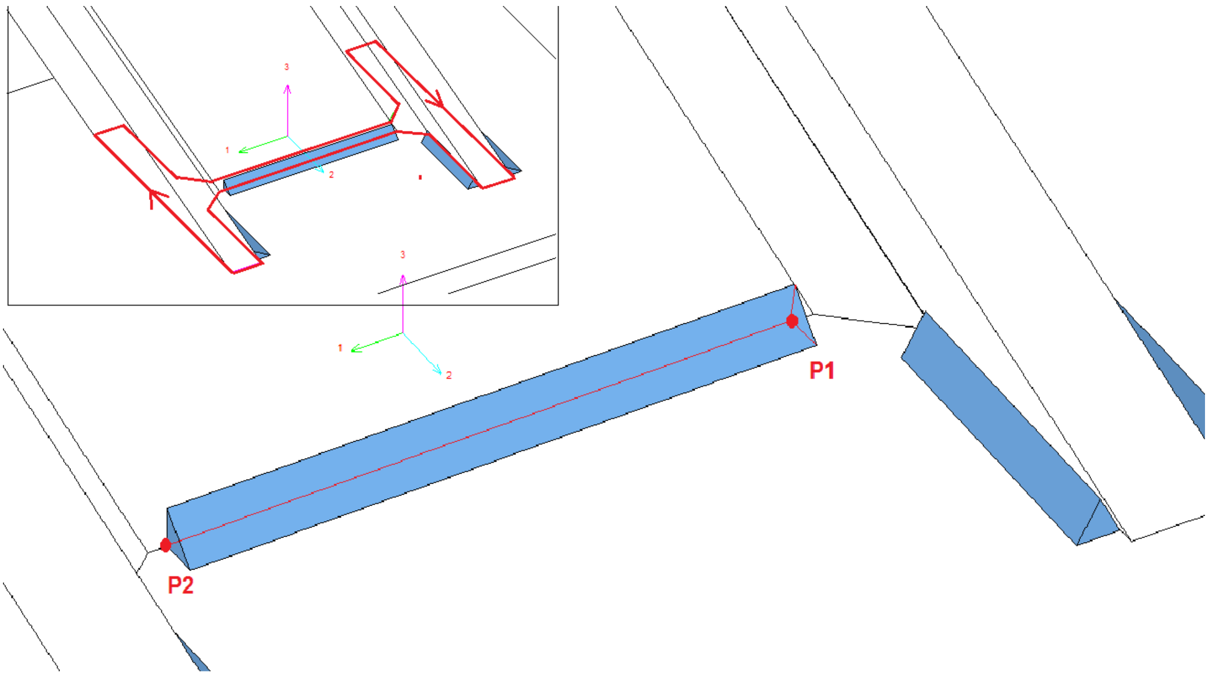
Weld layouts can be fillet weld layouts or penetration weld layouts. All weld seams in a weld layout are of the same kind.

The weld layout is positioned and oriented thanks to a point P in 3D global space, a move from that point, and an orientation matrix like that of any other component. This defines a local origin and a local set of axes (1, 2, 3). Axis 3 is always normal to the two coplanar faces. Axis 1 and 2 lay over the two coplanar faces.



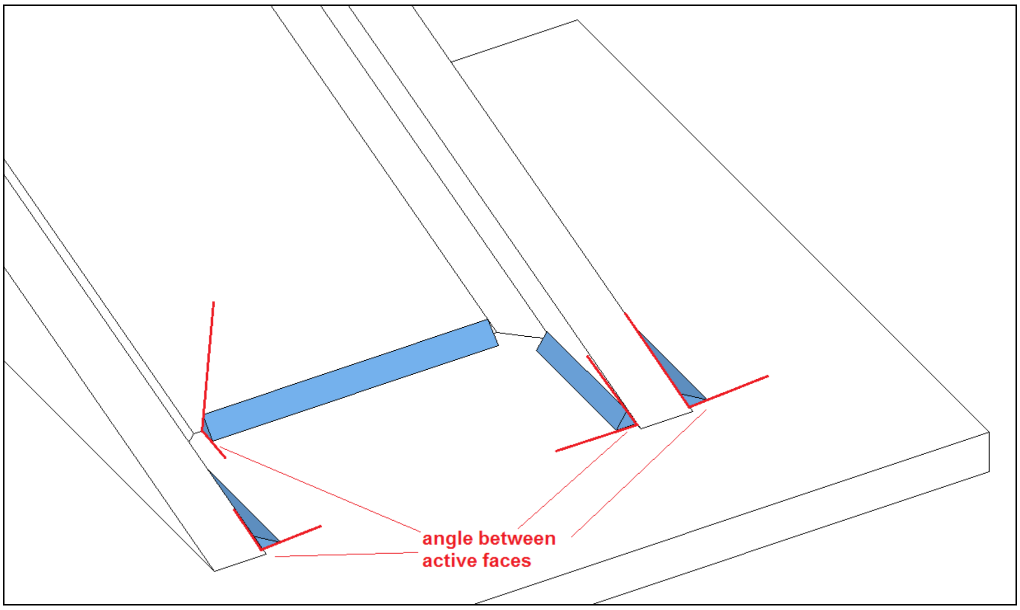
Local axes for a weld layout

The position of each weld seam is set by two points, P1, and P2, i.e. the extremities of each weld (see picture below).

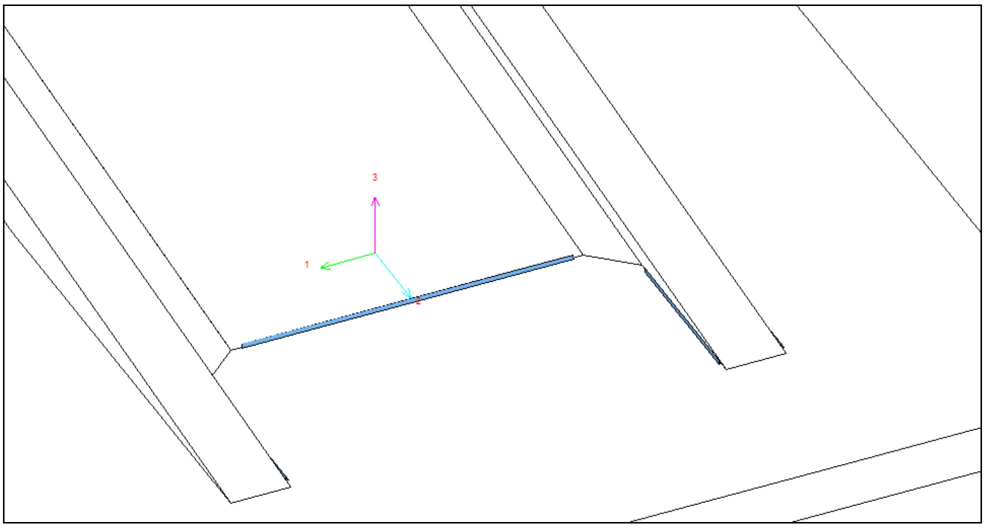


The sequence of the two points is very important. The weld seam points P1-P2 must lay over the perimeter of one of the two coplanar faces. The sides of these faces are ordered in such a way that circulation is counter clockwise when seen from outward normal of the face, belonging to a given object. The two points of each weld must follow the same order of the sides of the receiving face.

Another important concept is that of "angle between active faces". A weld seam in a fillet weld has three longitudinal faces, and two transverse (triangular) end faces. One of the three longitudinal faces is in contact with air, and it is not active. The other two faces are in contact with object welded, A and B. These latter two faces are named "active faces". The angle between the active faces is usually 90°. However it may be higher or lower. Angle between 120° and 60° are normal, higher or lower angles are considered not standard and a warning message is issued.



When defining penetration welds, the same rule apply. The program in fact needs this information to correctly draw the welds and to correctly detect connection.



The same weld layout but as penetration weld. A conventional display is used to let the user pick the welds by mouse.

The extremities P1 and P2 of each weld seam may be different by the extremities of the side to which the seam belong. On a side, more than one seam can be placed. A weld seam segment PI-P2 must be wholly contained in a side of the receiving face.

In fillet welds the "thickness of the weld" *t* is related to the side of the weld *s* by the following relationship:

$$s = \frac{t}{\sin(\alpha)}$$

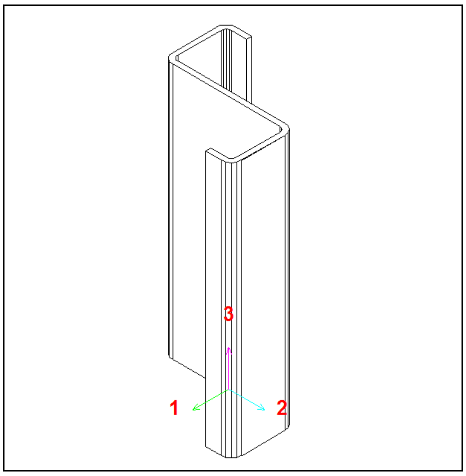
where α is the angle between active faces. In penetration welds, *t* is the thickness welded. In full penetration welds of a thickness *u* there are two different possible choices: adding two welds having thickness *u*/2, or adding just one with thickness *u*.

In fillet welds, the throat size *a* is got by the following rule:

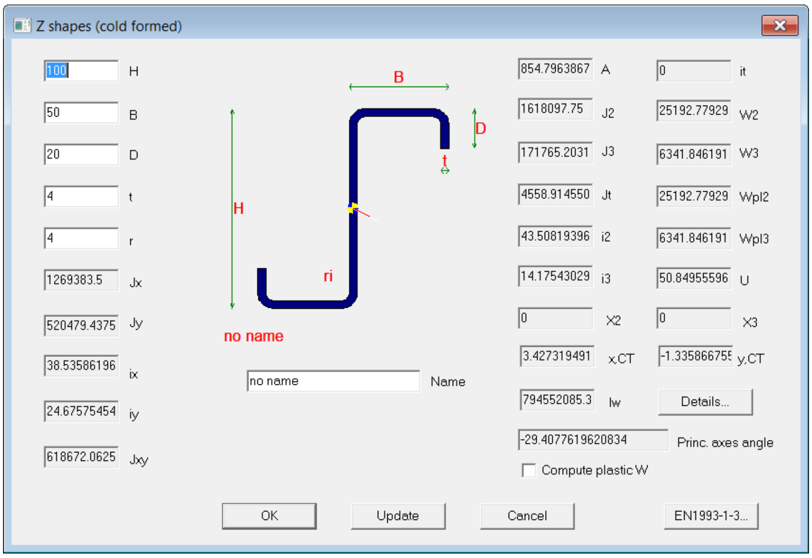
$$a = \frac{t \sin\left(\frac{\pi - \alpha}{2}\right)}{\sin(\pi - \alpha)}$$

4.4 Cross section trunks

Cross-section trunks are components got by extruding a cross-section. Unlike members, they do not use as local axes the principal axes of the cross section, but usually the "contruction axes". Like members, the origin of local system (1, 2, 3) is placed at cross-section centroid. Local axis 3 is the axis of extrusion.



A cross-section trunk in space with its local coordinate system



The cross-section properties. First principal axis of cross-section is marked.

The insertion point is the centroid of the first extremity.

Cross-section trunks can receive work-processes like any other component.

Cross-section trunks must have a material assigned.

Simple angles cannot be added as cross-section trunks, they have a specific object related (see below).

4.5 Plates

A plate in CSE is a plane surface extruded by a thickness in the direction normal to the surface itself. There are a number of special plates, resembling special 2D simple shapes.

Each of these special plates has a code, which identifies the shape itself.

Moreover, a generic plate can be specified by enumerating the corners of a closed polygon. This is a generic plate.

All plates have a thickness. The origin of the local cs (1, 2, 3) depends on the type of the plate. The most part of the plates are fully identified by a code and a vector of 10 parameters, with the sizes. If the number of parameters needed to define the shape of the surface is lower than 10, the extra parameters are unused.

In the following subsections, all the plate code and shapes will be listed, explaining for each plate the parameters needed, and the position of the origin of the local cs.

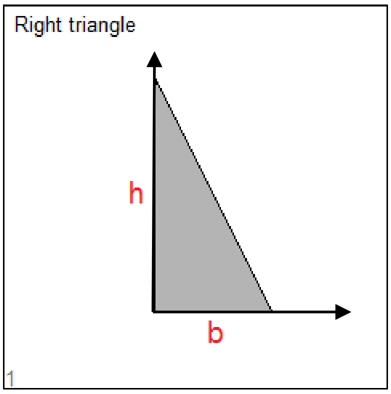
4.5.1 Typical plates

In the following pictures, typical plates are listed. In the bottom left corner of the picture is displayed the code of each typical plate.

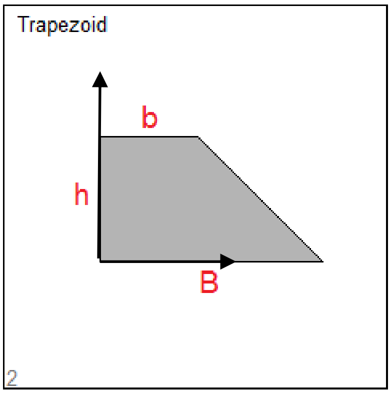
The sequence of the parameters is in the picture caption. So, when defining a typical plate in .D3O file using ten parameters p1, p2, p3…p10, one can assign the correct meaning to each one. Parameters not listed are useless and dummy. So for plate type 1 (right triangle), only two parameters are needed, and the others are dummy.

In the following:

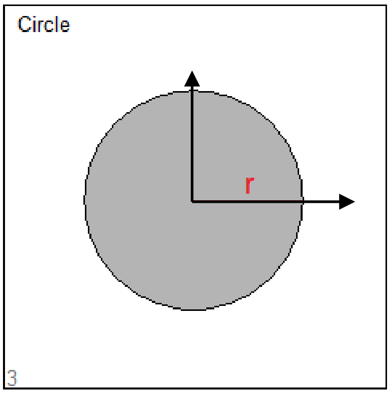
- Two axes are displayed, one horizontal and one vertical.
- Local axis 1 is horizontal.
- Local axis 2 is vertical.
- Origin is the insertion point of the plate.
- Thickness is extruded in $+x_3$ direction (positive axis 3).



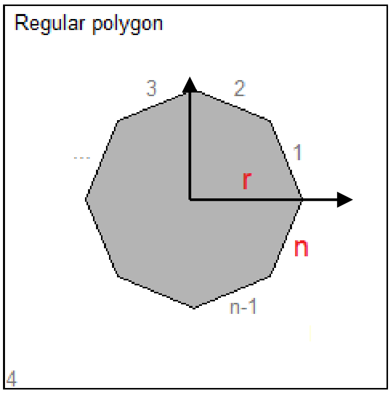
p1=b; p2=h;



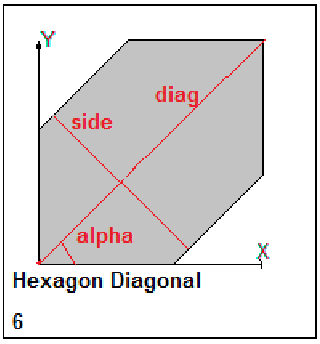
p1=b ; p2= B; p3=h;



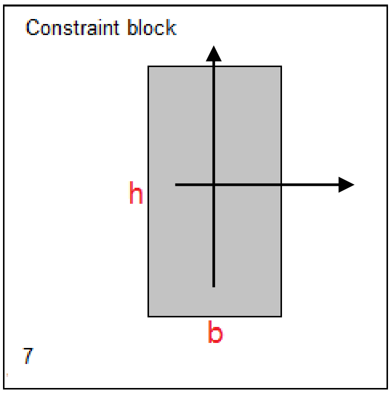
p1= r;



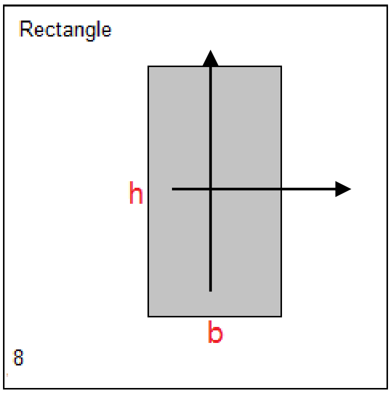
p1= r; p2= n;



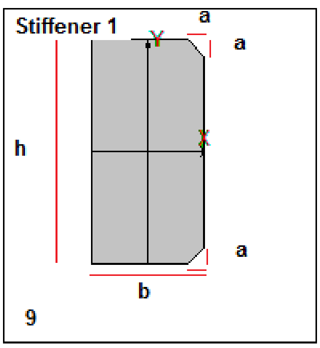
p1=alpha, p2=side; p3=diag



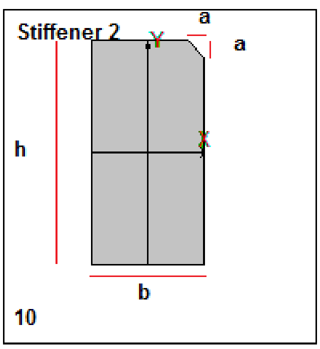
p1= b; p2= h;



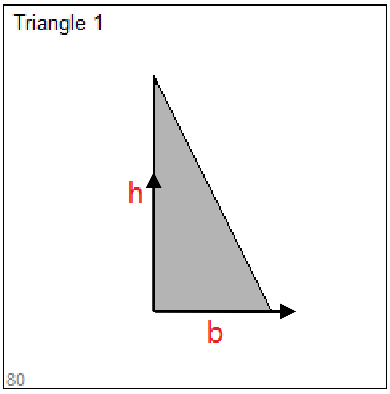
p1= b; p2= h



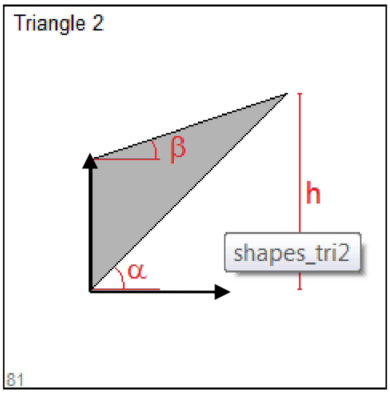
p1=b; p2=h; p3=a



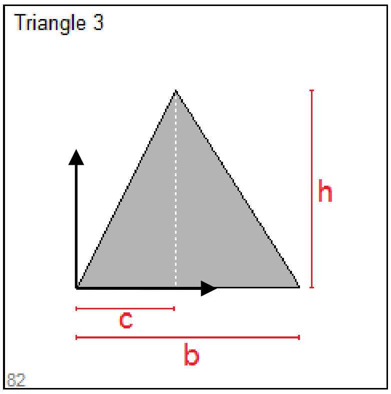
p1=b; p2=h; p3=a



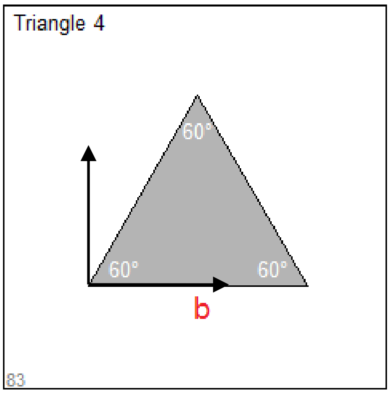
p1= b; p2= h;



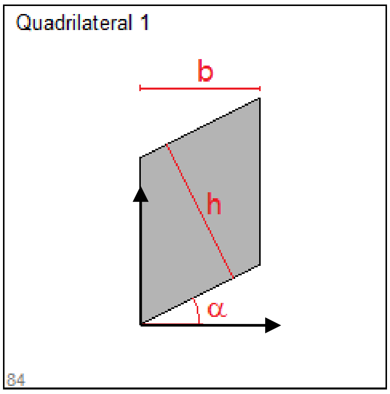
p1=h ; p2= alpha; p3= beta;



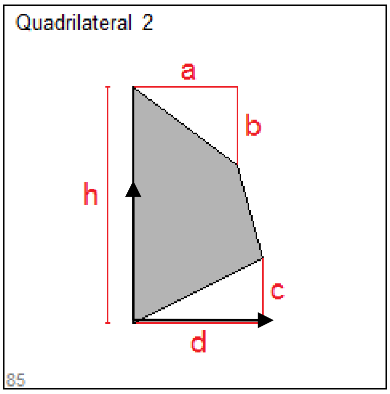
p1=b ; p2=h ; p3= c;



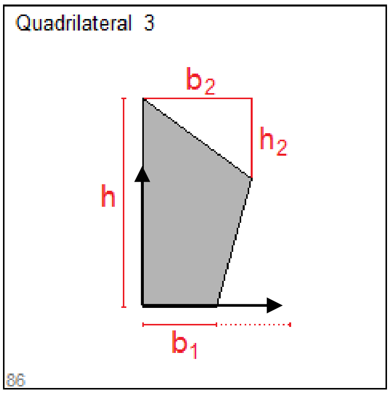
p1= b;



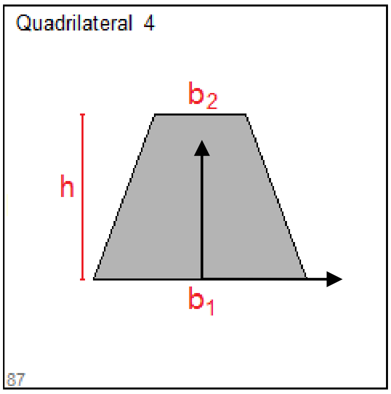
p1= b; p2= h; p3= alpha;



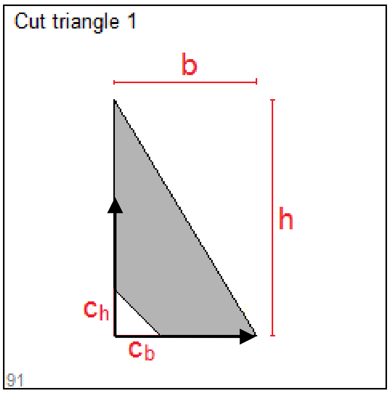
p1=h ; p2= a; p3= b; p4= c; p5= d;



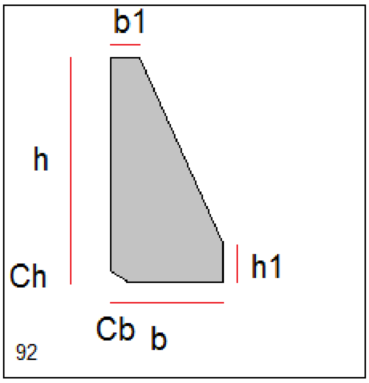
p1=b1 ; p2= b2; p3= h; p4= h2;



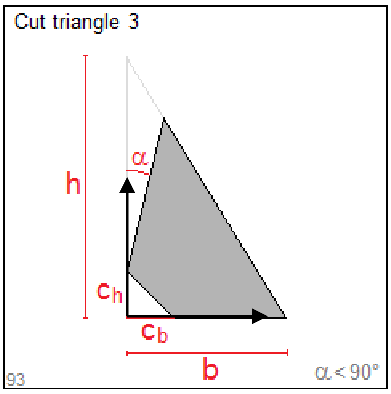
p1= b1; p2= b2; p3= h;



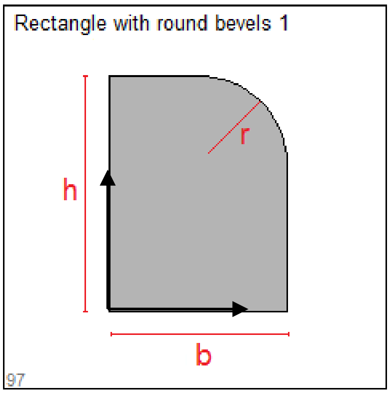
p1= b; p2= h; p3= cb; p4= ch;



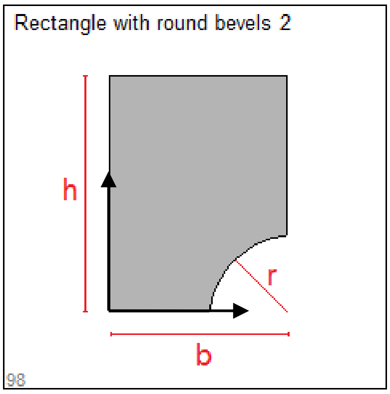
p1=b;p2=h; p3=b1;p4=h1;p5=cb;p6=ch



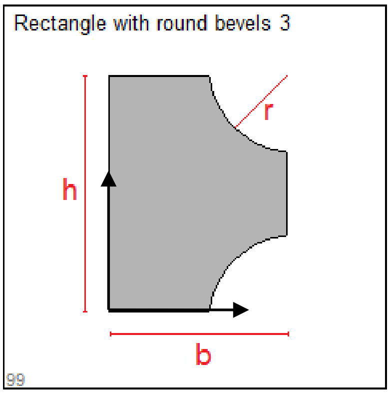
p1= b; p2= h; p3= cb; p4= ch; p5= alpha;



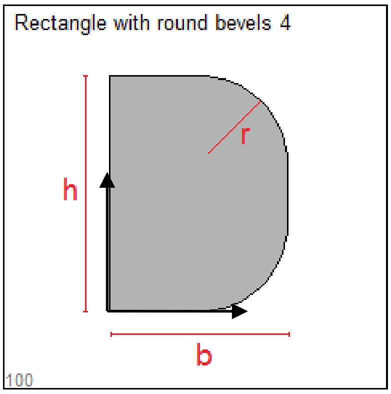
p1=b ; p2= h; p3= r;



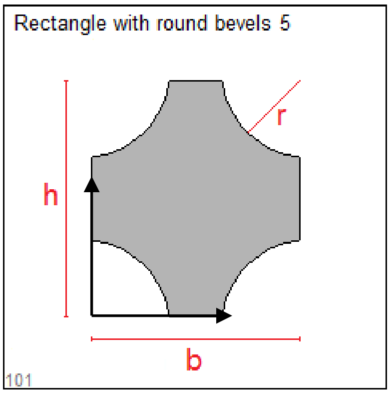
p1= b; p2= h; p3= r;



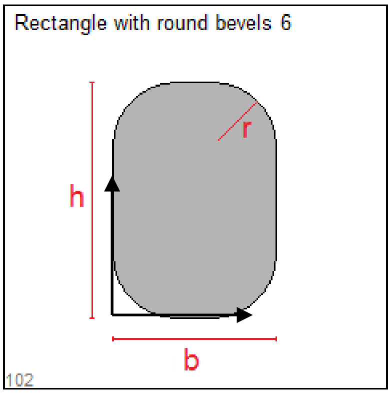
p1= b; p2= h; p3= r;



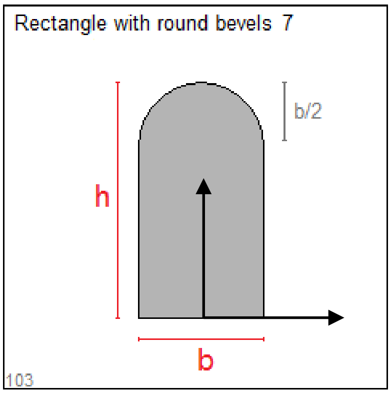
p1= b; p2= h; p3= r;



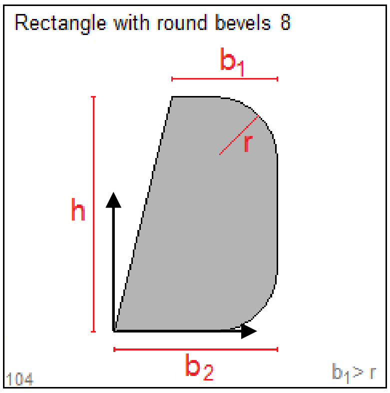
p1=b ; p2= h; p3= r;



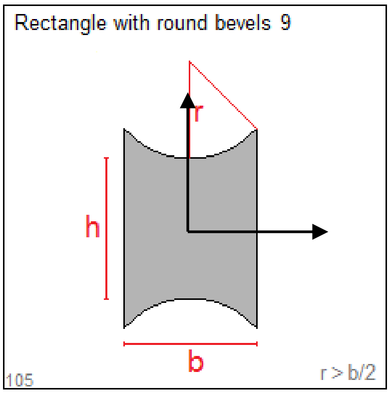
p1= b; p2= h; p3= r;



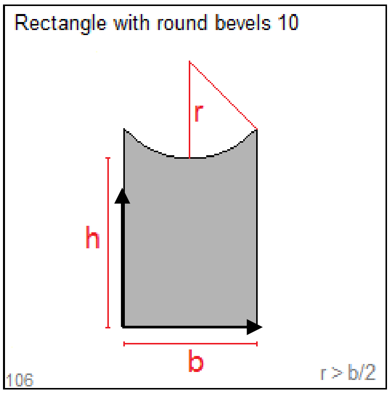
p1= b; p2= h;



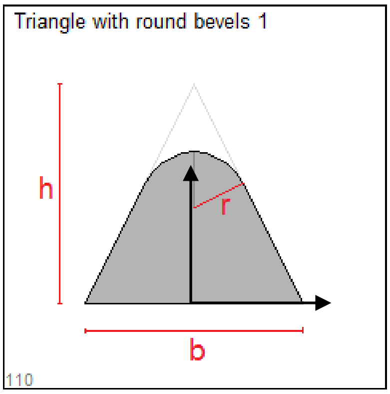
p1= b1; p2= b2; p3= h; p4= r;



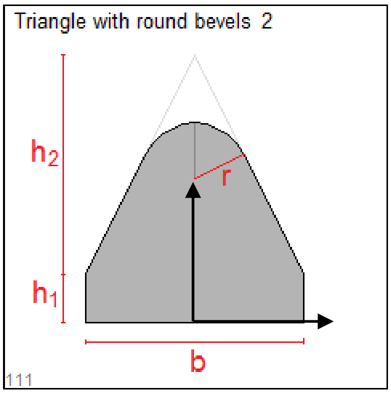
p1= b; p2= h; p3= r;



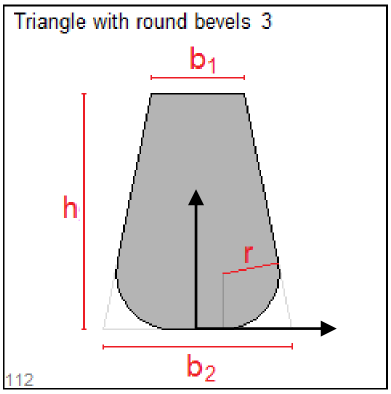
p1= b; p2= h; p3= r;



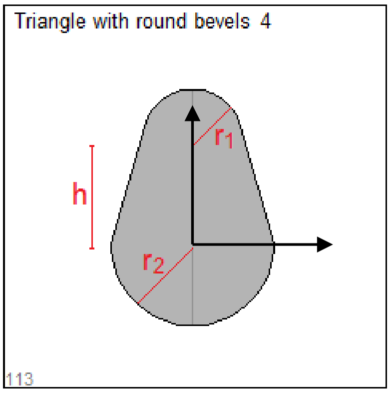
p1= b; p2= h; p3= r;



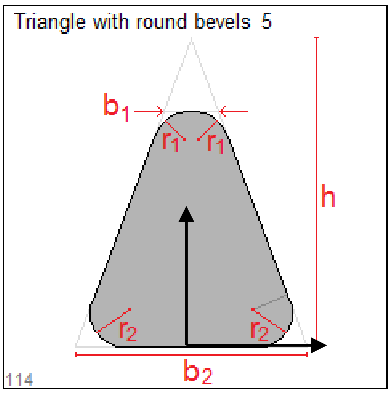
p1= b; p2= h1; p3= h2; p4= r;



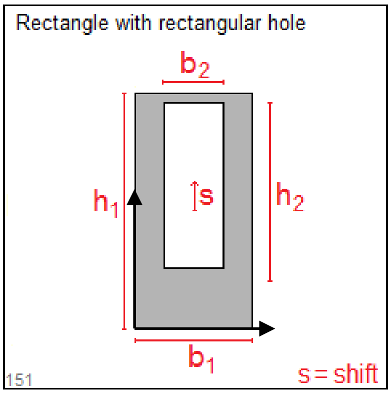
p1= b1; p2= b2; p3= h; p4= r;



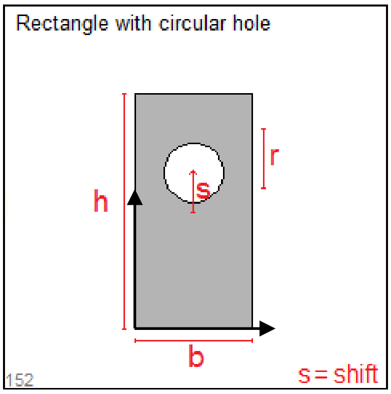
p1= r1; p2= r2; p3= h;



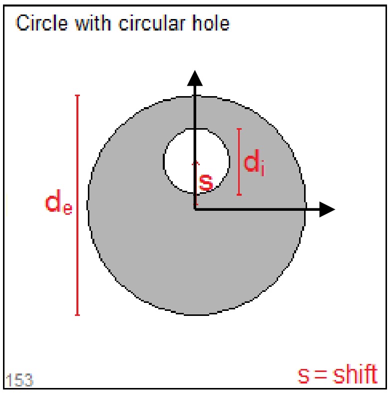
p1= h; p2= b1; p3= b2; p4= r1; p5= r2;



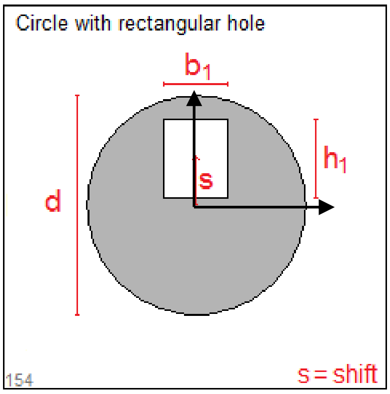
p1= b1; p2= h1; p3= b2; p4= h2; p5= s;



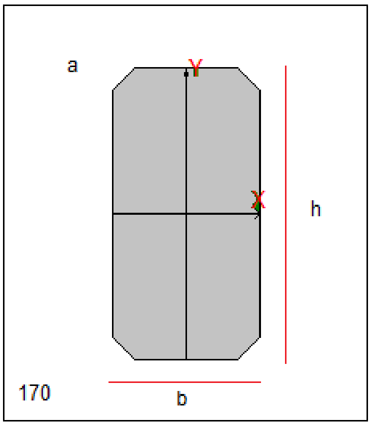
p1= b; p2= h; p3= d; p4= s;



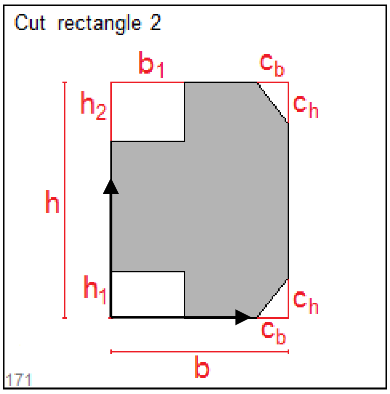
p1= de; p2= di; p3= s;



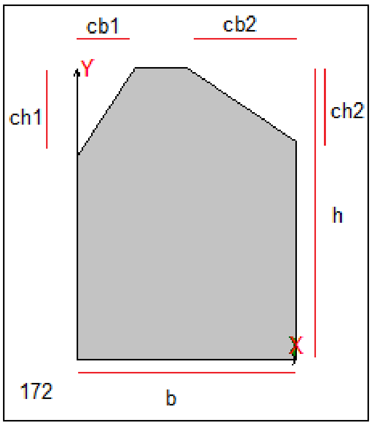
p1= d; p2= b1; p3= h1; p4= s;



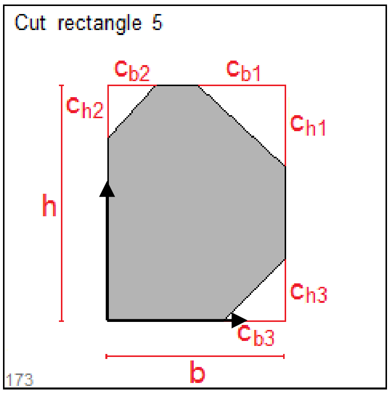
p1=b; p2=h; p3=a



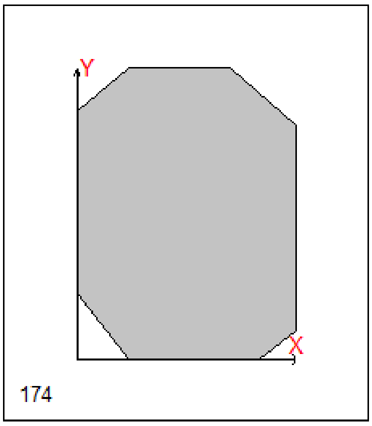
p1= b; p2= h; p3= b1; p4= h1; p5= h2; p6= cb; p7= ch;



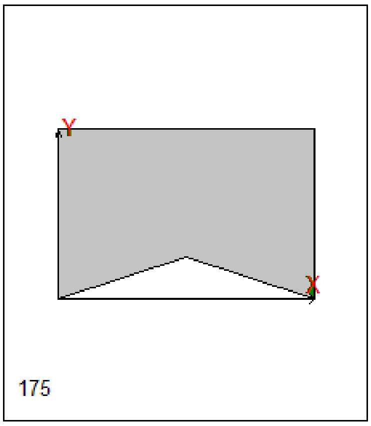
p1=b; p2=h; p3=cb1; p4=ch1; p5=cb2; p6= ch2;



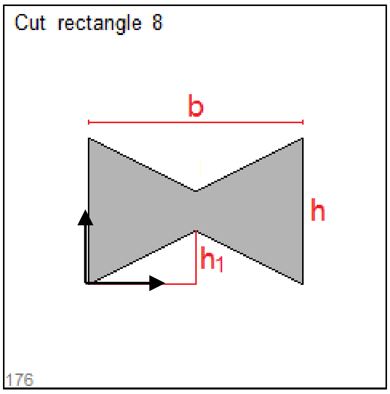
p1= b; p2= h; p3= cb1; p4= ch1; p5= cb2; p6= ch2; p7= cb3;p8= ch3;



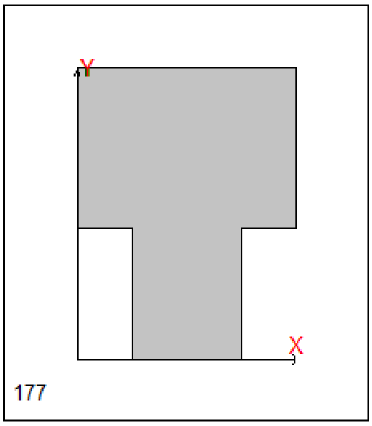
p1= b; p2= h; p3= cb1; p4= ch1; p5= cb2; p6= ch2; p7= cb3;p8= ch3; p9=cb4; p10=ch4



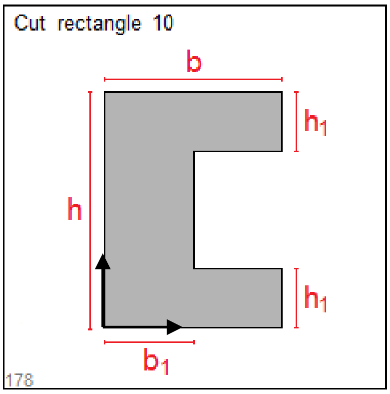
p1= b; p2= h; p3= h1;



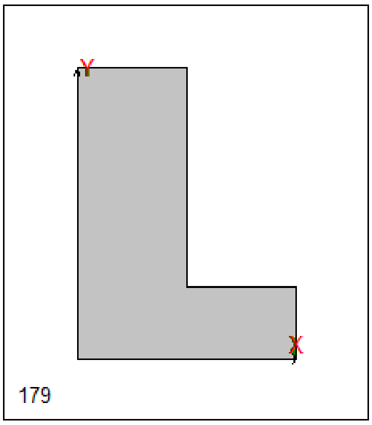
p1= b; p2= h; p3= h1;



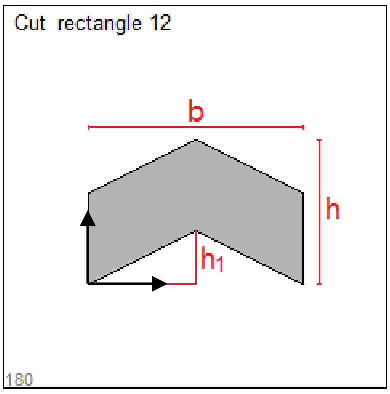
p1= b; p2= h; p3= b1; p4=h1



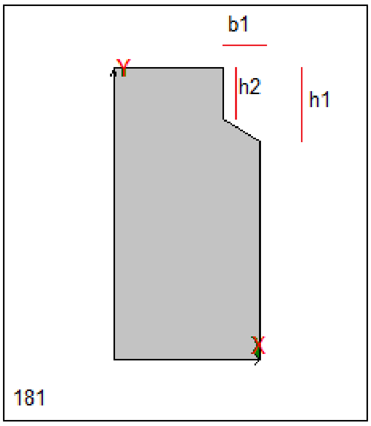
p1= b; p2= h; p3= b1; p4= h1;



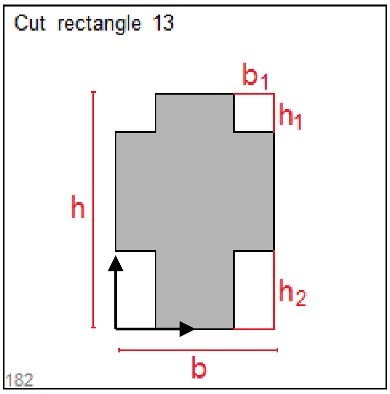
p1= b; p2= h; p3= b1; p4= h1;



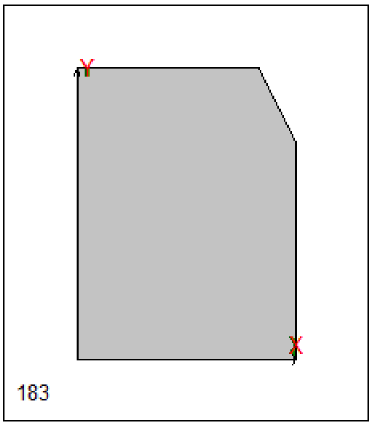
p1= b; p2= h; p3= h1;



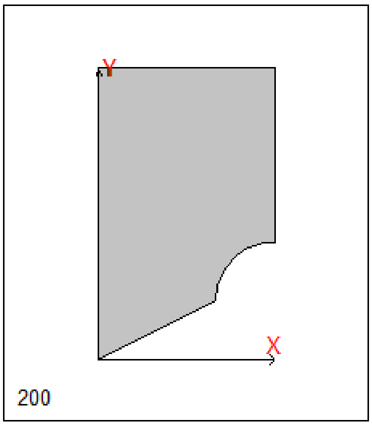
p1=b; p2=h; p3=b1; p4=h1; p5=h2



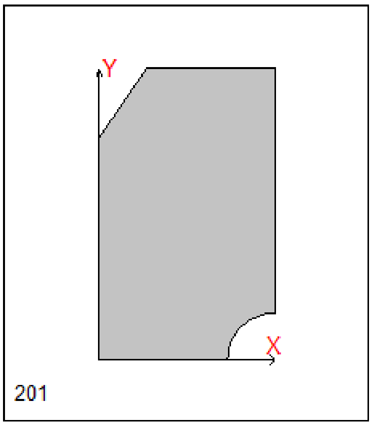
p1= b; p2= h; p3= b1; p4= h1; p5= h2;



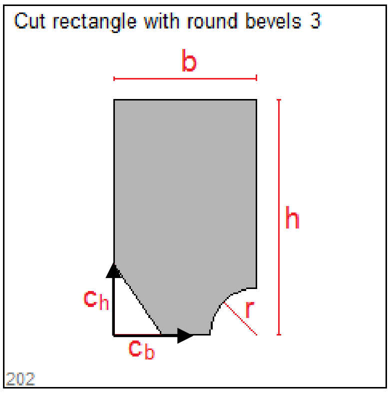
p1=b; p2=h; p3=cb; p4= ch



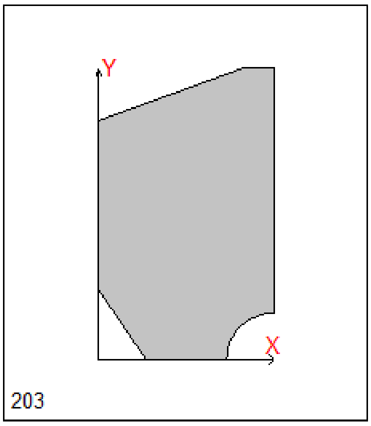
p1=b; p2=h; p3=r; p4=alpha(deg)



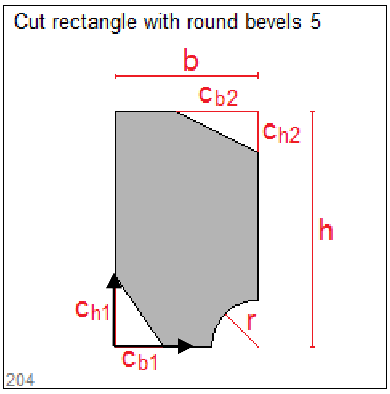
p1=b; p2=h; p3=cb; p4=ch; p5=r



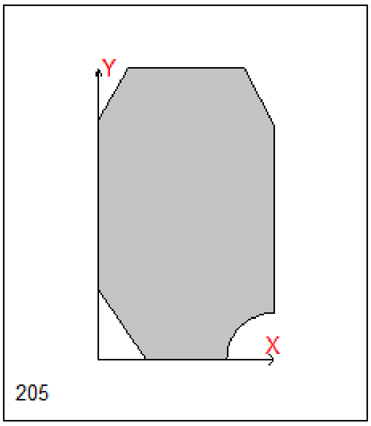
p1= b; p2= h; p3= cb; p4= ch; p5= r;



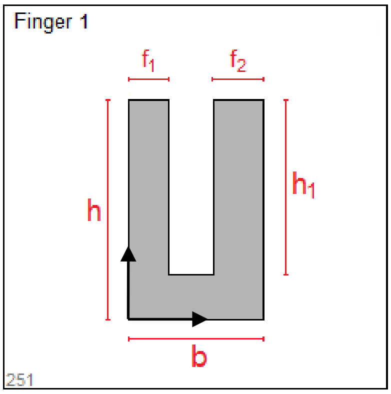
p1= b; p2= h; p3= cb1; p4= ch1; p5=cb2, p6=ch2, p7= r;



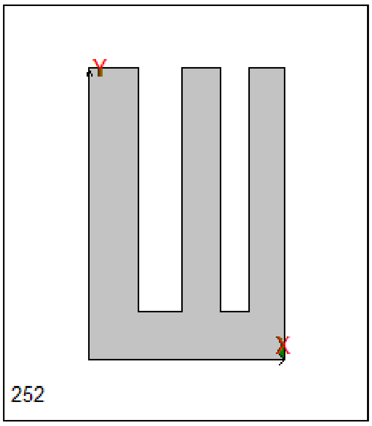
p1= b; p2= h; p3= cb1; p4= ch1; p5= cb2; p6= ch2; p7= r;



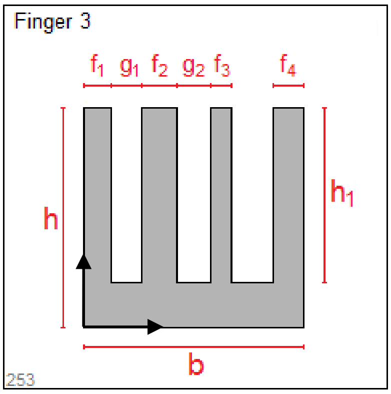
p1= b; p2= h; p3= cb1; p4= ch1; p5= cb2; p6= ch2; p7=cb3, p8=ch3; p9= r;



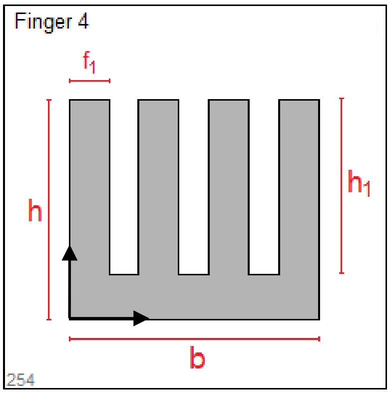
p1= b; p2= h; p3= h1; p4= f1; p5= f2;



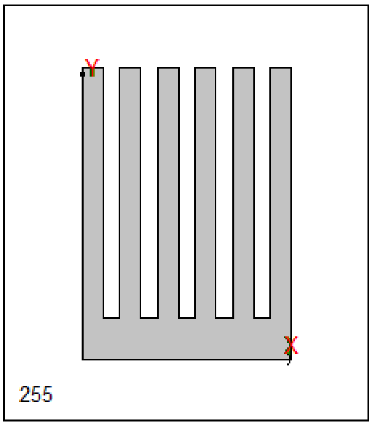
p1= b; p2= h; p3= h1; p4= f1; p5=g1; p6= f2; p7=f3



p1= b; p2= h; p3= h1; p4= f1; p5= g1; p6= f2; p7= g2;p8= f3; p9= f4



p1= b; p2= h; p3= h1; p4= f;



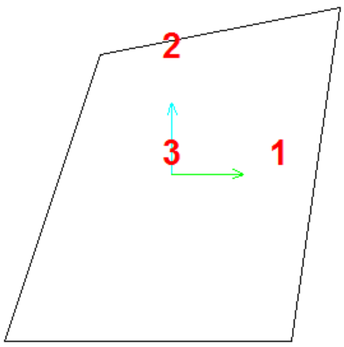
p1= b; p2= h; p3= h1; p4= f;

4.5.2 Generic plates

Generic plates are defined by two closed polylines: one for the external perimeter, and one for the internal perimeter of a possible hole.

External polyline is always needed. Internal polyline is optional.

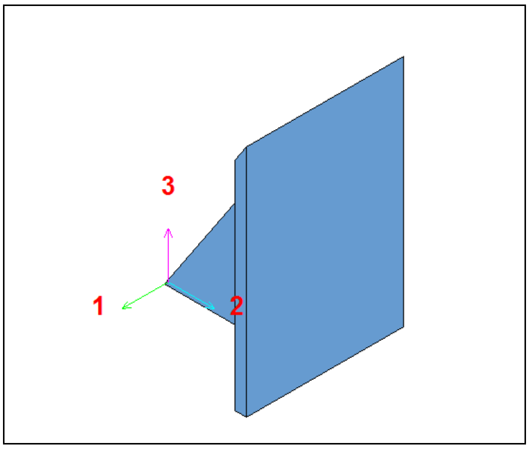
Each polyline is defined by a set of points (x_1, x_2) in plane (1, 2).



For instance, the previous generic plate (4 points polygon) is identified by the following coordinates:

```
4
-1.75000000e+002  -1.75000000e+002 ;  Point     1 (x1, x2)
 1.25000000e+002  -1.75000000e+002 ;  Point     2 (x1, x2)
 1.75000000e+002   1.75000000e+002 ;  Point     3 (x1, x2)
-7.50000000e+001   1.25000000e+002 ;  Point     4 (x1, x2
0
```

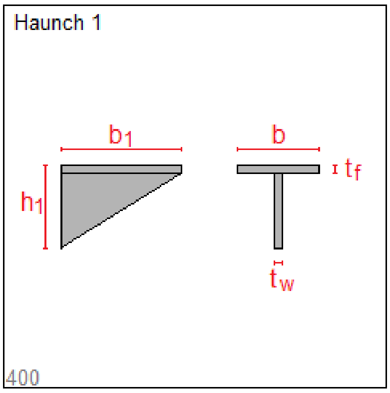
4.6 Composed Plates

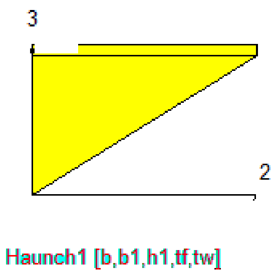


An example of CPLATE

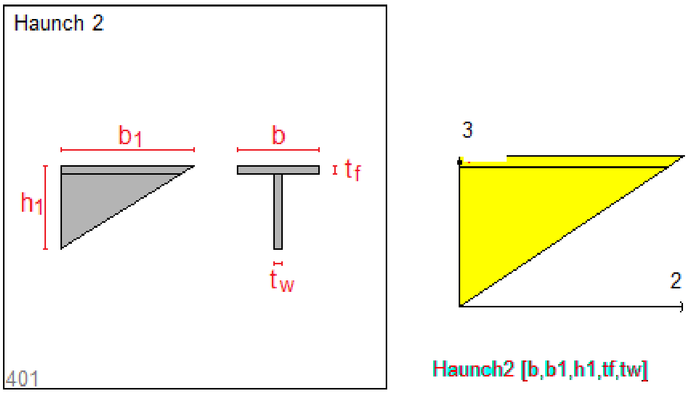
Composed plates are special objects which are identified by a type number, and by a set of 10 parameters. Usually, less than 10 parameters are needed, but the definition of a CPLATE always takes 10 parameters as input.

In the following pictures, the type number is in the bottom left corner of left pictures, and the origin of local coordinate system (1, 2, 3) is shown in the right pictures. In the pictures caption is listed the correct sequence of the parameters to be used.

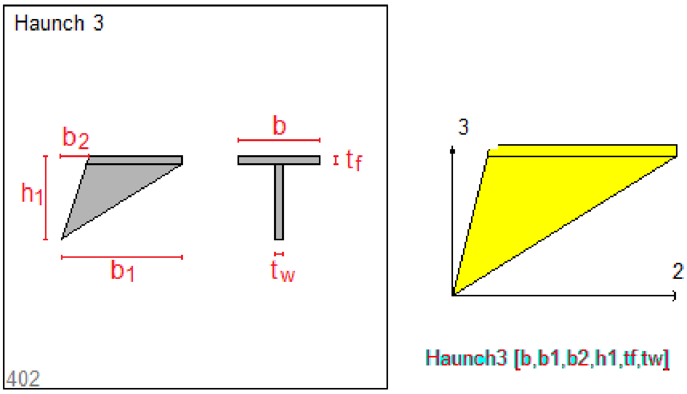




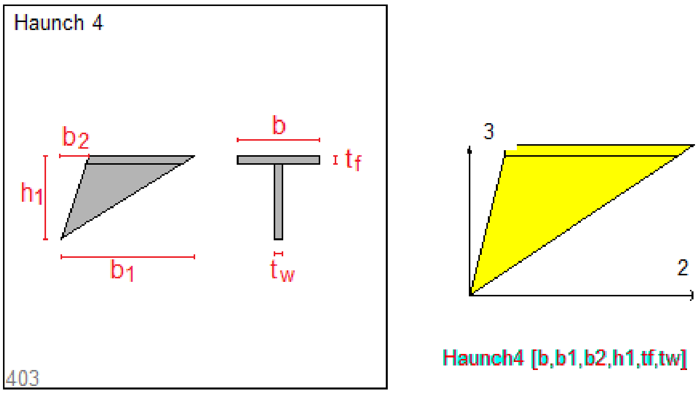
p1= b; p2= b1; p3= h1; p4= tf; p5= tw;



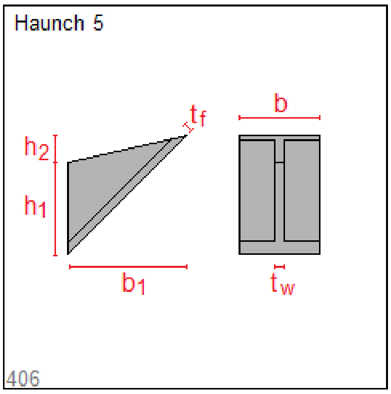
p1= b; p2= b1; p3= h1; p4= tf; p5= tw;

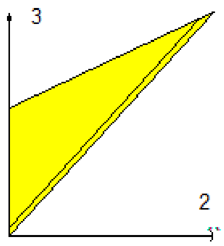


p1= b; p2= b1; p3= b2; p4= h1; p5= tf; p6= tw

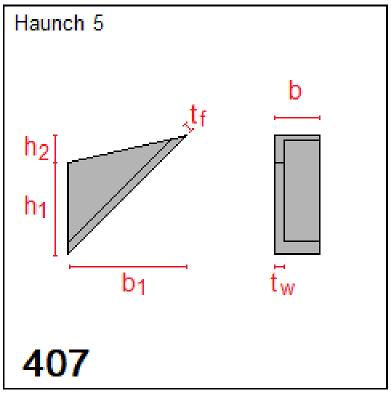


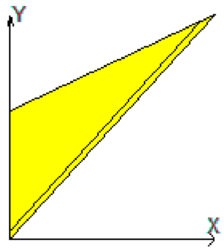
p1= b; p2= b1; p3= b2; p4= h1; p5= tf; p6= tw





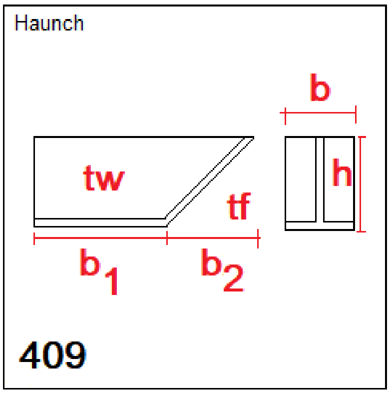
Haunch5 [b,b1,h1,h2,tw,tf]

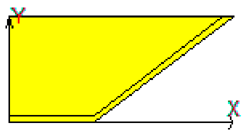




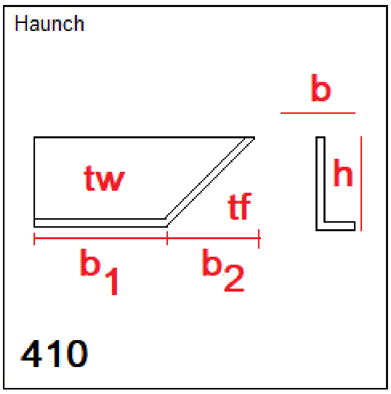
Haunch6 ([) [b,b1,h1,h2,tw,tf]

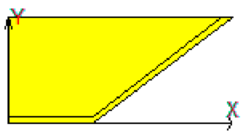
Number 408 is leftish.



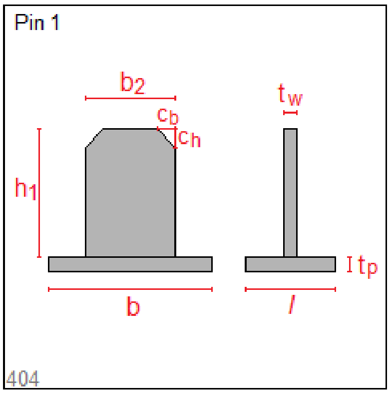


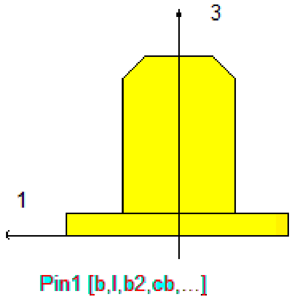
Haunch8 (]) [b,b1,b2,h,tw,tf]



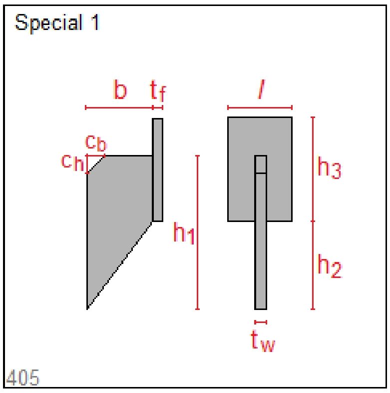


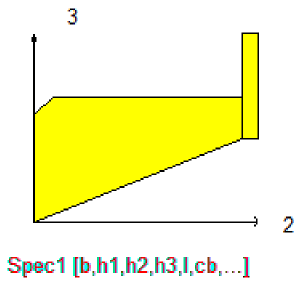
Haunch9 (|) [b,b1,b2,h,tw,tf]





p1= b; p2= l; p3= b2; p4= cb; p5= ch; p6= h1; p7=tp; p8=tw

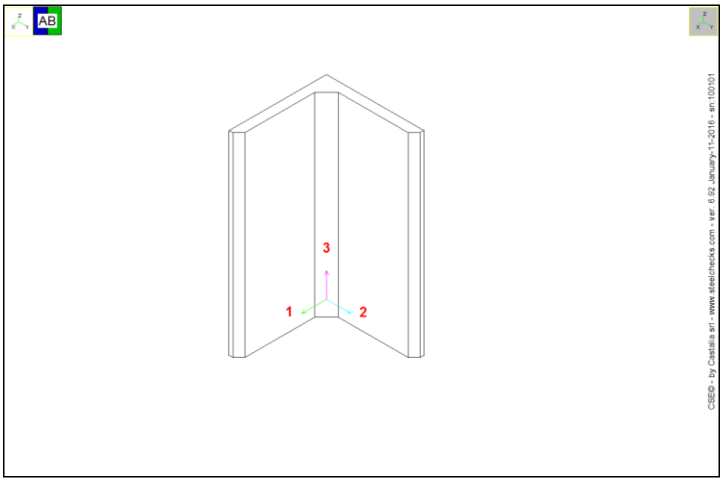




p1= b; p2= h1; p3= h2; p4= h3; p5= l; p6= cb; p7=ch; p8=tw; p9=tf

CPLATEs must have a material and can receive work processes like any other component. CPLATES can be modeled by plate-shell finite elements, as are made up of several thin plates.

4.7 Angles



An example of angle

An “angle” is a special cross-section trunk got by extruding an L-shaped cross-section by a given length. The origin of the local coordinate system (1, 2, 3), i.e. the insertion point, is in the corner of the first extruded cross-section. Composed angles must be added as cross-section trunks, but can also be added adding more angles.

Angles must have a material and can receive work processes like any other component.

5 Work Processes

A member or a cleat can be modified by applying some work processes. A work process is a modification of the geometry of an object.

The final shape of the object is got by applying one after another all the work processes defined for it. The sequence of the work processes is very important and must be respected.

If an original object O is modified by a first work process WP1, then the result will be a different object O_1. Work process WP2 is then applied to object O_1 and cannot be applied to the original object O, which does not exist anymore.

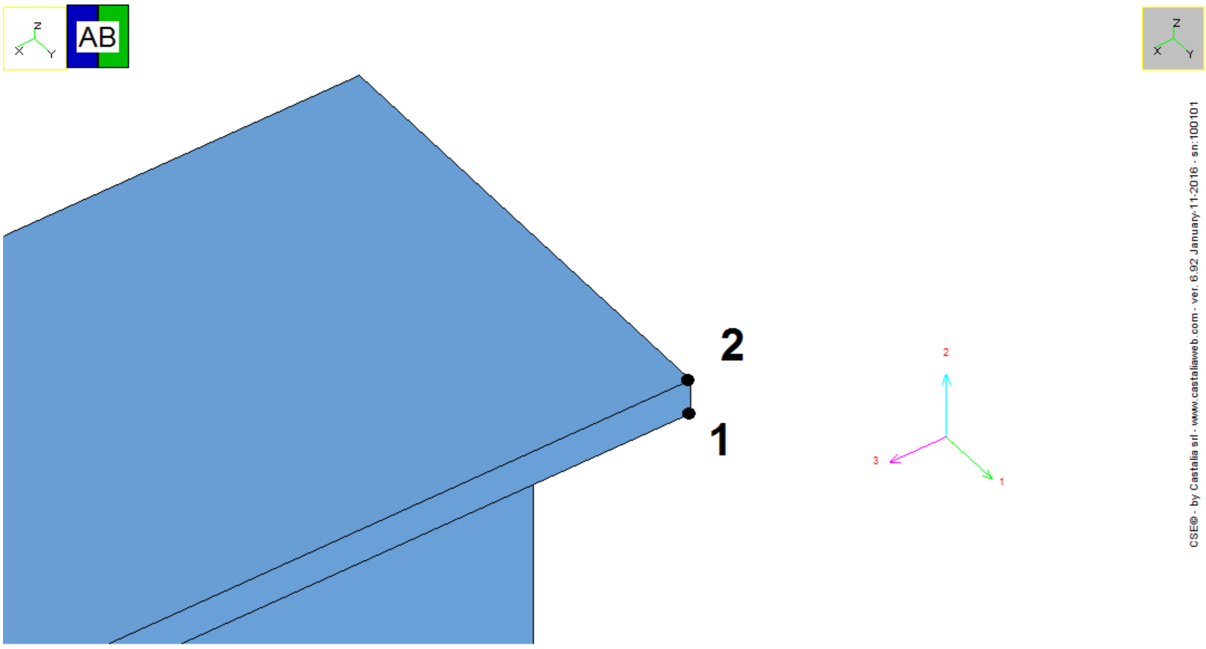
Objects O, O_1, O_2…. O_N are all referred to the same coordinate system, i.e. the cs (1, 2, 3) of the original object O.

Work processes are defined in the local coordinate system. In order to specify correctly the work processes, points in the 3D space defined **by local axes (1, 2, 3)** are used.

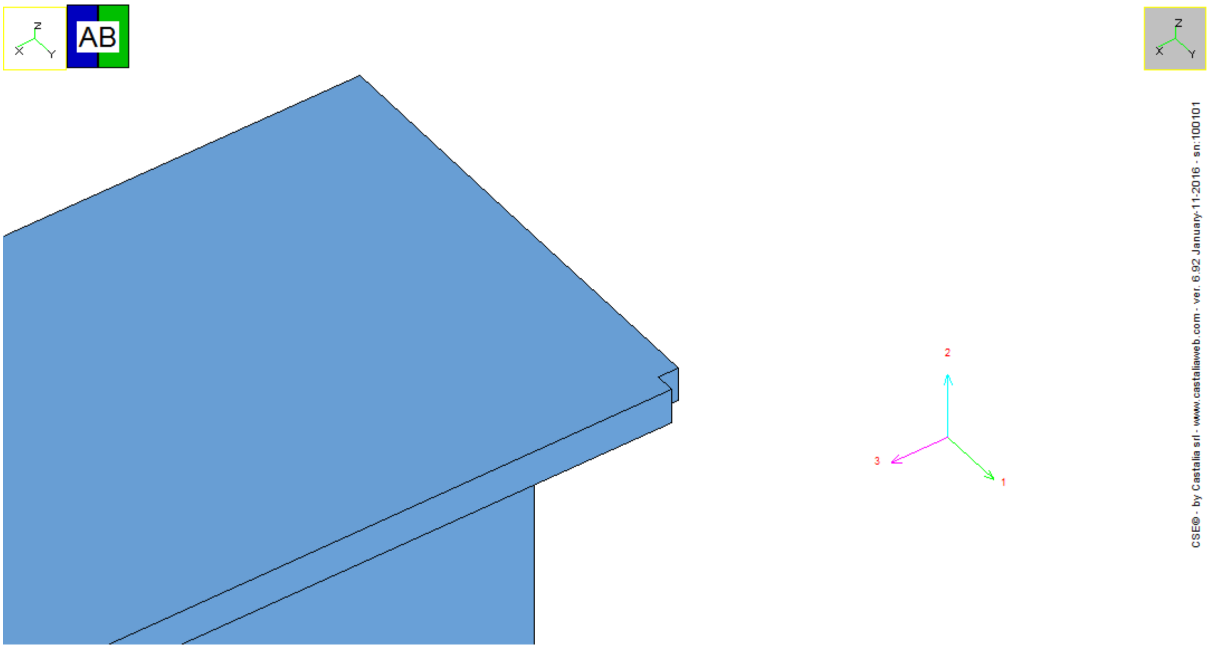
In the following sub-sections, each possible work process will be defined.

5.1 Square bevel

A square bevel is defined by two points P1 and P2 defined **in local sc (1, 2, 3)**, and by a bevel size d.



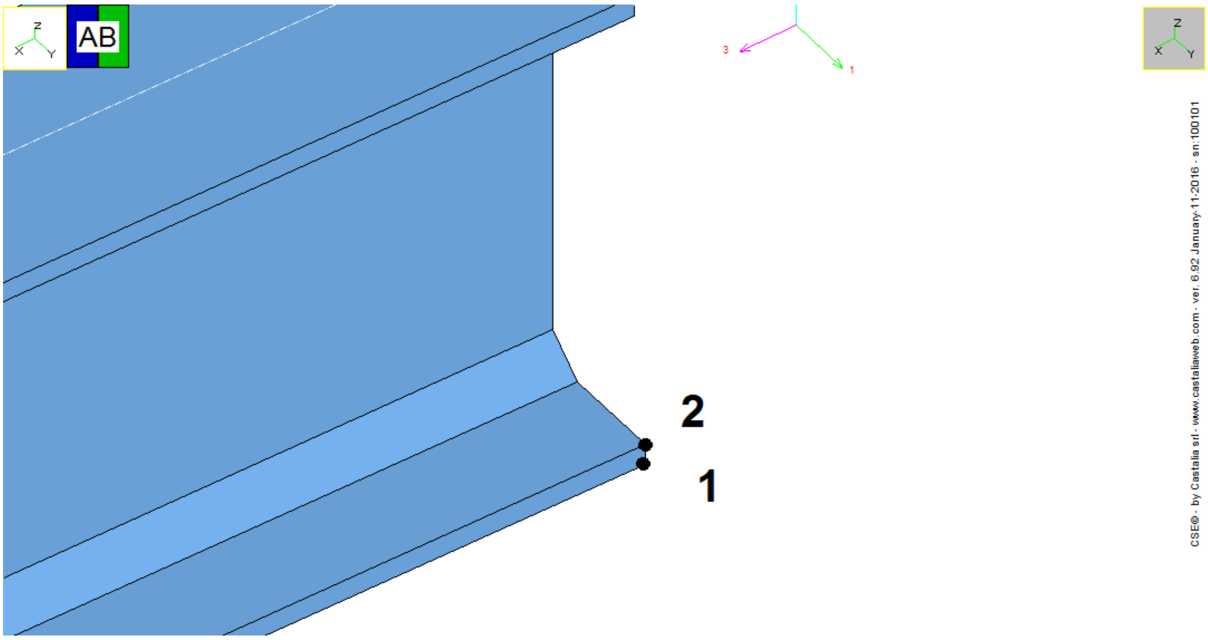
Before the bevel application



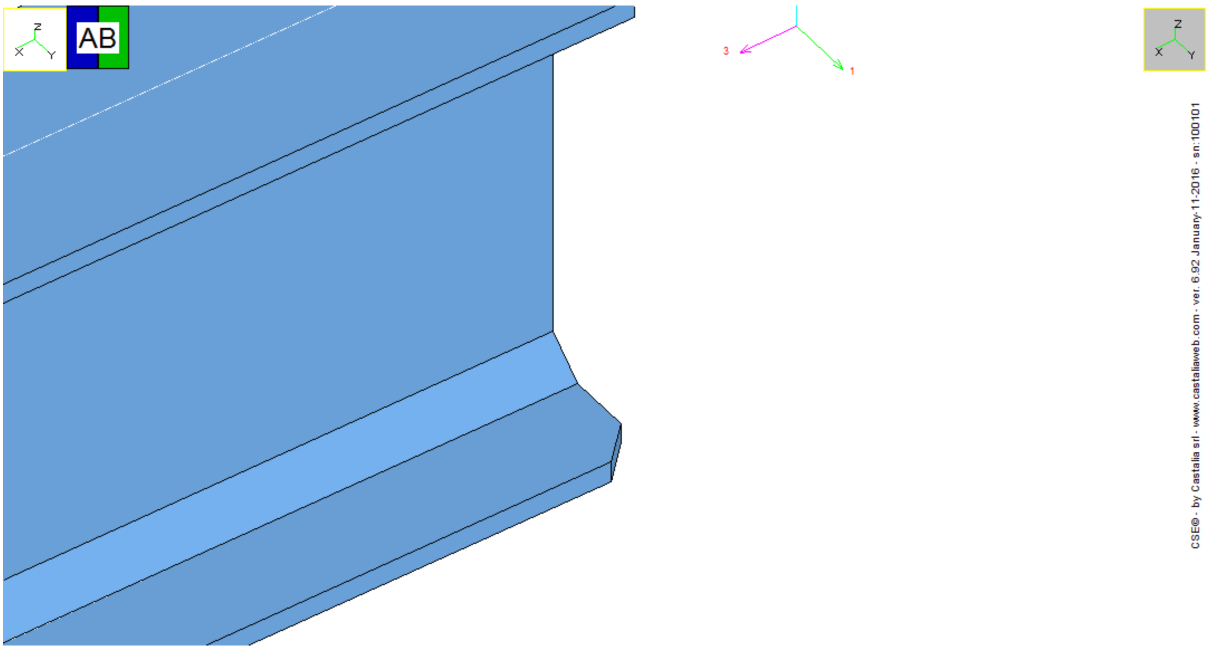
After the bevel application

5.2 Triangular bevel (equal sides)

A triangular (equal side) bevel is defined by two points P1 and P2 defined **in local sc (1, 2, 3)**, and by a bevel size d.



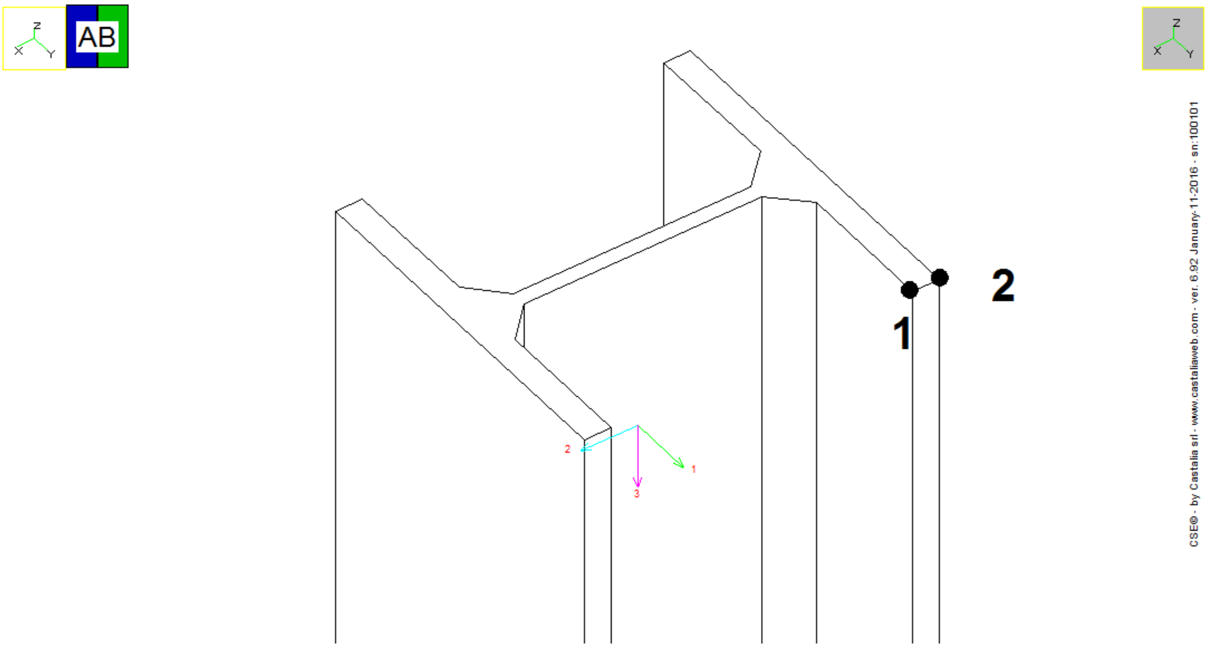
Before bevel application



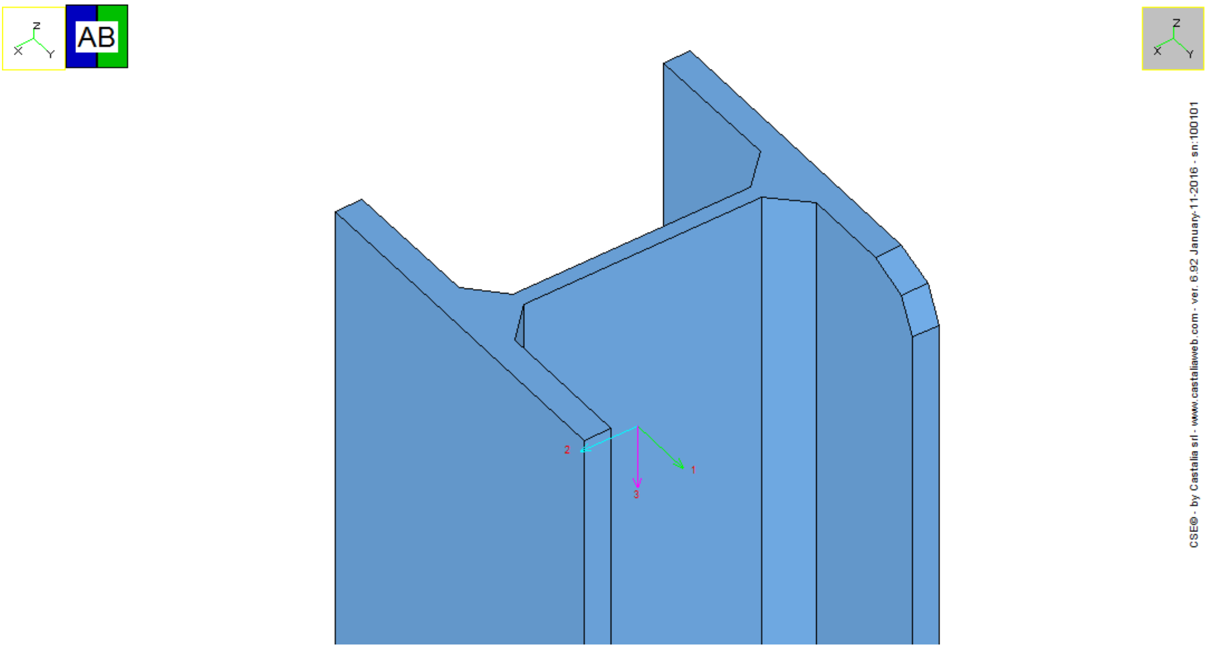
After bevel application

5.3 Circular bevel

A circular bevel is defined by two points P1 and P2 defined **in local sc (1, 2, 3)**, and by a bevel radius r.



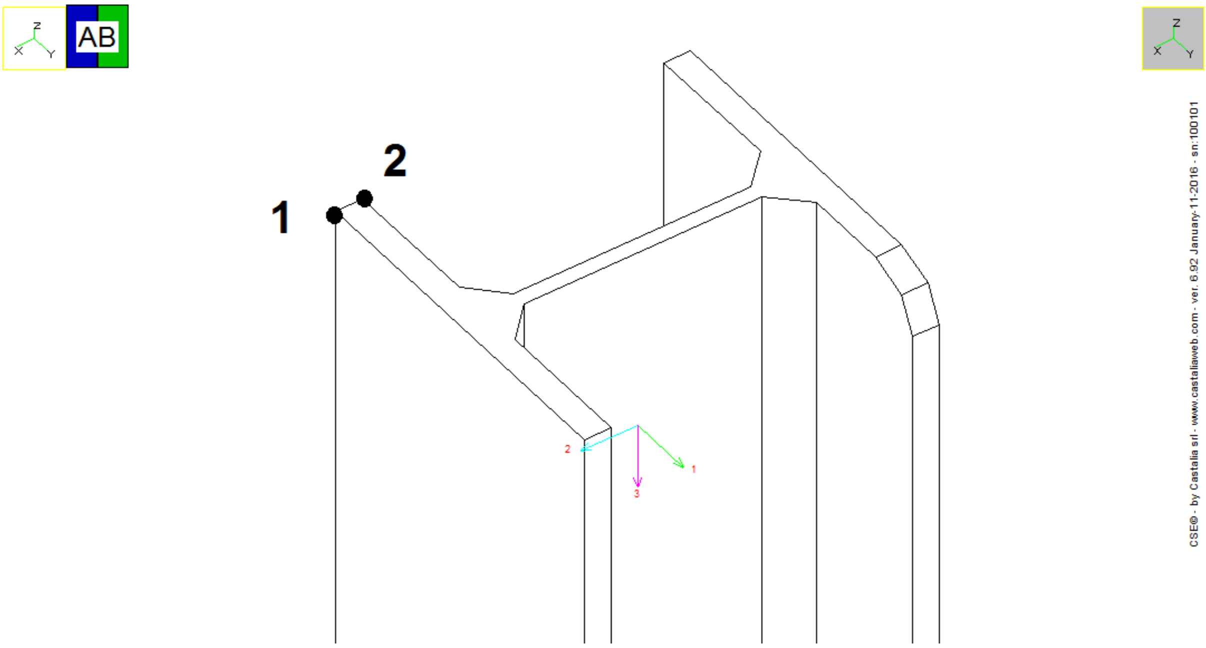
Before bevel application, local axis 3 points downward



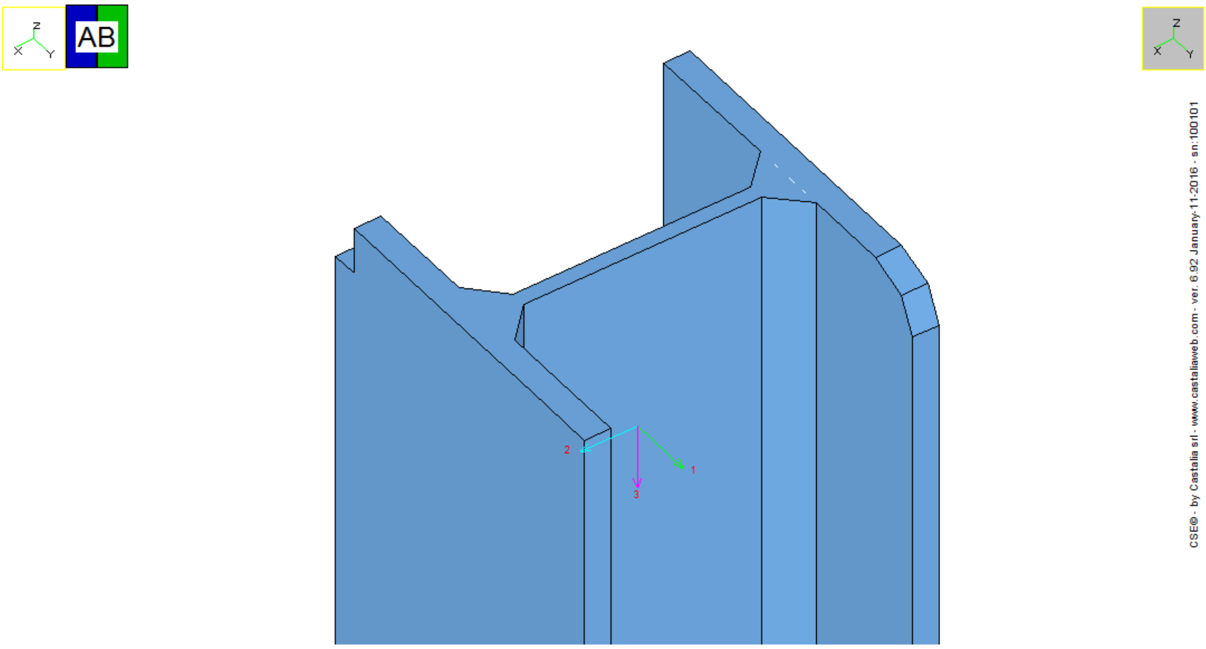
After bevel application

5.4 *Rectangular bevel*

A rectangular bevel is defined by two points P1 and P2 defined **in local sc (1, 2, 3)**, and by two bevel sizes b, and h.



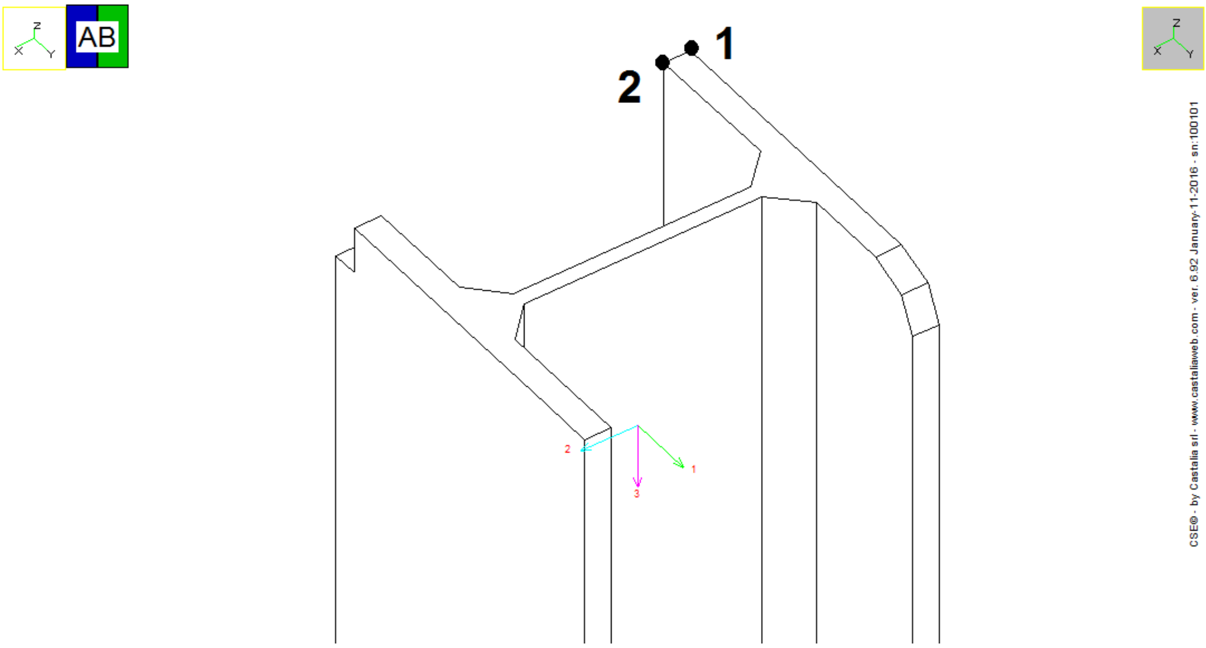
Before bevel application



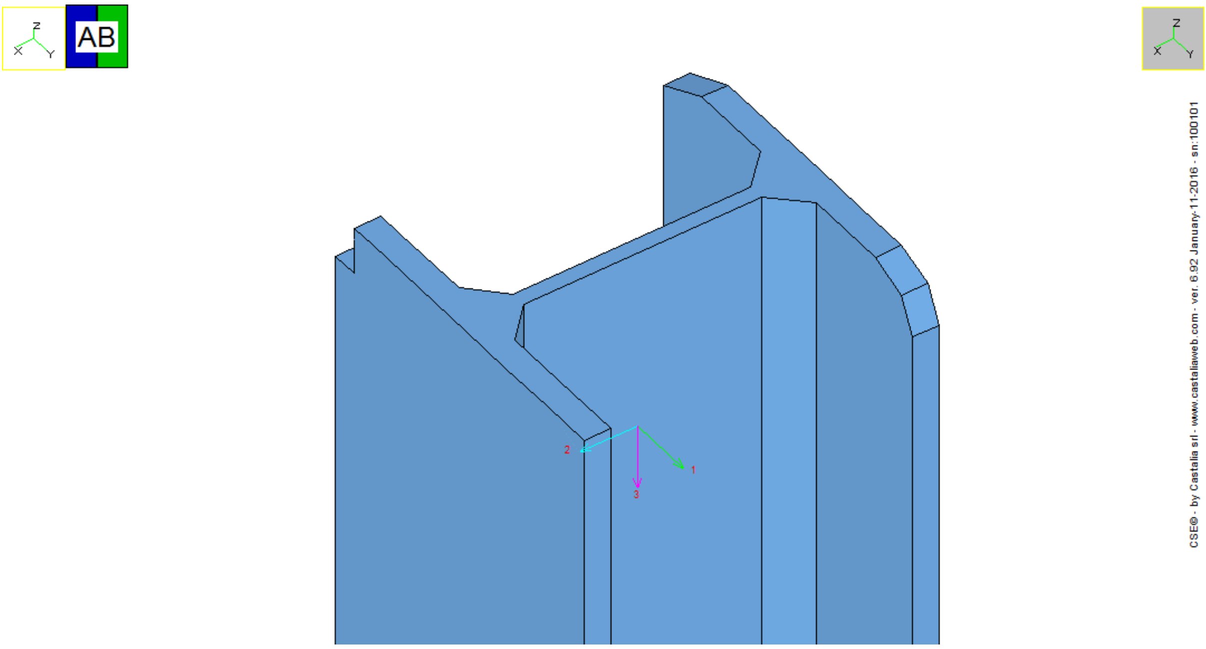
After bevel application (b= 15, h=30)

5.5 Triangular bevel (un equal sides)

A triangular bevel is defined by two points P1 and P2 defined **in local sc (1, 2, 3)**, and by two bevel sizes b, and h.



Before bevel application

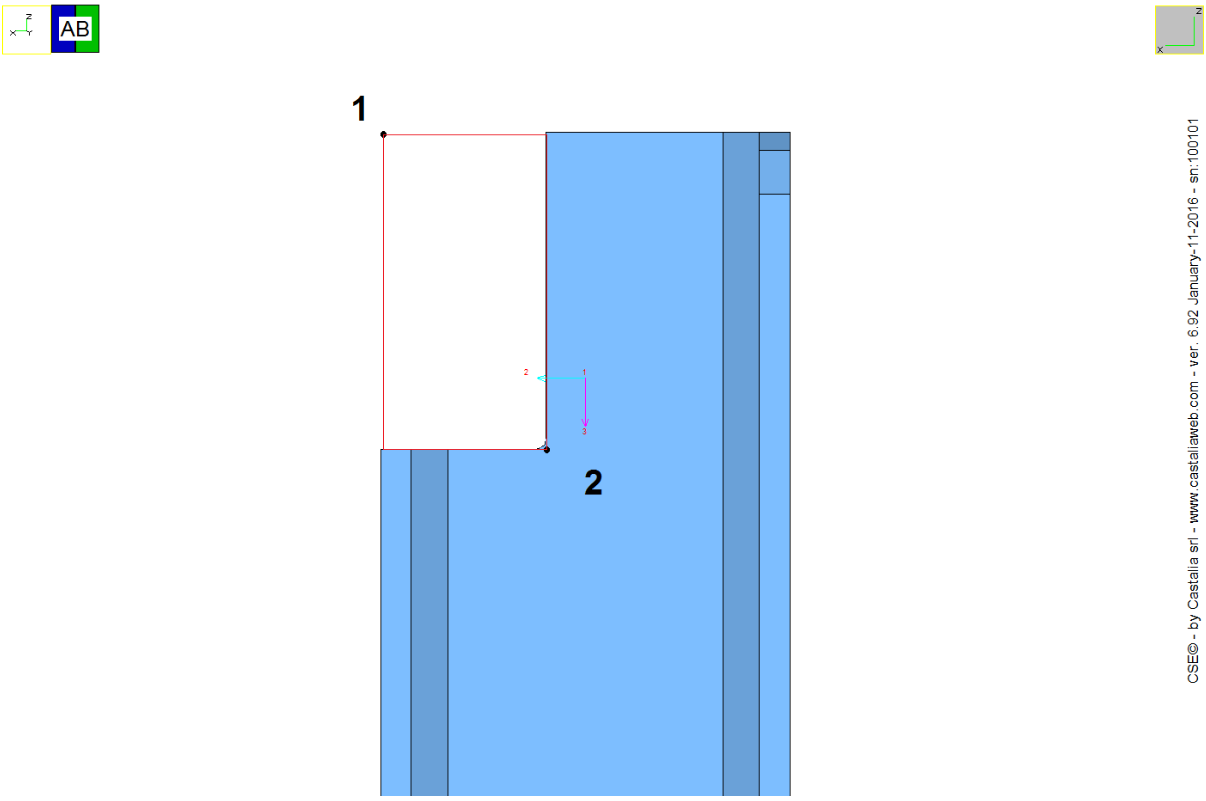


After bevel application (b=15 h=30)

5.6 Cut by box

By using a "cut by box" an infinite rectangular-base frustum F is applied to an object O, and the resulting object is got by the Boolean operation $O_{new} = O - F$.

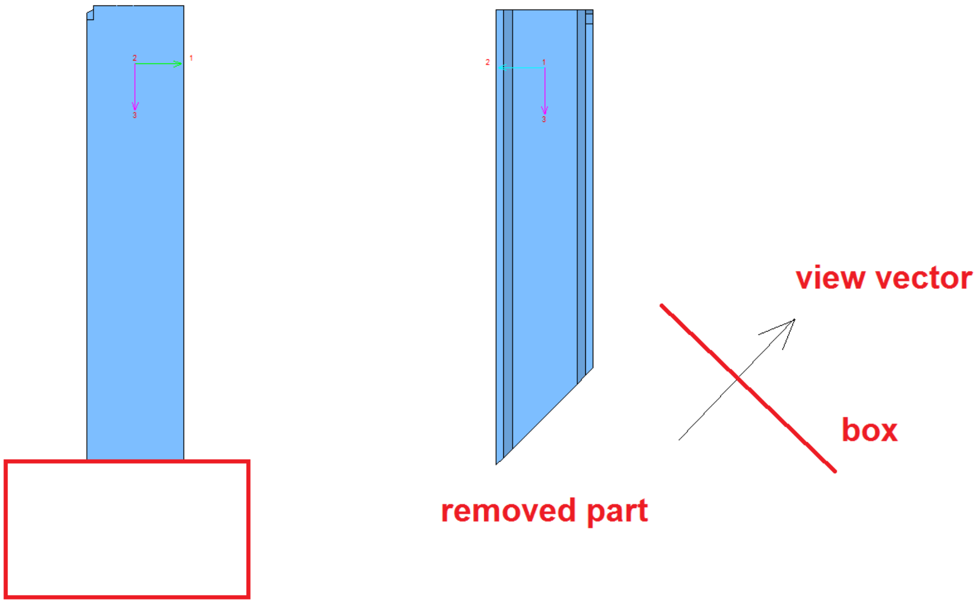
For instance the following picture shows a cut by box:



where the box is identified by choosing two opposite corners P1 and P2. The work process may also apply bevels to the four corners of the box, so, for each corner, it must be specified if the bevel is to be applied or not.

A box can be used to apply holes to plates and sub-plates.

Usually the box is defined in plane (1-2), or (2-3) or (1-3). The "view vector" is the vector normal to the plane of the box. However, a cut by box can also be defined using an inclined view vector, as in the picture below.



If the box is defined in local plane (1-2) the view vector will be axis 3 or minus axis 3.

If the box is defined in local plane (3-1) the view vector will be axis 2 or minus axis 2.

If the box is defined in local plane (2-3) the view vector will be axis 1 or minus axis 1.

A cut by box is defined by:

- A view vector in space (1, 2, 3), i.e. referring to object local axes.
- Four points defined in a projecting plane (u, v) normal to view vector, and with the origin in the projected origin of (1, 2, 3) cs. The coordinates of these four points are (u, v). Corners are numbered counterclockwise, starting from the corner with minimum u and minimum v. Projecting onto plane, also a third coordinate w can be found, the depth, but this is not used in box definition (it's a dummy value).
- Four Boolean codes referring to applicable bevels (0 no bevel, 1 bevel)
- A bevel radius (equal for all corners)

If view vector is local axis **v1**, then the coordinates will be (2, 3): $u=x_2$; $v= x_3$;

If view vector is local axis **–v1**, then the coordinates will be (-2, 3): $u= -x_2$; $v= x_3$.

If view vector is local axis **v2**, then the coordinates will be (-1, 3): $u=-x_1$; $v= x_3$; (**)

If view vector is local axis -**v2**, then the coordinates will be (1, 3): $u=x_1$; $v= x_3$;

If view vector is local axis **v3**, then the coordinates will be (1, 2): $u=x_1$; $v= x_2$;

If view vector is local axis -**v3**, then the coordinates will be (-1, 2): $u=-x_1$; $v= x_2$;

In general, the following conversions apply.

Let **view** be the view vector in local coordinate system (1, 2, 3). Let's assume by definition

C1 = **view**.x1

C2=**view**.x2

C3=**view**.x3

S3 = sqrt(1-C3²)

If S3 is not null, matrix T is defined as:

$$T = \begin{vmatrix} -\frac{c_2}{s_3} & \frac{c_1}{s_3} & 0 \\ -\frac{c_1 c_3}{s_3} & -\frac{c_2 c_3}{s_3} & s_3 \\ c_1 & c_2 & c_3 \end{vmatrix}$$

If S3 is null, then T is defined as

$$T = \begin{vmatrix} c_3 & 0 & 0 \\ 0 & 1 & s_3 \\ c_1 & c_2 & c_3 \end{vmatrix}$$

Given a Point **P** in space (1, 2, 3,), having coordinates **g**=(x_1, x_2, x_3), the projected coordinates **f**=(u, v, w) of the point can be got by

Tg=**f**

Given a point P(u, v, w) (one of the corner of the box in projected space (u, v, w), it will be described by a vector **f**(x_1, x_2, x_3) in un-projected system (1, 2, 3). The following system of equations

Tg=**f**

can be inverted to give the vector **g** as solution.

Let's assume the view vector is (0, 1, 0), i.e. axis 2, **v2**. Then

C1=0

C2=1

C3=0

S3=1

$$T = \begin{vmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{vmatrix}$$

A point {x_1, x_2, x_3} in projected plane gets:

$u=-x_1$

$v=x_3$

$w=x_2$ (unused in box definition)

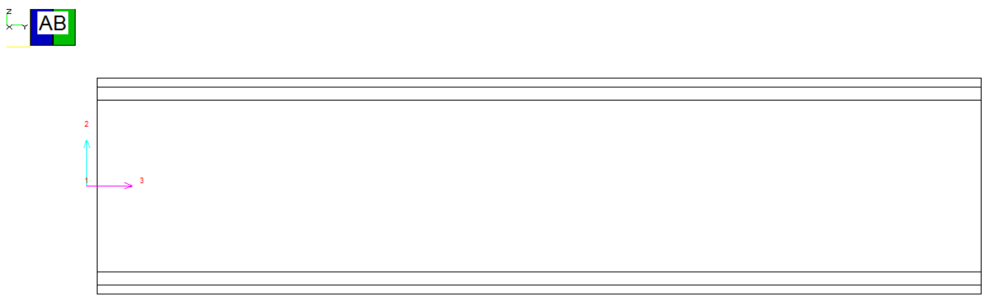
These is formula (**) anticipated previously. So if the four corners of the box are defined in space (1, 2, 3) along with the view vector (0, 1, 0), the four corners of the box have the projected coordinates that can be found by applying (**).

In order to define the work process, the view vector **view** and the projected coordinates of the box must be input.

5.7 Cut by poly

A cut by poly is very similar to a cut by box. The only difference is that as cut-tool it is used a closed polyline defined by a number of points in projected plane. Also here it is necessary a view vector, the projected coordinates of the polyline vertices, and a Boolean code addressing the existence of the bevel at each corner.

The maximum number of points of the polyline is presently 13.



Before a cut by poly. Here view is (-1, 0, 0)

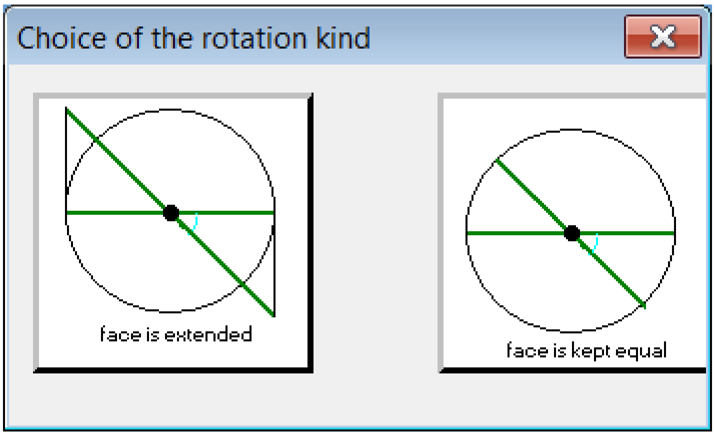
After a cut by poly (no bevel).

5.8 Face rotation

Objects can be modified by applying a rotation to one of their faces.

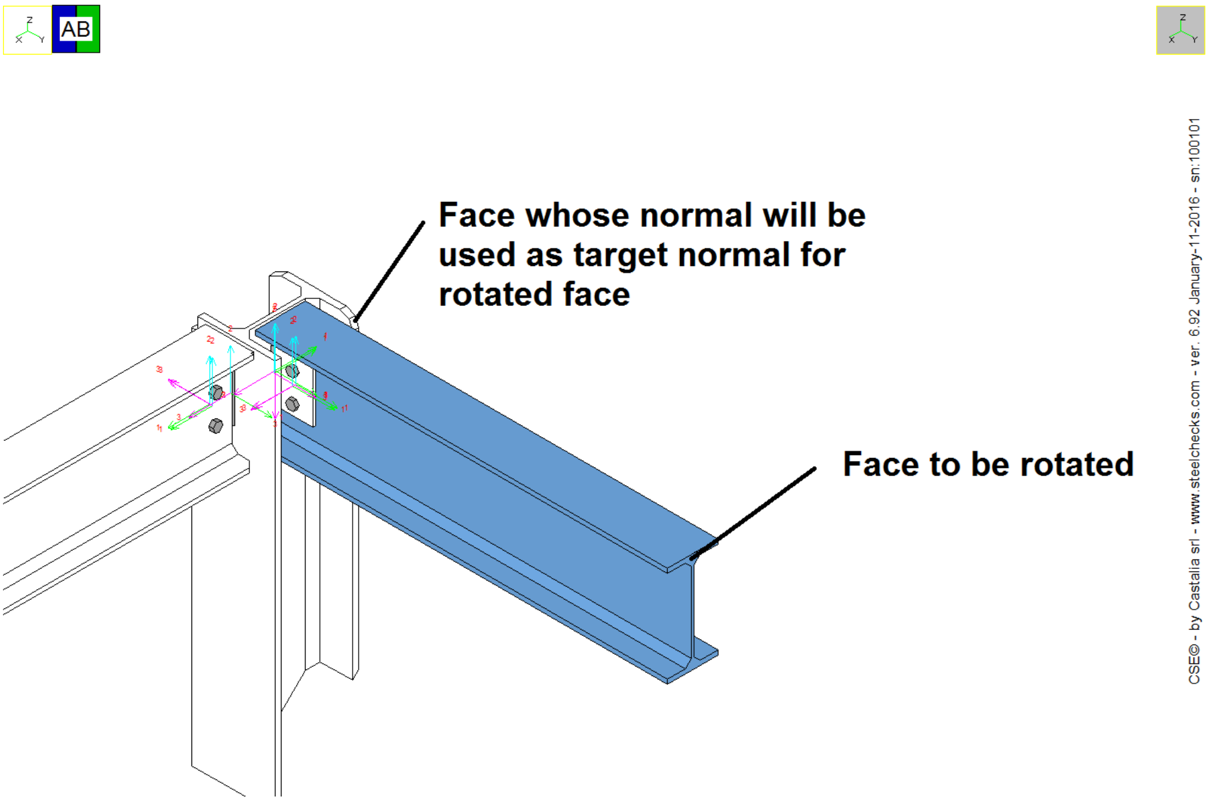
The face to be rotated is identified by its outward normal vector in space (1, 2, 3) before rotation, and by the coordinates (x_1, x_2, x_3) in object local cs (1, 2, 3) of one of its points. This basically defines a plane. All coplanar faces of the object will be rotated, which is useful for end-faces of composed objects.

The rotation is identified by a vector in space (1, 2, 3) which is the outward normal that the rotated face will get *after rotation*, and by a boolean code specifying the “rotation mode”.

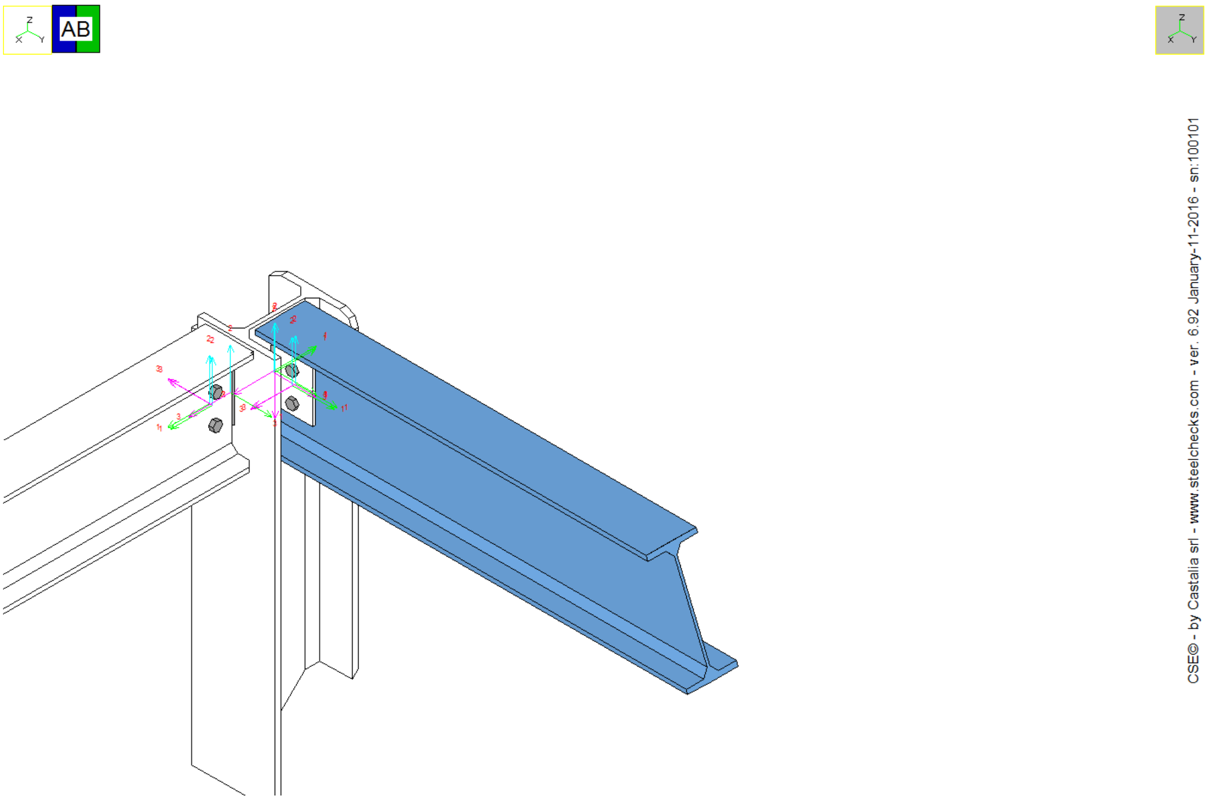


There are two rotation modes:

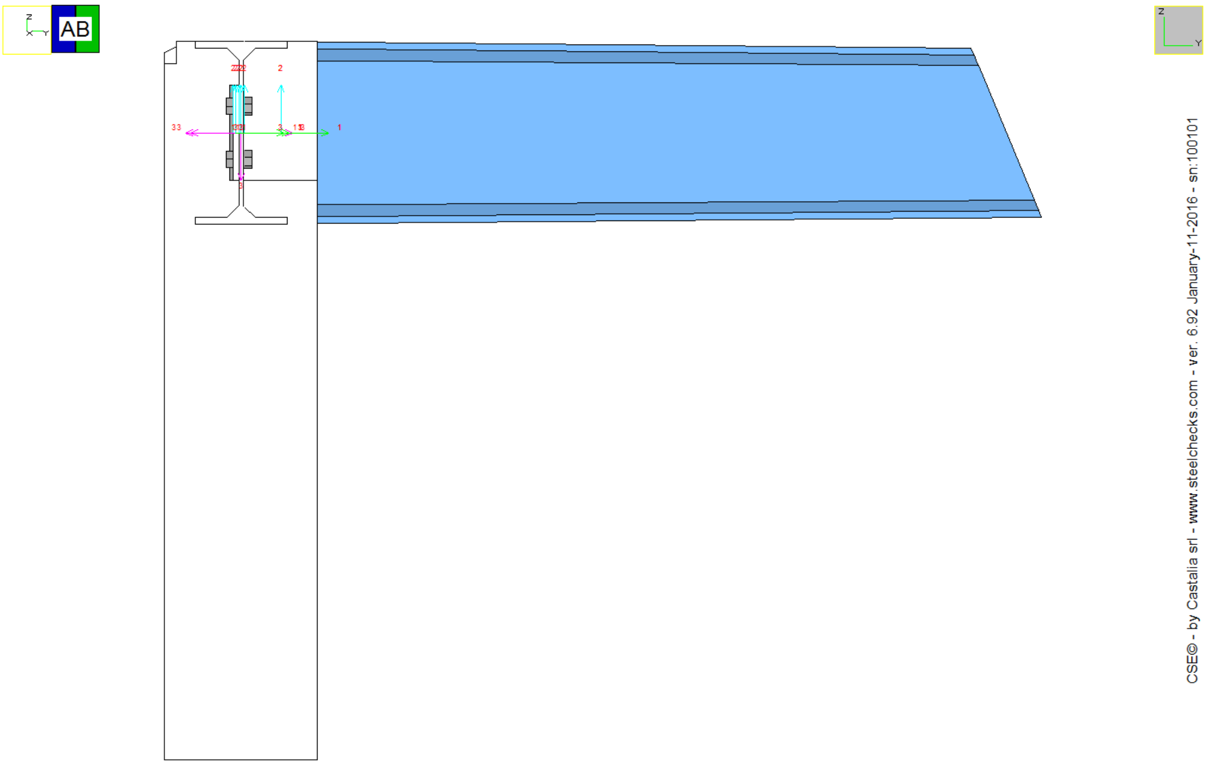
- A first mode rotates the face extending its sizes.
- A second mode rotates the face keeping all its sizes identical.



Face before rotation



Face after rotation: it has been extended



Here face has been kept equal. The member is now tapered.

5.9 Face translation

Objects can be modified by applying a translation to one of their faces.

The face to be rotated is identified by its outward normal vector in space (1, 2, 3), and by the coordinates (x_1, x_2, x_3) in object local cs (1, 2, 3) of one of its points. This basically defines a plane. All coplanar faces of the object will be shifted, which is useful for end-faces of composed objects.

The shift is a distance, which is positive if directed as the outward normal of the face to be translated, and negative if directed as the inward normal.

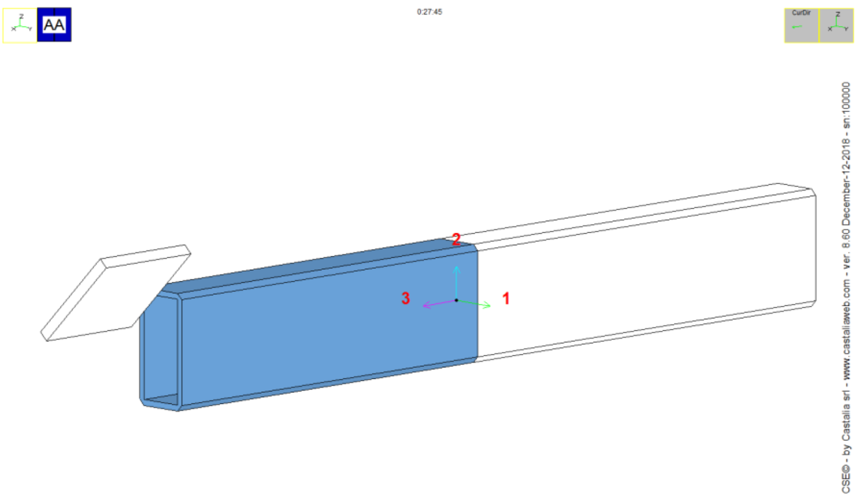
5.10 Cut by plane

The cut-by-plane work process has been added in December 2018. Given a plane of equation

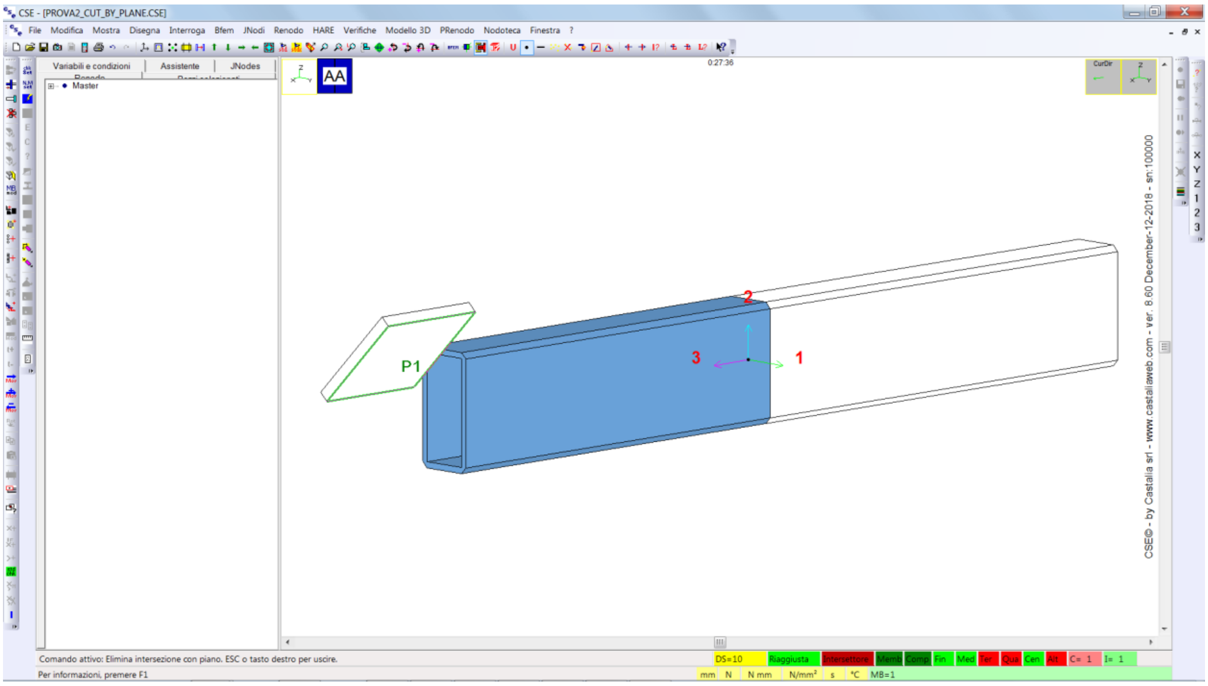
$$ax_1+bx_2+cx_3+d=0$$

in the local reference system (**1**, **2**, **3**) of the object that will receive the work process, the part of the object posed in negative half space will be removed ($ax_1+bx_2+cx_3+d<0.$), and the parts laying in the positive half space will be kept ($ax_1+bx_2+cx_3+d>0$).

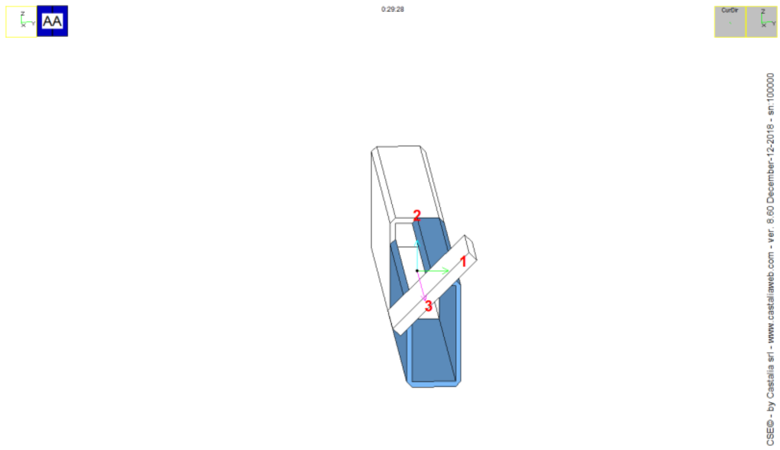
It must be underlined that the (x_1,x_2,x_3) are the local coordinates in the local reference system of the object.



Selected object before cut by plane



Choice of the plane



After the cut

Given the equation of a plane in global reference system

ax+by+cz+d=0

The equation of the plane in local system

$$Ax_1+Bx_2+Cx_3+D=0$$

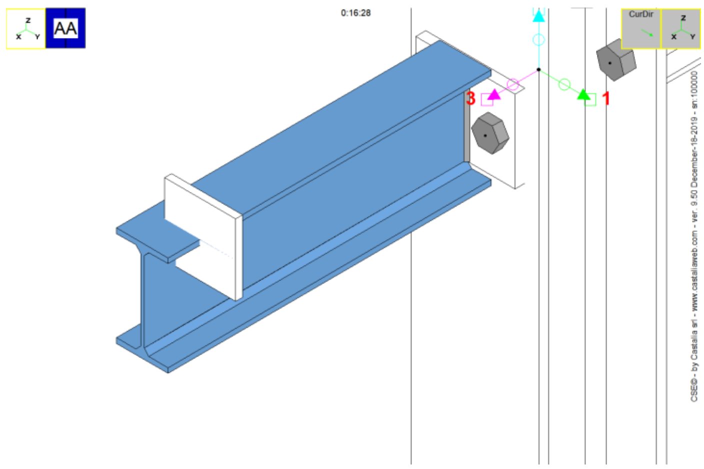
Can be derived thanks to the following transformation:

$$D = (\mathrm{point} + \mathrm{mv}) \cdot (a,b,c) + d$$

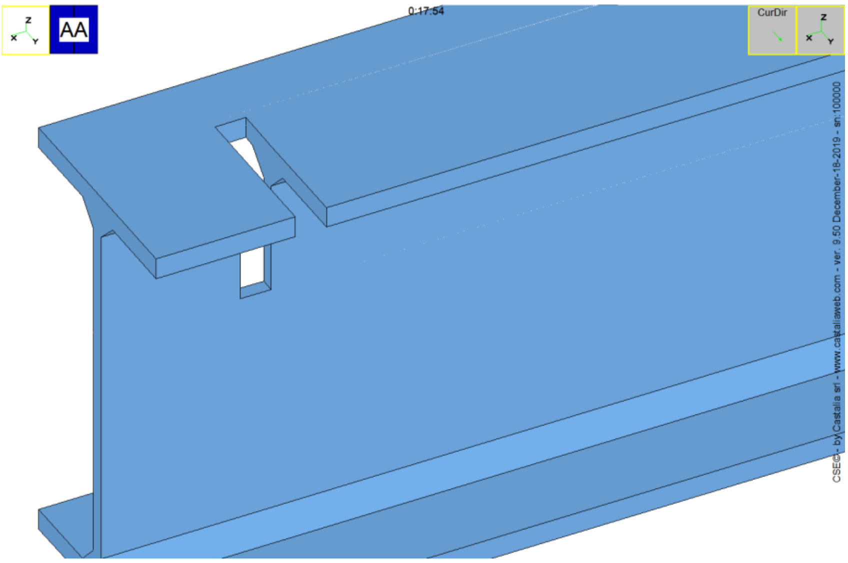
$$\{A,B,C\}^T = \mathbf{T}^\mathrm{T} \{a,b,c\}$$

5.11 Boolean subtraction

Given a component A, that will receive the work-process, and another component B, the Boolean subtraction (A-B) is a new solid, that will replace A, obtained subtracting B from A.



Boolean subtraction from member A, of plate B



The resulting member, (A-B).

In CSE the operation needs the selection of the receiving object A, and the selection of one face of the tool-object B.

In the D3O, the object B is referenced in the local system of object A, as collection of faces with normal pointing outward (points of the face are ordered counterclockwise, when seen from outside).

6 The .D3O file

6.1 *Introduction*

In order to manage the interface to and from CSE, an exchange file format has been set up. The file has “D3O” extension, and is a normal ASCII file.

CSE is able to export a D3O file and is also able to import it. Also, using a D3O file, it is possible to upgrade a model if some component has been moved or changed in a different program, outside CSE.

D3O file manages lists of objects. They are placed in space at their proper positions (insertion points).

File .D3O is a text file which can be written by your application (or even by hand or by using EXCEL) and that describes all the data needed to define a set of connections, including fittings, bolt layouts, and weld layouts.

.D3O file is written using Newton, mm and °C as force, length and temperature measurement units.

Basically the file is the addition of several blocks (they are 4), each of one preceded by a delimiter string, and ended by a delimiter string. Row of comments between the blocks (not inside blocks or sub-blocks) may be input and they are marked by a “$” in first column.

A data row can end with a “;” and the characters read after “;” are explanatory comments, like here:

```
   440.0000     440.0000      10.0000     100.0000     100.0000  ;  p1  p2  p3  p4 p5
```

The blocks delimiter strings are:

MATERIALS
END MATERIALS

CROSS SECTIONS
END CROSS SECTIONS

MEMBER COLLECTION
END MEMBER COLLECTION

OBJECT COLLECTION
END OBJECT COLLECTION

The blocks **MEMBER COLLECTION** and **OBJECT COLLECTION** are divided into sub-blocks.

Each of the four blocks contains a vector of entities:

- Materials used in the model are listed between **MATERIALS** and **END MATERALS**. These materials are used by members and cleats.
- The cross sections used in the model are listed between **CROSS SECTIONS** and **END CROSS SECTIONS**. These cross-sections are used by members.
- The members of the models are listed between **MEMBER COLLECTION** and **END MEMBER COLLECTION**.
- The bolt layouts, weld layouts, and cleats are listed between **OBJECT COLLECTION** and **END OBJECT COLLECTION**.

If no element of a given kind is available the related block may be omitted. Material vector cannot be omitted as it is necessary both for members and for objects collections.

The possible sub blocks for MEMBER COLLECTION are:

NEWMEMBER MODE0

This sub-block is related to a single member. It does not need a closing tag.

The possible SUB-BLOCKS for OBJECT COLLECTION are:

NEW BOLTLAYOUT MODE0

NEW WELDLAYOUT MODE0

NEWCLEAT PLATE MODE0

NEWCLEAT CPLATE MODE0

NEWCLEAT TRUNK MODE0

NEWCLEAT ANGLE MODE0

These sub-blocks each are related to a single object (a cleat or a bolt layout or a weldlayout). They do not need a closing tag.

MODE0 is a tag specifying an input mode. The same object can ideally be described in different ways, "MODE0" is the first way to do that.

Here is examples file, where are listed the following items:

1 material;

2 cross sections;

1 member;

1 plate;

1 weld layout

1 bolt layout;

```
MATERIALS
1
    1   2.100000e+005 3.000000e-001 7.700850e-005 1.200000e-005 2.350000e+002 3.600000e+002 "S235"
END MATERIALS
CROSS SECTIONS
2
    1     1   "HE 200 B            "
200.000000 200.000000 9.000000 15.000000 18.000000
    2     1   "IPE 240             "
240.000000 120.000000 6.200000 9.800000 15.000000
END CROSS SECTIONS
MEMBER COLLECTION
NEWMEMBER MODE0
"Member 1"   "Unknown" ;   Internal Identifier External Identifier
 0.00000000e+000    0.00000000e+000   0.00000000e+000   ; position
 0.00000000e+000    0.00000000e+000   0.00000000e+000   ; move from position
 0.00000000e+000    1.00000000e+000   0.00000000e+000   ; axis 1
-1.00000000e+000    0.00000000e+000   0.00000000e+000   ; axis 2
 0.00000000e+000    0.00000000e+000   1.00000000e+000   ; axis 3
 0.00000000e+000    0.00000000e+000   0.00000000e+000   ; original p1
 0.00000000e+000    0.00000000e+000   2.50000000e+003   ; original p2
1   0                      ; sect1  sect2
      0.0000         0.0000            ; elongation1(+/-)  elongation2(+/-)
    1  ;  material number
    0   ; number of work processes
END MEMBER COLLECTION
OBJECT COLLECTION
NEWCLEAT PLATE MODE0
"P1"   "" ;   Internal Identifier External Identifier
 1.44316515e-014   -4.50000000e+000  -1.50000000e+001   ; position
```

```
 0.00000000e+000    0.00000000e+000   0.00000000e+000   ; move from position
 6.12300000e-017    1.00000000e+000   0.00000000e+000   ; axis 1
-1.00000000e+000    6.12300000e-017   0.00000000e+000   ; axis 2
 0.00000000e+000    0.00000000e+000   1.00000000e+000   ; axis 3
    8   1.50000000e+001  ;  type  thickness
    440.0000     440.0000      10.0000     100.0000     100.0000  ;  p1  p2  p3  p4 p5
    100.0000     100.0000     100.0000     100.0000     100.0000  ;  p6  p7  p8  p9 p10
    1  ;  material number
    0   ; number of work processes
NEW WELDLAYOUT MODE0
"W1"   "" ;   Internal Identifier External Identifier
 1.42351824e-014    3.55879560e-015   0.00000000e+000   ; position
 0.00000000e+000    0.00000000e+000   0.00000000e+000   ; move from position
 1.00000000e+000   -0.00000000e+000   0.00000000e+000   ; axis 1
 0.00000000e+000    1.00000000e+000   0.00000000e+000   ; axis 2
-0.00000000e+000   -0.00000000e+000   1.00000000e+000   ; axis 3
   0     8; kind nwelds
   1     11.000     90.000  8.50000000e+001 -2.25000000e+001  8.50000000e+001 -1.00000000e+002; weld# thick[i],…
   2      7.000     90.000 -6.70000000e+001 -4.50000000e+000  6.70000000e+001 -4.50000000e+000; weld# thick[i],…
   3     11.000     90.000 -8.50000000e+001 -1.00000000e+002 -8.50000000e+001 -2.25000000e+001; weld# thick[i],…
   4     11.000     90.000 -1.00000000e+002  1.00000000e+002 -1.00000000e+002 -1.00000000e+002; weld# thick[i],…
   5     11.000     90.000 -8.50000000e+001  2.25000000e+001 -8.50000000e+001  1.00000000e+002; weld# thick[i],…
   6      7.000     90.000  6.70000000e+001  4.50000000e+000 -6.70000000e+001  4.50000000e+000; weld# thick[i],…
   7     11.000     90.000  8.50000000e+001  1.00000000e+002  8.50000000e+001  2.25000000e+001; weld# thick[i],…
   8     11.000     90.000  1.00000000e+002 -1.00000000e+002  1.00000000e+002  1.00000000e+002; weld# thick[i],…
NEW BOLTLAYOUT MODE0
"B1"   "" ;   Internal Identifier External Identifier
 1.44316515e-014   -4.50000000e+000   0.00000000e+000   ; position
 0.00000000e+000    0.00000000e+000   0.00000000e+000   ; move from position
 1.00000000e+000    0.00000000e+000   0.00000000e+000   ; axis 1
 0.00000000e+000    1.00000000e+000   0.00000000e+000   ; axis 2
 0.00000000e+000    0.00000000e+000   1.00000000e+000   ; axis 3
  0    6  1       22.0000 0       0.0000; boltset  boltclass isfullreactive  diameter, isprecision, extra
 1    4   ;  kind of boltlayout  number of bolts
   2     2      320.0000      320.0000  0 ;  nrows  ncols drows dcols  isemptyinside
 0.00000000e+000  0.00000000e+000        0.0000 ;  odx  ody  blangle
   2       15.0000     600.0000       0.0000       0.0000       0.0000;  nthicks thick1 thick2 thick3 thick4 thick5
            0.0000       0.0000       0.0000       0.0000       0.0000;         thick6 thick7 thick8 thick9 thick10
   2        0.0000       0.0000       0.0000       0.0000       0.0000;
            0.0000       0.0000       0.0000       0.0000      ;
END OBJECT COLLECTION
```

We shall now review all blocks one by one.

6.2 MATERIALS

This block of data is used to define a vector of materials. Material numbers in this vector (1 to nmat) will later be used to assign a material to members, or cleats.

The format of this block ***is the same of format .SR4***, which is used by CSE/SARGON to import or export fem models. The only difference is that in .D3O the block is closed by the row “END MATERIALS”.

```
MATERIALS
1
    1   2.1e+005 3.0e-001 7.700850e-005 1.2e-005 2.35e+002 3.60e+002 "S235"
END MATERIALS
```

MATERIALS

NUMBER_OF_MATERIALS

{MAT_NUMBER E NU WDEN ALPHA FY FU “NAME”}
NUMBER_OF_MATERIALS times

END MATERIALS

E	Young's modulus
NU	Poisson's coefficient
WDEN	weight density
ALPHA	thermal expansion coefficient
FY	the yield stress (MPa)
FU	the ultimate stress (MPa)
"NAME"	Name of the material between quotation marks

6.3 CROSS SECTIONS

This block of data is used to define a vector of cross-sections. Cross-section numbers in this vector (1 to nsections) will later be used to assign a cross section to members.

The format of this block ***is the same of format .SR4***, which is used by CSE/SARGON to import or export fem models. The only difference is that in .D3O the block is closed by the row "END CROSS SECTIONS".

```
CROSS SECTIONS
6
    1     1   "HE 200 B            "
200.000000 200.000000 9.000000 15.000000 18.000000
    2     4   "L 75x50x7           "
75.000000 50.000000 7.000000 7.000000 3.500000
    3     3   "UPN 200             "
200.000000 75.000000 8.500000 11.500000 11.500000 6.000000
    4    35   "TMIPE 160           "
80.000000 82.000000 5.000000 7.400000 9.000000
    5    22   "L2T100x10d25        "
100.000000 100.000000 10.000000 12.000000 6.000000 25.000000 1
    6     9   "720px_cstd          "
267.000000 150.000000 100.000000 8.000000 7.000000 10.000000
```

```
END CROSS SECTIONS
```

CROSS SECTIONS

NUMBER_OF_CROSS_SECTIONS

{CROSS_SECTION_NUMBER CROSS_SECTION_KIND “NAME”}

{data depending on cross section kind }

NUMBER_OF_CROSS_SECTIONS times

END CROSS SECTIONS

The block **{data depending on the cross section kind}** is as follows.

With the exception of:

1) Cross-sections whose kind is 0

2) cross section Cold, Poli, Composed

this block is one only row of data. It lists cross section dimensions:

{p1 p2 ……pn}

where “n” is the number of parameters depending on the cross-section kind.

If a cross-section kind “0” is specified, then the NAME will be used to find the right cross-section in the Sargon / CSE cross-section archive (file .SMA).

The file “WSR_SR4_SHPCVT.TXT” in the Sargon or CSE installation folder will be opened.

If Tekla is specified as the program which created the D3O (when import command is executed), then a different file will be opened, and its name is “WSR_TKL_SHPCVT.TXT”. In this way D3O might embed the same cross section names of those used in Tekla.

The file has two columns. The NAME specified will be searched **in the second column**. If it will be found, the cross-section with the related name **in the first column**, will be searched **in file .SMA** (Sargon or CSE cross section archive). If found, that cross section will be matched to “NAME”. If no section will be found in SMA file, or no section will be found in the second column, an error message will be issued.

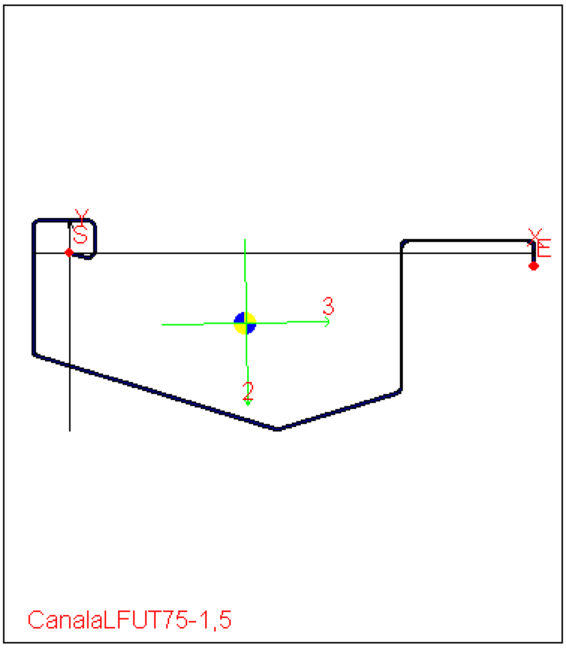
If a cross section kind different from “0” is specified, but not of the kind “Cold”, “Poli” or “Composed” (27, 28, 34), then the following table applies.

Cross-section kind	Cross section kind number	Parameters (in order: p1,p2,...)
Hsm (rolled I or H)	1	h,b,a,e,r
Ipn (I rolled, tapered flanges)	2	h,b,a,e,r,r1
Usm (rolled channels, tapered flanges)	3	h,b,a,e,r,r1
Lsm (angles)	4	h,b,a,r,r1
Tsm (T rolled, tapered web)	5	unused
Psh (plates or rectangular)	6	h,b
O_ (round or CHS)	7	D,t
Hsh (welded I or H, un-equal flanges)	9	h,b,c,a,e,d
Lsh (L)	10	h,b,a,e,
Ush (C)	11	h,b,a,e,
Tsh (T)	12	h,b,a,e,
Osh (box)	13	h,b,a,e,
U_O ([ ])	15	h,b,a,e,r,r1,d
U_H (] [)	16	h,b,a,e,r,r1,d
L2_T_sm (2 angles _||_)	22	h,b,a,r,r1,d,side side: 0 if contact along short side side: 1 if contact along long side
L2_CR_sm (2 angles ><)	23	h,b,a,r,r1,d
L4_CR_sm (4 angles +)	24	h,b,a,r,r1,d
Juan (generic)	25	A,A2,A3,J1,J2,J3
Rhs (rectangular hollow sections, round corners)	26	h,b,a,r
Composed (generic composed)	27	**See subsection**

Cold (generic cold formed)	28	**See subsection**
Omcf (hat, cold formed, with or without lips)	29	h,b,d,t,r
Ucf (C, cold formed, with or without lips)	30	h,b,d,t,r
Zcf (Z, cold formed, with or without lips)	31	h,b,d,t,r
Lcf (L, cold formed, with or without lips)	32	h,b,d,t,r
U_H_cf (2 channels Ucf] [)	33	h,b,d,t,r,dist
Poli (generic made up by polygons)	34	**See subsection**
Thsm (Tee cut from Hsm)	35	h,b,a,e,r
Asb (asymmetric beam flanges)	36	h,b_{up},b_{dn},a,e,r
Usm2 (channels, parallel flanges)	37	h,b,a,e,r

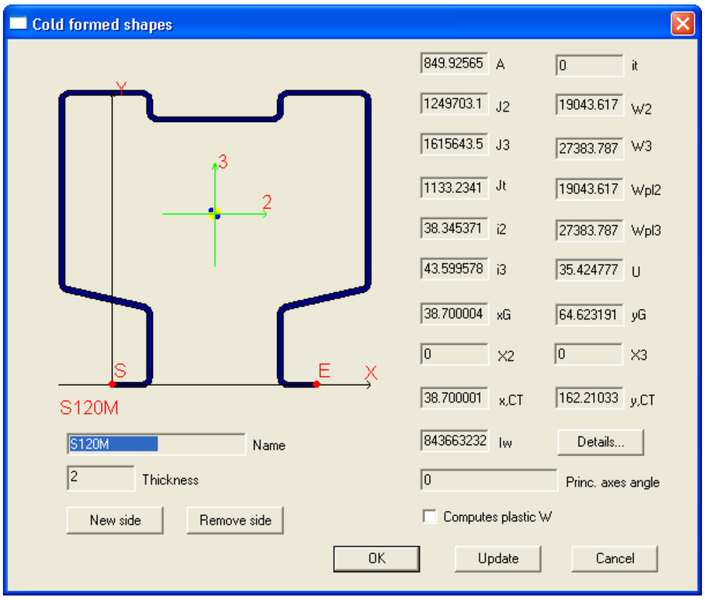
The cross sections of the kind Cold, Composed and Poli require more data and will be explained in the following subsections.

6.3.1 Cold



The following is an example of data block defining a generic cold formed section. The cross-section image also follows.

```
25 1

1 0 2.000000e+000 0.000000e+000 0.000000e+000 1.100000e+001 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000

2 0 2.000000e+000 1.100000e+001 0.000000e+000 1.400000e+001 3.000000e+000 1.100000e+001 3.000000e+000 1.570796e+000 3.000000e+000

1 0 2.000000e+000 1.400000e+001 3.000000e+000 1.400000e+001 2.660053e+001 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000

2 0 2.000000e+000 1.400000e+001 2.660053e+001 1.165916e+001 2.952722e+001 1.100000e+001 2.660053e+001 1.349267e+000 3.000000e+000

1 0 2.000000e+000 1.165916e+001 2.952722e+001 -1.695916e+001 3.597279e+001 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000

2 0 2.000000e+000 -1.695916e+001 3.597279e+001 -1.930000e+001 3.889947e+001 -1.630000e+001 3.889947e+001 -1.349267e+000 3.000000e+000

1 0 2.000000e+000 -1.930000e+001 3.889947e+001 -1.930000e+001 1.075000e+002 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000

2 0 2.000000e+000 -1.930000e+001 1.075000e+002 -1.630000e+001 1.105000e+002 -1.630000e+001 1.075000e+002 -1.570796e+000 3.000000e+000

1 0 2.000000e+000 -1.630000e+001 1.105000e+002 1.100000e+001 1.105000e+002 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000

2 0 2.000000e+000 1.100000e+001 1.105000e+002 1.400000e+001 1.075000e+002 1.100000e+001 1.075000e+002 -1.570796e+000 3.000000e+000

1 0 2.000000e+000 1.400000e+001 1.075000e+002 1.400000e+001 1.035000e+002 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000

2 0 2.000000e+000 1.400000e+001 1.035000e+002 1.700000e+001 1.005000e+002 1.700000e+001 1.035000e+002 1.570796e+000 3.000000e+000

1 0 2.000000e+000 1.700000e+001 1.005000e+002 6.040000e+001 1.005000e+002 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000

2 0 2.000000e+000 6.040000e+001 1.005000e+002 6.340000e+001 1.035000e+002 6.040000e+001 1.035000e+002 1.570796e+000 3.000000e+000

1 0 2.000000e+000 6.340000e+001 1.035000e+002 6.340000e+001 1.075000e+002 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000

2 0 2.000000e+000 6.340000e+001 1.075000e+002 6.640000e+001 1.105000e+002 6.640000e+001 1.075000e+002 -1.570796e+000 3.000000e+000

1 0 2.000000e+000 6.640000e+001 1.105000e+002 9.370000e+001 1.105000e+002 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000

2 0 2.000000e+000 9.370000e+001 1.105000e+002 9.670000e+001 1.075000e+002 9.370000e+001 1.075000e+002 -1.570796e+000 3.000000e+000

1 0 2.000000e+000 9.670000e+001 1.075000e+002 9.670000e+001 3.889947e+001 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000

2 0 2.000000e+000 9.670000e+001 3.889947e+001 9.435917e+001 3.597279e+001 9.370000e+001 3.889947e+001 -1.349267e+000 3.000000e+000

1 0 2.000000e+000 9.435917e+001 3.597279e+001 6.574084e+001 2.952722e+001 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000

2 0 2.000000e+000 6.574084e+001 2.952722e+001 6.340001e+001 2.660053e+001 6.640001e+001 2.660053e+001 1.349267e+000 3.000000e+000

1 0 2.000000e+000 6.340001e+001 2.660053e+001 6.340001e+001 3.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000

2 0 2.000000e+000 6.340001e+001 3.000000e+000 6.640001e+001 0.000000e+000 6.640001e+001 3.000000e+000 1.570796e+000 3.000000e+000

1 0 2.000000e+000 6.640001e+001 0.000000e+000 7.740001e+001 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000 0.000000e+000
```



The first field is:

NUMBER_OF_SIDES END_CODE

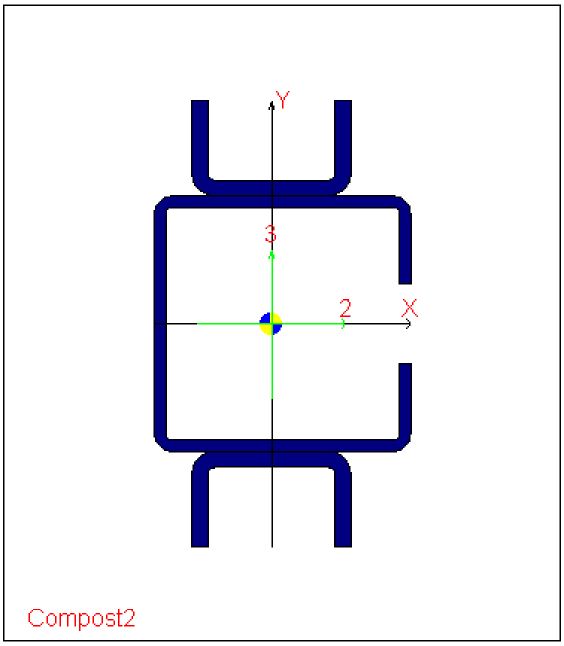
Where end code may be 1 (for open cross-section) or 2 (for closed cross-section).

Then for each side a row of data must be input, with the following format:

SIDE_KIND	1 for straight, 2 for circular
HOLE_FLAG	0 is not an hole, 1 it is a hole
THICKNESS	thickness of side
X1	x start
Y1	y start
X2	x end
Y2	yend
XCENTER	x of center for circular sides, 0 for straight sides
YCENTER	y of center for circular sides, 0 for straight sides
BETA	opening angle of side (in degrees) for circular side, 0 for straight side
RADIUS	mid-thickness radius for circular sides, 0 for straight sides

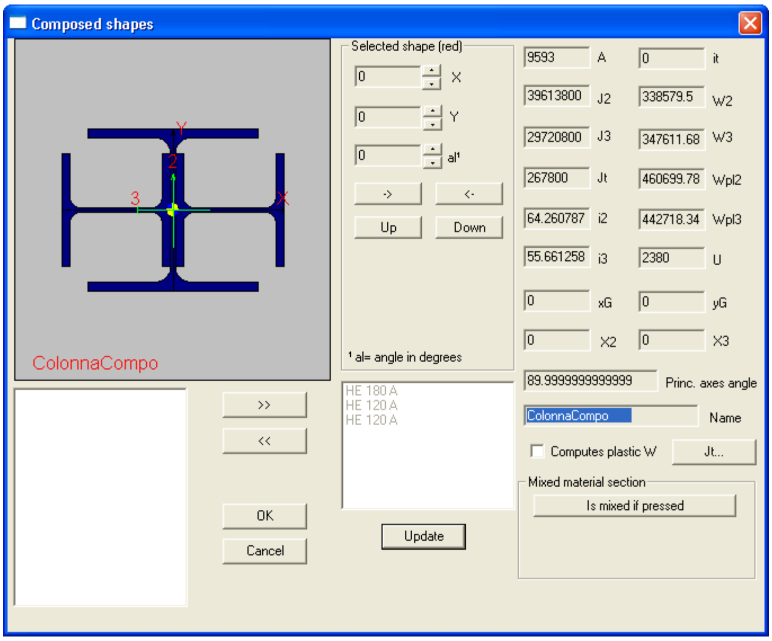
Sides must be continuous with their first derivative. No cusps are allowed. Thickness of all sides must be the same.

6.3.2 Composed



The following is an example of data block defining a generic composed section. The cross-section image also follows.

```
3
   1   1   0.000000e+000   0.000000e+000  0.000000e+000  "HE180A"
    171.000000 180.000000 6.000000 9.500000 15.000000
   2   1   -6.000000e+001   0.000000e+000  9.000000e+001  "HE120A"
    114.000000 120.000000 5.000000 8.000000 12.000000
   3   1   6.000000e+001   0.000000e+000  9.000000e+001  "HE120A"
    114.000000 120.000000 5.000000 8.000000 12.000000
```



The first row is the number of simple cross-section that will be joined together (3 in the example).

TOTAL_SIMPLE_CROSS_SECTIONS_NUMBER

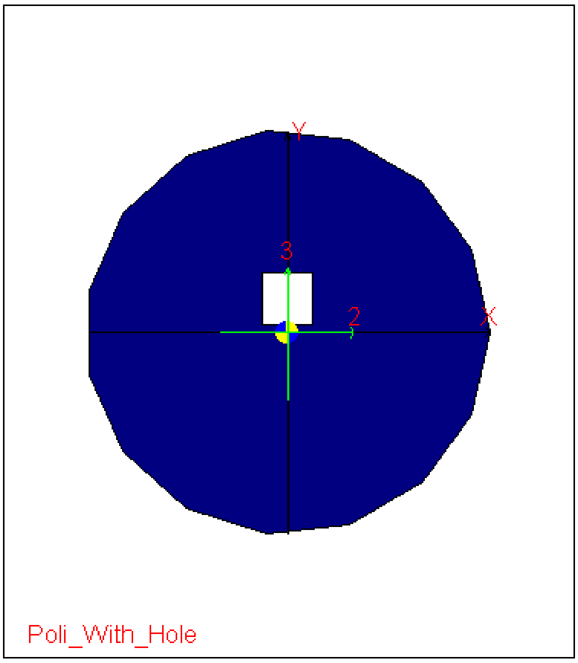
The, for each of these cross sections there are 2 rows. A first row with the following data:

SIMPLE_CROSS_SECTION_NUMBER	the simple cross section number
SIMPLE_CROSS_SECTION_KIND	the kind number according to previously explained coding
SIMPLE_CROSS_SECTION_X	the x position of the center of this cross section in plane XY
SIMPLE_CROSS_SECTION_Y	the y position of the center of this cross section in plane XY
SIMPLE_CROSS_SECTION_ANGLE	the angle in degrees of axis 2 of this cross section in plane XY, to axis X (positive if counter clock wise)
SIMPLE_CROSS_SECTION_NAME	the name of the simple cross section between quotation marks

Next row is a row of data depending on the simple cross section kind. It is just as the rows previously explained for standard cross-sections. See **{data depending on the cross section kind}** above**.**

If the SIMPLE_CROSS_SECTION_KIND is 0, then the SIMPLE_CROSS_SECTION_NAME will be used to detect correct cross-section in file “WSR_SR4_SHPCVT.TXT”. No more data will habe to be read for that SIMPLE cross section.

6.3.3 Poli



First row is:

NPOLY number of polygons.

Then for each polygon there are the following rows:

ROW 1:

POLY_CODE POLY_NPOINTS

POLY_CODE:	0	if the polygon is a hole
	1	if the polygon is not a hole
POLY_NPOINTS	the number of points in the polygon (3 for a triangle)	

ROW 2 to POLY_NPOINTS +1

X POINT_1 Y POINT_1

X POINT_2 Y POINT_2

……………………….

X POINTPOLY_NPOINTS **Y POINT_POLY_NPOINTS**

X and y of the current point

6.4 WORK PROCESSES

In the following, both members and cleats may have "work processes" applied to them. As the description of the work processes is quite the same for all objects, we are going to describe it here. In the following sections, the WORPROCESS data block will be synthetically listed as **WORKPROCESSES**.

The WORKPROCESSES block is conceived in this way:

NWORK_PROCESSES

{WORK_PROCESS_TAG

DATA}

Repeated NWORK_PROCESS times.

Here is an example with 3 work processes.

```
    3   ; number of work processes
BEVEL TRIANGULAR
      15.0000       15.0000; dx  dy
  1.50000000e+002  1.50000000e+002  0.00000000e+000 ; point 1
  1.50000000e+002  1.50000000e+002  3.00000000e+001 ; point 2
BEVEL RECTANGULAR
      15.0000       30.0000; dx  dy
 -1.50000000e+002  1.50000000e+002  0.00000000e+000 ; point 1
 -1.50000000e+002  1.50000000e+002  3.00000000e+001 ; point 2
CUTBYBOX
  0.00000000e+000  0.00000000e+000  1.00000000e+000 ; view vector
     4        15.0000 ; npoints  radius
     0     -150.0000     -150.0000; isbevel  xi   yi
```

```
    0        0.4969     -150.0000; isbevel  xi   yi
    1        0.4969       -0.8994; isbevel  xi   yi
    0     -150.0000       -0.8994; isbevel  xi   yi
```

In the following subsections all the available work processes will be described.

6.4.1 Triangular bevel

These are the cards:

BEVEL TRIANGULAR

SIZEA SIZEB

PX1 PX2 PX3

QX1 QX2 Qx3

SIZEA	base of triangle
SIZEB	height of triangle
PX1,PX2,PX3	coordinates of first point (P) in local (1, 2, 3) cs.
QX1,QX2,QX3	coordinates of second point (Q) in local (1, 2, 3) cs.

```
BEVEL TRIANGULAR
      15.0000      15.0000; dx  dy
  1.50000000e+002  1.50000000e+002  0.00000000e+000 ; point 1
  1.50000000e+002  1.50000000e+002  3.00000000e+001 ; point 2
```

6.4.2 Rectangular bevel

These are the cards:

BEVEL RECTANGULAR

SIZEA SIZEB

PX1 PX2 PX3

QX1 QX2 Qx3

SIZEA	base of rectangle
SIZEB	height of rectangle
PX1,PX2,PX3	coordinates of first point (P) in local (1, 2, 3) cs.
QX1,QX2,QX3	coordinates of second point (Q) in local (1, 2, 3) cs.

```
BEVEL RECTANGULAR
      15.0000      15.0000; dx  dy
 -1.50000000e+002  1.50000000e+002  3.00000000e+001 ; point 1
 -1.50000000e+002  1.50000000e+002  0.00000000e+000 ; point 2
```

6.4.3 Circular Bevel

These are the cards:

BEVEL CIRCULAR

RADIUS

PX1 PX2 PX3

QX1 QX2 Qx3

RADIUS	radius of circular bevel
PX1,PX2,PX3	coordinates of first point (P) in local (1, 2, 3) cs.
QX1,QX2,QX3	coordinates of second point (Q) in local (1, 2, 3) cs.

```
BEVEL CIRCULAR
      15.0000; radius
  1.50000000e+002  1.50000000e+002  3.00000000e+001 ; point 1
  1.50000000e+002  1.50000000e+002  0.00000000e+000 ; point 2
```

6.4.4 Face Rotation

These are the cards:

ROTATE FACE

MODE TNX1 TNX2 TNX3

CNX1 CNX2 CNX3

PX1 PX2 PX3

MODE	if 0 face is extended, if 1 face is left unchanged
TNX1 TNX2 TNX3	Target normal vector components for the face (after rotation).
CNX1 CNX2 CNX3	Current normal vector components for the face (before rotation).
PX1 PX2 PX3	Coordinates of a point of the face before rotation, (1, 2, 3) local cs.

```
ROTATE FACE
   0   9.23879533e-001  0.00000000e+000  3.82683432e-001 ; mode  target
vect
  1.00000000e+000  0.00000000e+000  0.00000000e+000 ; NORMAL
  1.50000000e+002 -1.50000000e+002  0.00000000e+000 ; point
```

6.4.5 Face Translation

These are the cards:

SHIFT FACE

SHIFT

CNX1 CNX2 CNX3

PX1 PX2 PX3

SHIFT	Shift to be applied, positive if according to outward normal of face
CNX1 CNX2 CNX3	Current normal vector components for the face (before rotation).
PX1 PX2 PX3	Coordinates of a point of the face before rotation, (1, 2, 3) local cs.

```
SHIFT FACE
      27.0000 ; shift
```

```
  0.00000000e+000  1.00000000e+000  0.00000000e+000 ; NORMAL
  1.56213203e+002  1.50000000e+002  0.00000000e+000 ; point
```

6.4.6 Cut by box

These are the cards:

These are the cards:

CUTBYBOX

VIEWX1 VIEWX2 VIEWX3

NPOINTS RADIUS

{ISBEVELi Ui Vi}

Repeated NPOINTS times

VIEWX1,VIEWX2,VIEWX3	View vector components in local cs (1, 2, 3)
NPOINTS	The number of points of the box: it is always 4
RADIUS	Radius of circular bevel, if applied
ISBEVEL	0 if no bevel in corner, 1 if bevel in corner
Ui	u coordinate of corner of box point “i”, in projected cs
Vi	v coordinate of corner of box point “i”, in projected cs

```
CUTBYBOX
  0.00000000e+000  0.00000000e+000  1.00000000e+000 ; view vector
     4       15.0000 ; npoints  radius
     0     -150.0000     -150.0000; isbevel  xi   yi
     0        0.4969     -150.0000; isbevel  xi   yi
     1        0.4969       -0.8994; isbevel  xi   yi
     0     -150.0000       -0.8994; isbevel  xi   yi
```

6.4.7 Cut by Poly

These are the cards:

CUTBYPOLY

VIEWX1	**VIEWX2**	**VIEWX3**
NPOINTS	**RADIUS**	
{ISBEVELi	**Ui Vi}**	
		Repeated NPOINTS times

VIEWX1,VIEWX2,VIEWX3	View vector components in local cs (1, 2, 3)
NPOINTS	The number of points of the closed polyline
RADIUS	Radius of circular bevel, if applied
ISBEVEL	0 if no bevel in corner, 1 if bevel in corner
Ui	u coordinate of corner of polyline point “i”, in projected cs
Vi	v coordinate of corner of polyline point “i”, in projected cs

```
CUTBYPOLY
  0.00000000e+000  6.12323400e-017 -1.00000000e+000 ; view vector
     5        15.0000 ; npoints  radius
     0       150.0000     -150.0000; isbevel  xi   yi
     0       150.0000      -80.0000; isbevel  xi   yi
     0        -0.0684       -0.2832; isbevel  xi   yi
     0       -45.0000     -150.0000; isbevel  xi   yi
     0       160.0000     -174.8000; isbevel  xi   yi
```

6.4.8 Cut by plane

These are the cards:

CUTBYPLANE

A B C D

A, B, C, D Define the plane in object local system $Ax_1+Bx_2+Cx_3+D=0$

```
CUTBYPLANE
  1.00  0.00  1.00 235.9; plane
```

6.4.9 Boolean subtraction

Boolean subtraction to an object A is defined by describing an object B as a B-rep, i.e. a collection of faces. The final result will be (A-B).

The faces of B must be ordered in such a way that normal is pointing outward from the object itself. The normal to a face is a direction normal to the plane and such that the face point circulation is seen as anticlockwise.

Every face has a “meaning”, which is a number. If the object is an extrusion, faces “top” and “bottom” have the reserved meanings 1000 and 999 respectively. Otherwise face meaning, which must be unique, could be the face number.

These are the cards:

BOOLEAN SUBTRACTION

NPOINTS NFACES

{POINTNi X1i X2i X3i}

Repeated NPOINTS times

………………………………………………..

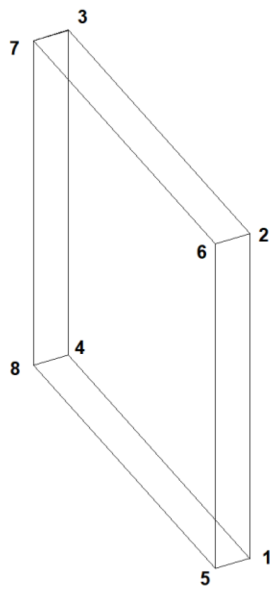
FACEi MEANING_FACEi NPOINT_FACEi

{POINTj}

Repeated NPOINT_FACEi times

…………………………………………………

Repeated NFACES times



For example, the object B sketched in the image above (a 100x100 30 mm extrusion in X_1 direction), may be used to cut a part from some object A. Assuming that the coordinates of the object B are expressed in the local system of object A, the following cards should be input:

```
BOOLEAN SUBTRACTION
  8   6; npoints  nfaces
   1   1000    50     20
   2   1000    50     120
   3   1000    -50     120
   4   1000    -50     20
   5   1030    50     20
   6   1030     50     120
   7   1030    -50     120
   8   1030    -50     20
1  1  4;   face face_meaning  npoint_face (right)
2
1
4
3
2  2  4;   face face_meaning npoint_face (left)
6
7
```

```
8
5
3  3  4;   face face_meaning npoint_face (front)
1
2
6
5
4  4  4;   face face_meaning  npoint_face  (back)
7
3
4
8
5  999  4;   face face_meaning  npoint_face  (top)
2
3
7
6
6  998   4;   face face_meaning npoint_face (bottom)
1
5
8
4
```

6.5 MEMBER COLLECTION

NEWMEMBER MODE0

"INTERNAL_NAME"   "EXTERNAL_NAME"

XP   YP   ZP

XM YM ZM

v1x v1y v1z

v2x v2y v2z

v3x v3y v3z

XE1 YE1 ZE1

XE2 YE2 ZE2

SECT1 SECT2

ELONG1 ELONG2

MATNUM

WORKPROCESSES

INTERNAL_NAME	Name of the object inside CSE
EXTERNAL_NAME	Name of the object outside CSE
XP	global X coordinate of position point (usually E1X, see below)
YP	global Y coordinate of position point (usually E1Y, see below)
ZP	global Z coordinate of position point (usually E1Z, see below)
XM	global X shift from position point
YM	global Y shift from position point
ZM	global Z shift from position point
v1x, v1y, v1z	vector components of local axis 1 (principal axis of cross section)
v2x, v2y, v2z	vector components of local axis 2 (2nd principal axis of cross-section)
v3x, v3y, v3z	vector components of local axis 3 (axis from 1st to 2nd extremity, i.e. **E2-E1**, see below)
XE1, YE1,ZE1	global coordinates (X, Y, Z) of the first extremity of member **E1**, as in original FEM model. This keeps into account rigid offets applied to beam finite elements, if any. This is the centroid of 1st extremity cross-section.
XE2, YE2,ZE2	global coordinates (X, Y, Z) of the second extremity of member, **E2**, as in original FEM model. This keeps into account rigid offets applied

	to beam finite elements, if any. This is the centroid of 2nd extremity cross-section
SECT1	cross-section number in cross-section vector.
SECT2	equal to "0" if prismatic member. Otherwise, the cross section number in cross-section vector, at second member extremity-
ELONG1	elongation (if positive) or shortening (if negative) of 1st extremity
ELONG2	elongation (if positive) or shortening (if negative) of 2nd extremity
MATNUM	Material number
WORKPROCESSES	The section WORKPROCESSES has already been explained (§6.4)

Here is an example of addition of a member:

```
NEWMEMBER MODE0
"Member 1"   "Unknown" ;   Internal Identifier External Identifier
 0.00000000e+000    0.00000000e+000   0.00000000e+000   ; position
 0.00000000e+000    0.00000000e+000   0.00000000e+000   ; move from
position
 0.00000000e+000    1.00000000e+000   0.00000000e+000   ; axis 1
-1.00000000e+000    0.00000000e+000   0.00000000e+000   ; axis 2
 0.00000000e+000    0.00000000e+000   1.00000000e+000   ; axis 3
 0.00000000e+000    0.00000000e+000   0.00000000e+000   ; original p1
 0.00000000e+000    0.00000000e+000   2.50000000e+003   ; original p2
1   0                      ; sect1  sect2
      0.0000          0.0000            ; elongation1(+/-)  elongation2(+/-)
    1  ;  material number
    0   ; number of work processes
```

6.6 OBJECT COLLECTION

An object collection is a set of components, which may be:

1. Bolt layouts;

2. Weld layouts;
3. Plate components;
4. Composed plate components;
5. Cross section trunks;
6. Angles

Objects can be input with each possible sequence, that is, they must not be grouped by kind.

In the next subsections, each component data set will be explained in detail.

6.6.1 BOLTLAYOUT OBJECTS

6.6.1.1 General

```
NEW BOLTLAYOUT MODE0
"B1"   "" ;   Internal Identifier External Identifier
 1.44316515e-014   -4.50000000e+000   0.00000000e+000   ; position
 0.00000000e+000    0.00000000e+000   0.00000000e+000   ; move from position
 1.00000000e+000    0.00000000e+000   0.00000000e+000   ; axis 1
 0.00000000e+000    1.00000000e+000   0.00000000e+000   ; axis 2
 0.00000000e+000    0.00000000e+000   1.00000000e+000   ; axis 3
  0    6  1       22.0000 0       0.0000; boltset  boltclass isfullreactive  diameter, isprecision, extra
 1    4   ;  kind of boltlayout  number of bolts
   2     2      320.0000      320.0000  0 ;  nrows  ncols drows dcols  isemptyinside
 0.00000000e+000  0.00000000e+000        0.0000 ;  odx  ody  blangle
   2       15.0000     600.0000       0.0000       0.0000       0.0000;  nthicks thick1 th2 th3 th4 th5
            0.0000       0.0000       0.0000       0.0000       0.0000;         th6 th7 th8 th9 th10
   2        0.0000       0.0000       0.0000       0.0000       0.0000;
            0.0000       0.0000       0.0000       0.0000;
```

NEW BOLTLAYOUT MODE0

“NAME_OF_BOLTLAYOUT”   “EXTERNAL_NAME_OF_BL”

XP   YP   ZP

XM   YM   ZM

v1x   v1y   v1z

v2x   v2y   v2z

v3x v3y v3z

BOLTSET BOLTCLASS ISFULL DIAM PRECISION EXTRA

KIND NBOLT

DATABOLT

OD1 OD2 BLANGLE

NTHICKS TH1 TH2 TH3 TH4 TH5

TH6 TH7 TH8 TH9 TH10

NTHICKS AIR1-2 AIR2-3 AIR3-4 AIR4-5 AIR5-6

AIR6-7 AIR7-8 AIR8-9 AIR9-10

NAME_OF_BOLTLAYOUT	the name of the bolt layout like “B1”, “B2”, “AA.B1”. Cannot be null
EXTERNAL_NAME_OF_BL	the name of the bolt layout in external program, can be null: “”
XP	global X coordinate of position point
YP	global Y coordinate of position point
ZP	global Z coordinate of position point
XM	global X shift from position point
YM	global Y shift from position point
ZM	global Z shift from position point
v1x, v1y, v1z	vector components of local axis 1
v2x, v2y, v2z	vector components of local axis 2
v3x, v3y, v3z	vector components of local axis 3
BOLSET	boltset is identified by a number, **see section BOLTSET below**
BOLTCLASS	bolt class is identified by a number, **see section BOLTCLASS below**
ISFULL	1 if shear is carried by gross section, 0 if shear is carried by threaded part of shaft
DIAM	Diameter of the bolt shaft in mm, **see section DIAMETER below**
PRECISION	1 if holes are precision holes, 0 if not

EXTRA	if PRECISION is 0, then EXTRA is the difference between hole diameter and shaft diameter.
KIND	kind of the bolt layout: 0 regular; 1 staggered; 2 circular; 3 free set
NBOLT	number of bolts
DATABOLT	**see section DATABOLT below**
OD1	offset of bolt layout center in local 1 direction
OD2	offset of bolt layout center in local 2 direction
BLANGLE	angle of bolt layout grid axes gx and gy (degrees, positive counterclockwise). It's angle α. This value is not used by free set bolt layouts (see below).
NTHICKS	number of different thickness drilled by each bolt. No air is allowed.
TH1	first thickness from head to nut, millimeters
TH2-TH10	second….tenth thickness from head to nut, millimeters
AIR1-2	air stratum in mm, between 1^{st} and 2^{nd} thickness drilled
AIRi-i+1	air stratum in mm, between i^{th} and $(i+1)^{th}$ thickness drilled up to AIR9-10

6.6.1.2 DATABOLT

This block of data depends on bolt layout kind.

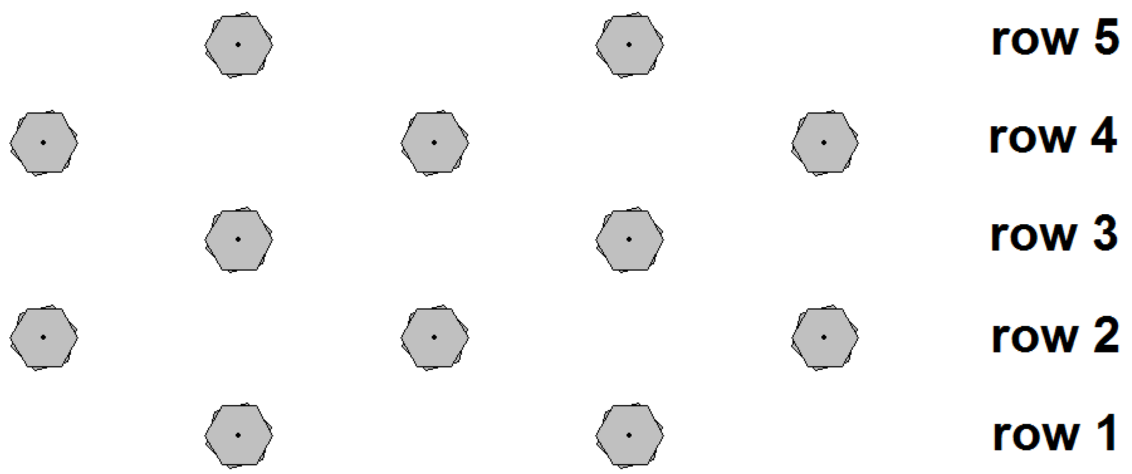
If KIND=1 the bolt layout is a simple regular rectangular grid

Then DATABOLT is:

NROWS NCOLS DROWS DCOLS ISEMPTYINSIDE

NROWS	number of rows
NCOLS	number of columns
DROWS	distance between rows
DCOLS	distance between columns
ISEMPTYINSIDE	if 1 only the perimeter bolts will be added

If KIND = 2 the bolt layout is a simple rectangular grid but with staggered holes

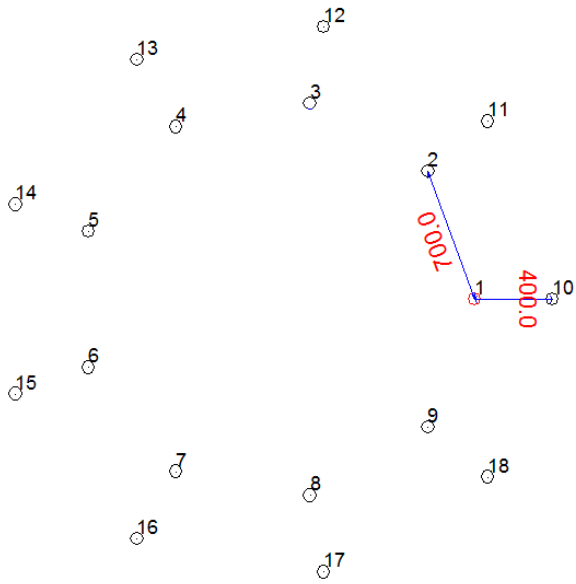


Then DATABOLT is:

NROWS **NCOLS** **DROWS** **DCOLS** **ISEMPTYINSIDE**

NROWS	number of rows
NCOLS	number of columns
DROWS	distance beteween rows
DCOLS	distance between columns
ISEMPTYINSIDE	presently dummy

If KIND = 3 the bolt layout is a circular grid



Nrows=2, Ncolumns = 9, Drow=400, Dcol = 700

Then DATABOLT is:

NROWS NCOLS DROWS DCOLS ISEMPTYINSIDE

NROWS	number of “rows”, i.e. number of circles
NCOLS	number of “columns”, i.e. number of bolts in each circle
DROWS	distance between circles (difference in radii, constant from circle to circle)
DCOLS	distance between bolts in internal circle (inner one). Outer circles will be scaled accordingly. The radius of first circle, i.e. first row is:

$$r_{\min} = r_{1strow} = 0.5\frac{DCOLS}{\sin(\pi / ncol)}$$

ISEMPTYINSIDE	presently dummy

If KIND = 4 the bolt layout is free set, the bolts will be identified by their local coordinates

Then DATABOLT is:

{IBOLT XBOLT YBOLT}

NBOLT TIMES

IBOLT	progressive number of current bolt

XBOLT	x bolt coordinate relative to center
YBOLT	y bolt coordinate relative to center

6.6.1.3 BOLTSET

Referring to BOLTSET data, the following values are applicable.

BOLTSET VALUE	
0	EURO HEXAGON
1	INDIAN HEXAGON
2	AISC (USA)
3	HSFB EURO
4	HSFB INDIAN
5	HSFB AISC (USA)
6	EURO PIN
7	INDIAN PIN
8	AISC PIN

Depending on the value of BOLTSET, some value of bolt class and bolt diameter are available. Bolt class is identified by a number. Bolt diameter by its value in mm. However, the diameter specified must be one of the available ones. HSFB stands for “High Resistance Friction Bolts”.

6.6.1.4 BOLTCLASS

In the next tables, the available bolt class values can be found. For each bolt and pin set, a table is provided. The values to be written in the .D3O files can be found in the third column of each table.

BOLTSET 0 **EURO HEX**

BOLT CLASSES EURO HEX	CLASS	VALUE
EBOLT_CLASS_46	4.6	**0**
EBOLT_CLASS_48	4.8	**1**
EBOLT_CLASS_56	5.6	**2**
EBOLT_CLASS_58	5.8	**3**
EBOLT_CLASS_66	6.6	**4**
EBOLT_CLASS_68	6.8	**5**
EBOLT_CLASS_88	8.8	**6**
EBOLT_CLASS_109	10.9	**7**

EBOLT_CLASS_129	12.9	**8**
EBOLT_CLASS_S235	S235	**9**
EBOLT_CLASS_S275	S275	**10**
EBOLT_CLASS_S355	S355	**11**
EBOLT_CLASS_S450	S450	**12**
EBOLT_CLASS_S235JR	S235JR	**13**
EBOLT_CLASS_S235J0	S235J0	**14**
EBOLT_CLASS_S235J2	S235J2	**15**
EBOLT_CLASS_S275JR	S275JR	**16**
EBOLT_CLASS_S275J0	S275J0	**17**
EBOLT_CLASS_S275J2	S275J2	**18**
EBOLT_CLASS_S355JR	S355JR	**19**
EBOLT_CLASS_S355J0	S355J0	**20**
EBOLT_CLASS_S355J2	S355J2	**21**
EBOLT_CLASS_S355K2	S355K2	**22**
EBOLT_CLASS_S450J0	S450J0	**23**

Available bolt classes for EURO HEX bolts.

BOLTSET 1 **INDIAN HEX**

BOLT CLASSES INDIAN HEX	CLASS	VALUE
IBOLT_CLASS_36	3.6	**0**
IBOLT_CLASS_46	4.6	**1**
IBOLT_CLASS_48	4.8	**2**
IBOLT_CLASS_56	5.6	**3**
IBOLT_CLASS_58	5.8	**4**
IBOLT_CLASS_68	6.8	**5**
IBOLT_CLASS_88	8.8	**6**
IBOLT_CLASS_98	9.8	**7**
IBOLT_CLASS_109	10.9	**8**
IBOLT_CLASS_129	12.9	**9**
IBOLT_CLASS_E165	E165	**10**
IBOLT_CLASS_E250	E250	**11**
IBOLT_CLASS_E300	E300	**12**
IBOLT_CLASS_E350	E350	**13**
IBOLT_CLASS_E410	E410	**14**
IBOLT_CLASS_E450D	E450D	**15**
IBOLT_CLASS_E450E	E450E	**16**

Available bolt classes for INDIAN HEX bolts.

BOLTSET 2 **AISC HEX**

BOLT CLASSES	CLASS	VALUE

AISC HEX		
ABOLT_CLASS_A307	A307	0
ABOLT_CLASS_A325T1	A325T1	1
ABOLT_CLASS_A325T2	A325T2	2
ABOLT_CLASS_A325T3	A325T3	3
ABOLT_CLASS_A490	A490	4

Available bolt classes for AISC HEX bolts.

BOLTSET 3 **EURO HSFB**

BOLT CLASSES EURO HSFB	CLASS	VALUE
E2BOLT_CLASS_88	8.8	0
E2BOLT_CLASS_109	10.9	1
E2BOLT_CLASS_129	12.9	2

Available bolt classes for EURO HSFB bolts.

BOLTSET 4 **INDIAN HSFB**

BOLT CLASSES INDIAN HSFB	CLASS	VALUE
I2BOLT_CLASS_88	8.8	0
I2BOLT_CLASS_109	10.9	1
I2BOLT_CLASS_129	12.9	2

Available bolt classes for INDIAN HSFB bolts.

BOLTSET 5 **AISC HSFB**

BOLT CLASSES AISC HSFB	CLASS	VALUE
A2BOLT_CLASS_A325T1	A325T1	0
A2BOLT_CLASS_A325T2	A325T2	1
A2BOLT_CLASS_A325T3	A325T3	2
A2BOLT_CLASS_A490	A490	3

Available bolt classes for AISC HSFB bolts.

BOLTSET 6 **EURO PIN**

BOLT CLASSES EURO PIN	CLASS	VALUE
EPIN_CLASS_46	4.6	**0**
EPIN_CLASS_48	4.8	**1**
EPIN_CLASS_56	5.6	**2**
EPIN_CLASS_58	5.8	**3**
EPIN_CLASS_66	6.6	**4**
EPIN_CLASS_68	6.8	**5**
EPIN_CLASS_88	8.8	**6**
EPIN_CLASS_109	10.9	**7**
EPIN_CLASS_129	12.9	**8**
EPIN_CLASS_S235	S235	**9**
EPIN_CLASS_S275	S275	**10**
EPIN_CLASS_S355	S355	**11**
EPIN_CLASS_S450	S450	**12**
EPIN_CLASS_S235JR	S235JR	**13**
EPIN_CLASS_S235J0	S235J0	**14**
EPIN_CLASS_S235J2	S235J2	**15**
EPIN_CLASS_S275JR	S275JR	**16**
EPIN_CLASS_S275J0	S275J0	**17**
EPIN_CLASS_S275J2	S275J2	**18**
EPIN_CLASS_S355JR	S355JR	**19**
EPIN_CLASS_S355J0	S355J0	**20**
EPIN_CLASS_S355J2	S355J2	**21**
EPIN_CLASS_S355K2	S355K2	**22**
EPIN_CLASS_S450J0	S450J0	**23**

Available bolt classes for EURO pins.

BOLTSET 7 **INDIAN PIN**

BOLT CLASSES INDIAN PIN	CLASS	VALUE
IPIN_CLASS_46	4.6	**0**
IPIN_CLASS_48	4.8	**1**
IPIN_CLASS_56	5.6	**2**
IPIN_CLASS_58	5.8	**3**
IPIN_CLASS_66	6.6	**4**
IPIN_CLASS_68	6.8	**5**
IPIN_CLASS_88	8.8	**6**
IPIN_CLASS_109	10.9	**7**
IPIN_CLASS_129	12.9	**8**
IPIN_CLASS_S235	S235	**9**
IPIN_CLASS_S275	S275	**10**
IPIN_CLASS_S355	S355	**11**
IPIN_CLASS_S450	S450	**12**

Available bolt classes for INDIAN pins.

BOLTSET 8 **AISC PIN**

BOLT CLASSES AISC PIN	CLASS	VALUE
APIN_CLASS_A307	A307	**0**
APIN_CLASS_A325T1	A325T1	**1**
APIN_CLASS_A325T2	A325T2	**2**
APIN_CLASS_A325T3	A325T3	**3**
APIN_CLASS_A490	A490	**4**

Available bolt classes for AISC pins.

6.6.1.5 BOLT DIAMETERS

The value of the available bolt diameters to be written in .D3O file can be found in the second columns of the following tables.

Each table refers to a bolt set, or to a pin set. Values should be input using exactly the values specified, so that the diameter can match on of the values in the vector of diameters.

EURO HEXAGON BOLTS	Value
Diameter **(mm)**	**(mm)**
8	8.0
10	10.0
12	12.0
14	14.0
16	16.0
18	18.0
20	20.0

22	22.0
24	24.0
27	27.0
30	30.0
33	33.0
36	36.0
39	39.0
42	42.0
45	45.0
48	48.0
52	52.0
56	56.0
60	60.0
64	64.0
68	68.0

Available diameters for Euro Hexagon Bolts

INDIAN HEXAGON BOLTS	**Value**
Diameter **(mm)**	**(mm)**
8	8.0
10	10.0
12	12.0
14	14.0
16	16.0
18	18.0

20	20.0
22	22.0
24	24.0
27	27.0
30	30.0
33	33.0
36	36.0
39	39.0
42	42.0
45	45.0
48	48.0
52	52.0
56	56.0
60	60.0
64	64.0
68	68.0

Available diameters for Indian Hexagon Bolts

AISC (USA) HEXAGON BOLTS	**Value**
Diameter **(in)**	**(mm)**
½	12.70
5/8	15.875
¾	19.050
7/8	22.2225
1	25.4

1 1/8	28.575
1 ¼	31.75
1 3/8	34.925
1 ½	38.10

Available diameters for AISC Hexagon Bolts

EURO HSFB BOLTS	**Value**
Diameter **(mm)**	**(mm)**
12	12.0
14	14.0
16	16.0
18	18.0
20	20.0
22	22.0
24	24.0
27	27.0
30	30.0
33	33.0
36	36.0

Available diameters for Euro HSFB Bolts

EURO HSFB BOLTS	**Value**
Diameter **(mm)**	**(mm)**
12	12.0

14	14.0
16	16.0
18	18.0
20	20.0
22	22.0
24	24.0
27	27.0
30	30.0
33	33.0
36	36.0

Available diameters for Euro HSFB Bolts

AISC HSFB BOLTS	**Value**
Diameter **(in)**	**(mm)**
½	12.70
5/8	15.875
¾	19.050
7/8	22.2225
1	25.4
1 1/8	28.575
1 ¼	31.75
1 3/8	34.925
1 ½	38.10

Available diameters for AISC HSFB Bolts

EURO PINS	Value
Diameter (mm)	**(mm)**
5	5.0
6	6.0
8	8.0
10	10.0
12	12.0
14	14.0
16	16.0
18	18.0
20	20.0
22	22.0
24	24.0
27	27.0
30	30.0
33	33.0
36	36.0
40	40.0
45	45.0
50	50.0
55	55.0
60	60.0
70	70.0
80	80.0
90	90.0

100	100.0

Available diameters for Euro Pins

INDIAN PINS	Value
Diameter (mm)	**(mm)**
5	5.0
6	6.0
8	8.0
10	10.0
12	12.0
14	14.0
16	16.0
18	18.0
20	20.0
22	22.0
24	24.0
27	27.0
30	30.0
33	33.0
36	36.0
40	40.0
45	45.0
50	50.0
55	55.0
60	60.0

70	70.0
80	80.0
90	90.0
100	100

Available diameters for Indian Pins

s

AISC PINS	**Value**
Diameter (in)	**(mm)**
1¼	31.75
1½	38.10
1¾	44.45
2	50.80
2¼	57.15
2½	63.50
2¾	69.85
3	76.20
3¼	82.55
3½	88.90
3¾	95.25

Available diameters for AISC Pins

6.6.2 WELDLAYOUT OBJECTS

This is an example of addition of one weld layout:

```
NEW WELDLAYOUT MODE0
"AB.W2"   "" ;   Internal Identifier External Identifier
```

```
1.00000000e+002   -6.59999990e+000   5.00000000e+003   ; position P
0.00000000e+000    0.00000000e+000   0.00000000e+000   ; move from P
0.00000000e+000    1.00000000e+000  -0.00000000e+000   ; axis 1
0.00000000e+000    0.00000000e+000   1.00000000e+000   ; axis 2
1.00000000e+000   -0.00000000e+000  -0.00000000e+000   ; axis 3
  0     2; kind nwelds
  1  4.950  90.000  3.50e+000 -6.25e+001  3.50e+000  6.25e+001;
  2  4.950  90.000 -3.50e+000  6.250e+001 -3.50e+000 -6.25e+001;
```

NEW WELDLAYOUT MODE0

“NAME_OF_WELDLAYOUT”   “EXTERNAL_NAME_OF_WL”

XP   YP   ZP

XM   YM   ZM

v1x   v1y   v1z

v2x   v2y   v2z

v3x   v3y   v3z

KIND   NWELDS

{WELD THICK   ANGLE X1START   X2START   X1END X2END}

NWELDS_TIMES

NAME_OF_WELDLAYOUT	the name of the bolt layout like “B1”, “B2”, “AA.B1”. Cannot be null
EXTERNAL_NAME_OF_WL	the name of the bolt layout in external program, can be null: “”
XP	global X coordinate of position point
YP	global Y coordinate of position point
ZP	global Z coordinate of position point
XM	global X shift from position point
YM	global Y shift from position point

ZM	global Z shift from position point
v1x, v1y, v1z	vector components of local axis 1
v2x, v2y, v2z	vector components of local axis 2
v3x, v3y, v3z	vector components of local axis 3
KIND	The kind of the weld layout: 0 fillet; 1 penetration
NWELDS	Number of weld seams
WELD	Weld number, from 1 to NWELDS
THICK	Thickness of the weld
ANGLE	Angle between active faces, in degrees
X1START	x1 coordinate of first point of weld
X2START	x2 coordinate of the first point of weld
X1END	x1 coordinate of the second and last point of weld
X2END	x2 coordinate of the second and last point of weld

Positions of welds in local plane (1, 2) (x1, x2) are set by writing the coordinates of the two extremities of the weld seams.

6.6.3 PLATE OBJECTS

Plate objects are defined by the following cards:

NEWCLEAT PLATE MODE0

“INTERNAL_NAME” “EXTERNAL_NAME”

XP YP ZP

XM YM ZM

v1x v1y v1z

v2x v2y v2z

v3x v3y v3z

TYPE THICKNESS

DATA

MATNUM

WORKPROCESSES

INTERNAL_NAME	Name of the object inside CSE
EXTERNAL_NAME	Name of the object outside CSE
XP	global X coordinate of position point
YP	global Y coordinate of position point
ZP	global Z coordinate of position point
XM	global X shift from position point
YM	global Y shift from position point
ZM	global Z shift from position point
v1x, v1y, v1z	vector components of local axis 1
v2x, v2y, v2z	vector components of local axis 2
v3x, v3y, v3z	vector components of local axis 3
TYPE	Type of the plate (see above). Type 5 is for generic plates (closed polylines).
THICKNESS	The thickness of the plate.
DATA	See below, section DATA
MATNUM	Material number
WORKPROCESSES	The section WORKPROCESSES has already been explained (§6.4)

Section DATA

This section is different depending on plate type.

For all types with the exception of type 5 (generic plate), DATA is as follows:

P1	**P2**	**P3**	**P4**	**P5**
P6	**P7**	**P8**	**P9**	**P10**

P1-P10	Parameters identifying the dimensions. 10 parameters must always be input, no matter they are unused. To get the meaning of parameters, see main section **Components-PLATES-Typical Plates** above, where all typical plates are described

Example:

```
    300.0000      300.0      100.0      100.0      100.0  ;  p1  p2  p3  p4 p5

    100.0000      100.0      100.0      100.0      100.0  ;  p6  p7  p8  p9 p10
```

For type 5 (generic plates), DATA is as follows:

NPOINTS_EXTERNAL

{X1i X2i}

Repeated NPOINTS_EXTERNAL TIMES

NPOINTS_INTERNAL

{X1i X2i}

Repeated NPOINTS_INTERNAL TIMES

NPOINTS_EXTERNAL	Number of points of the external closed polyline
X1i X2i	Coordinates of point i of external polyline
NPOINTS_INTERNAL	Number of points of the internal closed polyline (a hole). Can be 0.
X1i X2i	Coordinates of point i of internal polyline

Example:

```
    4  ;  number of points in external poly

-1.75000000e+002  -1.75000000e+002 ;  Point     1 (x1, x2)

 1.25000000e+002  -1.75000000e+002 ;  Point     2 (x1, x2)

 1.75000000e+002   1.75000000e+002 ;  Point     3 (x1, x2)

-7.50000000e+001   1.25000000e+002 ;  Point     4 (x1, x2

    0  ;  number of points in internal poly
```

Here is an example of a PLATE addition, complete with WORKPROCESSES subsection

```
NEWCLEAT PLATE MODE0
"AB.P4"   "" ;   Internal Identifier External Identifier
 1.15000e+003   -6.00000000e+001   5.27000000e+003   ; position
 0.00000e+000    0.00000000e+000   0.00000000e+000   ; move from position
 1.00000e+000    0.00000000e+000   0.00000000e+000   ; axis 1
 0.00000e+000    6.12323400e-017   1.00000000e+000   ; axis 2
 0.00000e+000   -1.00000000e+000   6.12323400e-017   ; axis 3
    8   3.00000000e+001  ;  type  thickness
    300.0000     300.0     100.0     100.0     100.0  ;  p1  p2  p3  p4 p5
    100.0000     100.0     100.0     100.0     100.0  ;  p6  p7  p8  p9 p10
    1  ;  material number
    4   ; number of work processes
BEVEL CIRCULAR
      15.0000; radius
  1.50000000e+002  1.50000000e+002  3.00000000e+001 ; point 1
  1.50000000e+002  1.50000000e+002  0.00000000e+000 ; point 2
BEVEL RECTANGULAR
      15.0000      15.0000; dx  dy
 -1.50000000e+002  1.50000000e+002  3.00000000e+001 ; point 1
 -1.50000000e+002  1.50000000e+002  0.00000000e+000 ; point 2
BEVEL TRIANGULAR
      15.0000      15.0000; dx  dy
  1.50000000e+002 -1.50000000e+002  3.00000000e+001 ; point 1
  1.50000000e+002 -1.50000000e+002  0.00000000e+000 ; point 2
CUTBYPOLY
  0.00000000e+000  6.12323400e-017 -1.00000000e+000 ; view vector
     5       15.0000 ; npoints  radius
     0      150.0000     -150.0000; isbevel  xi   yi
     0      150.0000      -80.0000; isbevel  xi   yi
```

```
     0       -0.0684       -0.2832; isbevel  xi   yi
     0      -45.0000     -150.0000; isbevel  xi   yi
     0      160.0000     -174.8000; isbevel  xi   yi
```

6.6.4 CPLATE OBJECTS

Composed plates objects (CPLATE) are defined by the following cards:

NEWCLEAT CPLATE MODE0

“INTERNAL_NAME” “EXTERNAL_NAME”

XP YP ZP

XM YM ZM

v1x v1y v1z

v2x v2y v2z

v3x v3y v3z

TYPE

P1 P2 P3 P4 P5

P6 P7 p8 P9 P10

MATNUM

WORKPROCESSES

INTERNAL_NAME	Name of the object inside CSE
EXTERNAL_NAME	Name of the object outside CSE
XP	global X coordinate of position point
YP	global Y coordinate of position point
ZP	global Z coordinate of position point
XM	global X shift from position point
YM	global Y shift from position point

ZM	global Z shift from position point
v1x, v1y, v1z	vector components of local axis 1
v2x, v2y, v2z	vector components of local axis 2
v3x, v3y, v3z	vector components of local axis 3
TYPE	Type of the CPLATE (see section 4.6 of this document).
P1-P5	First 5 parameters of CPLATE (see section 4.6 of this document)
P6-P10	Last 5 parameters of CPLATE (see section 4.6 of this document))
MATNUM	Material number
WORKPROCESSES	The section WORKPROCESSES has already been explained (§6.4)

Here is an example of addition of a CPLATE:

```
NEWCLEAT CPLATE MODE0
"AB.C1"   "" ;   Internal Identifier External Identifier
 1.00450000e+003   -6.00000e+001   5.12000000e+003   ; position
 0.00000000e+000    0.00000e+000   0.00000000e+000   ; move from position
 1.00000000e+000    0.00000e+000   0.00000000e+000   ; axis 1
 0.00000000e+000    1.00000e+000   0.00000000e+000   ; axis 2
 0.00000000e+000    0.00000e+000   1.00000000e+000   ; axis 3
  401   ;  type
    200.0     300.0     200.0      15.0       9.0  ;  p1  p2  p3  p4 p5
      0.0       0.0       0.0       0.0       0.0  ;  p6  p7  p8  p9 p10
    1  ;  material number
    0  ; number of work processes
```

6.6.5 TRUNK OBJECTS

Cross-section trunks are defined by the following cards:

NEWCLEAT TRUNK MODE0

“INTERNAL_NAME” “EXTERNAL_NAME”

XP YP ZP

XM YM ZM

v1x v1y v1z

v2x v2y v2z

v3x v3y v3z

LENGTH

CROSS_SECTION_DATA

MATNUM

WORKPROCESSES

INTERNAL_NAME	Name of the object inside CSE
EXTERNAL_NAME	Name of the object outside CSE
XP	global X coordinate of position point
YP	global Y coordinate of position point
ZP	global Z coordinate of position point
XM	global X shift from position point
YM	global Y shift from position point
ZM	global Z shift from position point
v1x, v1y, v1z	vector components of local axis 1
v2x, v2y, v2z	vector components of local axis 2
v3x, v3y, v3z	vector components of local axis 3
LENGTH	length of cross-section trunk (axis 3)
CROSS_SECTION_DATA	see below
MATNUM	Material number
WORKPROCESSES	The section WORKPROCESSES has already been explained (§6.4)

CROSS_SECTION_DATA

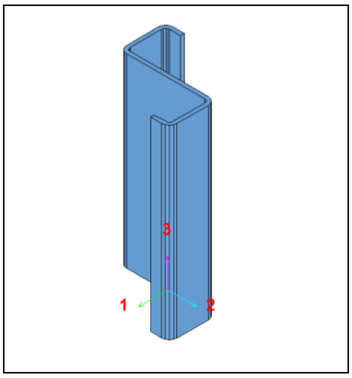
In this section, a single cross-section is described, using the same rules already used in the main section of file .D3O named "CROSS SECTIONS" described above.

So, the following data must be input:

{CROSS_SECTION_NUMBER CROSS_SECTION_KIND "NAME"}

{data depending on cross section kind }

See section 6.3.



An important difference of cross-section trunk, when considered in comparison with members, is the orientation of local axis (1, 2, 3). The principal axes of cross-section are not used for cross-section trunks, they are only used for members (as they are generated by finite elements). However, cross-section trunk origin is in the centroid of the "bottom" cross-section.

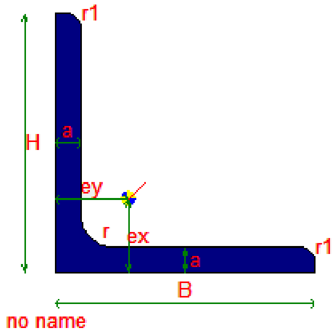
Here is an example of cards referring to a cross-section trunk:

```
NEWCLEAT TRUNK MODE0
"AB.T1"   "" ;   Internal Identifier External Identifier
-5.50000e+001    1.07441489e+003   5.12000000e+003   ; position
 0.00000e+000    0.00000000e+000   0.00000000e+000   ; move from position
 1.00000000e+000    0.00000000e+000   0.00000000e+000   ; axis 1
 0.00000000e+000    1.00000000e+000   0.00000000e+000   ; axis 2
 0.00000000e+000    0.00000000e+000   1.00000000e+000   ; axis 3
    200.0000  ;  length
    1    12   "Tee                    "
100.000000 100.000000 10.000000 15.000000
    1  ;  material number
```

```
    0    ; number of work processes
```

6.6.6 ANGLE OBJECTS

Angles are defined by the following cards:



NEWCLEAT ANGLE MODE0

"INTERNAL_NAME"   "EXTERNAL_NAME"

XP   YP   ZP

XM   YM   ZM

v1x   v1y   v1z

v2x   v2y   v2z

v3x   v3y   v3z

LENGTH

"CROSS_SECTION_NAME"

HDIM   BDIM   ADIM   RDIM   R1DIM

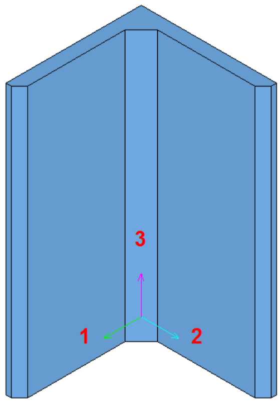
MATNUM

WORKPROCESSES

INTERNAL_NAME	Name of the object inside CSE
EXTERNAL_NAME	Name of the object outside CSE
XP	global X coordinate of position point
YP	global Y coordinate of position point

ZP	global Z coordinate of position point
XM	global X shift from position point
YM	global Y shift from position point
ZM	global Z shift from position point
v1x, v1y, v1z	vector components of local axis 1
v2x, v2y, v2z	vector components of local axis 2
v3x, v3y, v3z	vector components of local axis 3
LENGTH	length of angle (axis 3)
CROSS_SECTION_NAME	the name of the angle cross section
HDIM	height of L (see picture)
BDIM	base of L (see picture)
ADIM	thickness of L (see picture)
RDIM	major radius of L (see picture)
R1DIM	minor radius of L (see picture)
MATNUM	Material number
WORKPROCESSES	The section WORKPROCESSES has already been explained (§6.4)

Angles cannot be input as cross-section trunk. Their origin is not in the center of the cross-section, but in the outer corner, see picture below:



Here is an example of cards referring to an angle.

```
NEWCLEAT ANGLE MODE0
"AB.L1"   "" ;   Internal Identifier External Identifier
 6.000000e+001    1.00000000e+003   5.12000000e+003   ; position
 0.000000e+000    0.00000000e+000   0.00000000e+000   ; move from position
 1.00000000e+000    0.00000000e+000   0.00000000e+000   ; axis 1
 0.00000000e+000    1.00000000e+000   0.00000000e+000   ; axis 2
 0.00000000e+000    0.00000000e+000   1.00000000e+000   ; axis 3
 2.00000000e+002  ;  length
"L 100x10            "
    100.0000     100.0000      10.0000      12.0000       6.0000
    1  ;  material number
    0   ; number of work processes
```

7 Table of Content