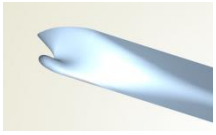
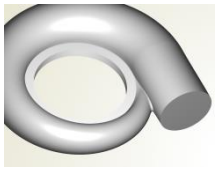
Introduction to Meta Surfaces

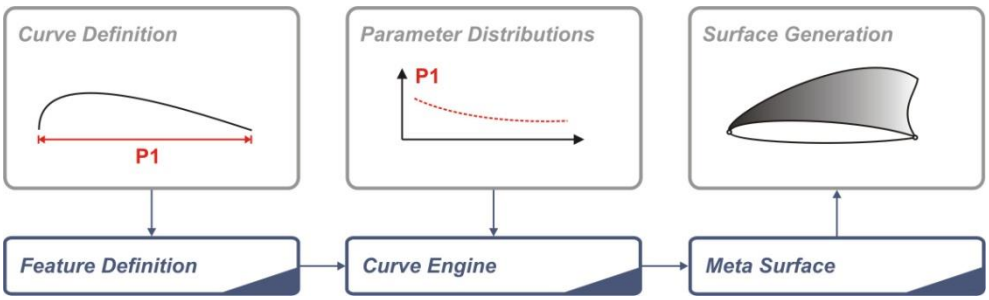




Meta surfaces are commonly used for parametric sweeps along a given path, direction or range. When sweeping any kind of curve definition, the surface shape can be further controlled by multiple curve parameters.

Example applications are volutes, ship hulls, blades, diffusers, wings, ducts and many more. Typically, these kinds of surfaces can be based on a 2D contour or profile description which consists of a set of parameters like radius, length, angle or an area value. Meta surfaces control such key parameters by functions along the sweeping range. Finally, such functions are then varied in design studies and optimization processes.

The process is as follows (see the example wing illustration below): The 2D profile is designed first with a single length parameter called "P1". This curve is then encapsulated in a *feature definition*. In a next step, a function is created that controls the values of "P1" within the sweeping range. The profile and its functions are then connected via so-called *curve engines*. Finally, a meta surface is created which is based on the curve engine.

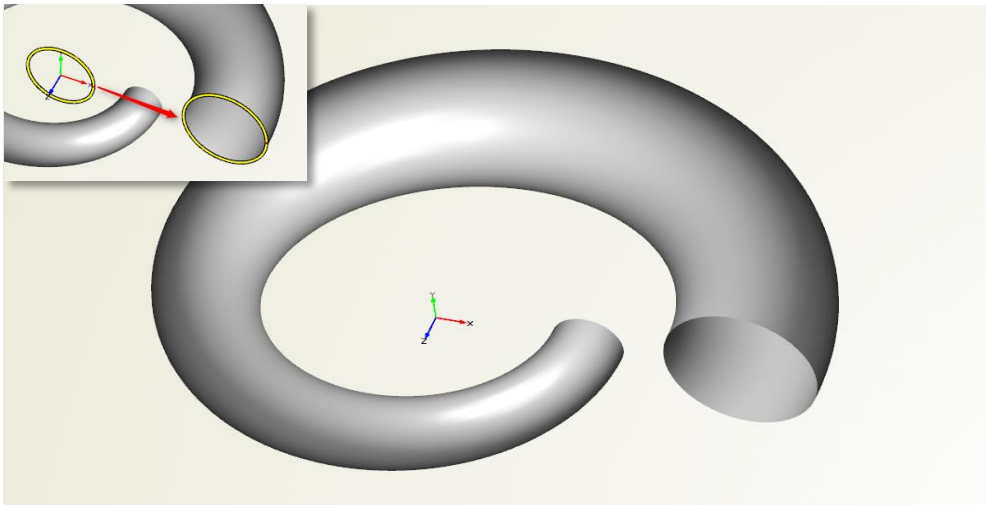


✓ Sweeps along a given 3D path: For basic sweeps along a path, note that there is also a sweep surface available via *CAD > surfaces > sweep surface*. It allows sweeping one or two contours but without controlling the shape while running along the path. Advanced sweeps along a 3D path are realized in combination with the sweep transformation *CAD > transformations > sweep transformation* and the meta surface entity. This transformation accomplishes the alignment and movement along a given 3D path.

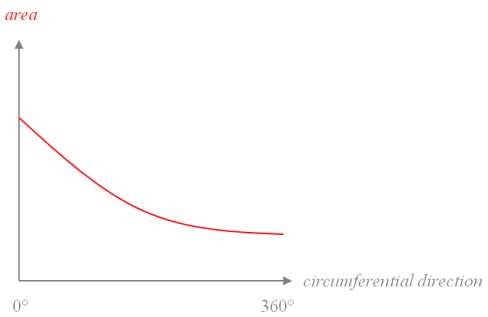
Parametric Spiral

In this tutorial a simple spiral gets modeled. Surface generation is based on a circle (i.e. this will be the curve definition) for which the sectional area is controlled in the circumferential direction. Additionally, the spiral radius is introduced and controlled to receive a typical spiral shape when smoothly decreasing its value.

The single steps of this tutorial refer to the illustration above (first page). Step 1 & 2 focus on the circle definition. The circle has to be moved to the starting position of the upcoming surface using a translation. Secondly, the rotation around the y-axis is defined.



Simple functions are created for the area parameter and the spiral radius to control the shape in the circumferential direction. Some comments on the meta surface attributes are also given so as to understand the surface attributes and finetune the final shape.

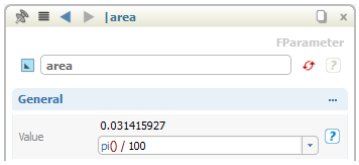


CAESES Project

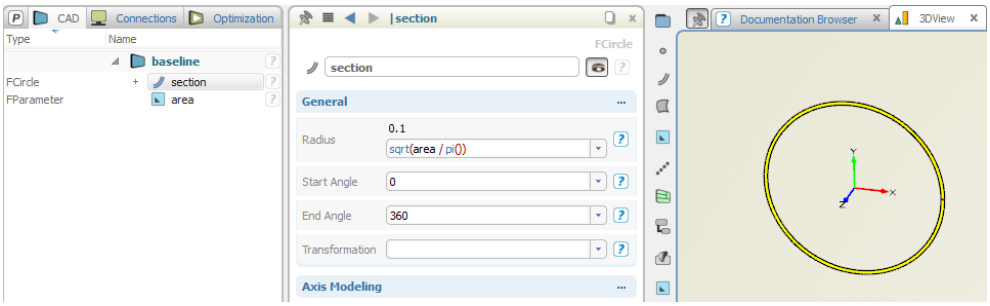
The resulting model can be found in the section *samples* > *tutorials* of the documentation browser.

1 Curve Definition: Section Area

In the first step, a circle is created. The radius will be controlled by a user-defined parameter for the area:



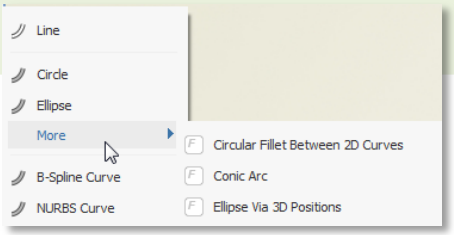
- Create a parameter via *CAD > parameters > parameter* and name it “area”.
- For a start, set the value of “area” to “pi()/100”.
- Create a circle via *CAD > curves > circle* and name it “section”.
- The area of a circle is given by $A = \pi r^2$ so enter the following expression into the circle attribute *radius*: “sqrt(area/pi())”.



You can now change the area parameter and the radius will be set accordingly. Try it out.

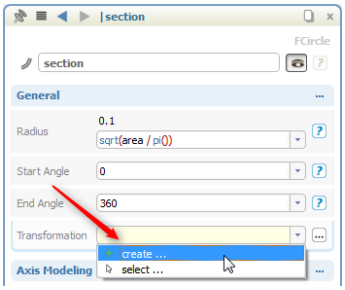
✓ Circles and ellipses are created in the origin of the xy-system by default. For sweeping it around the y-axis, we still have to move it in the x-direction which is given in step 2.

Note that there are other ways to create circles and ellipses, see the *more* menu.

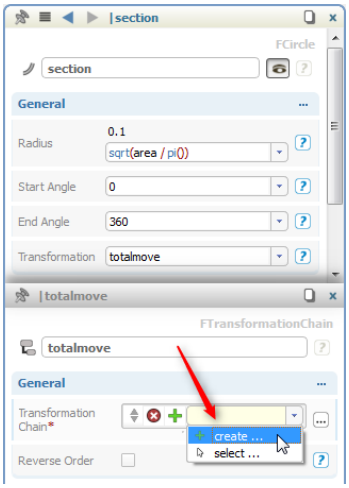


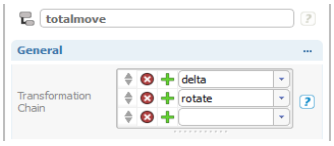
2 Curve Definition: Transformation

The circle will be rotated around the global y-axis during the surface generation process. We need to translate the section to the right place in space to start the sweep. For the total transformation of the section, a *translation* and a *rotation* are required. Both are combined in a *transformation chain*:

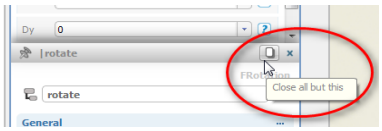


- For "section", create a *transformation chain* via the pull-down menu of the attribute *transformation*.
- Set the name of it to "totalmove".
- From the pull-down menu of the attribute *transformation chain* of "totalmove", create a *translation* and name it "delta".
- Set the *dx*-value of "delta" to "1".
- Again for "totalmove": From the pull-down menu of the attribute *transformation chain* (e.g. click at "+" button and then into an empty editor field), create a *rotation* and name it "rotate".

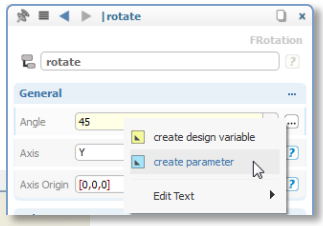


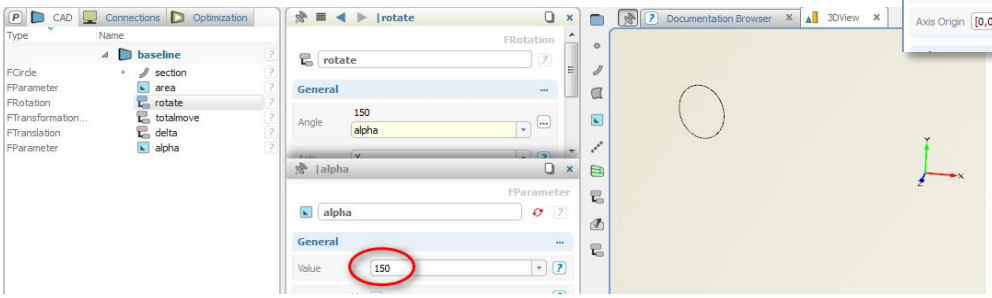


- Set the value of angle to "45".
- Set the rotation attribute *axis* to "Y".
- Close all but "rotate" by clicking at the white button in the upper right corner of "rotate".



- Create a parameter "alpha" for the attribute *angle* via the context menu.
- Change the value in the range 0-360° to see the transformation in the 3D view.

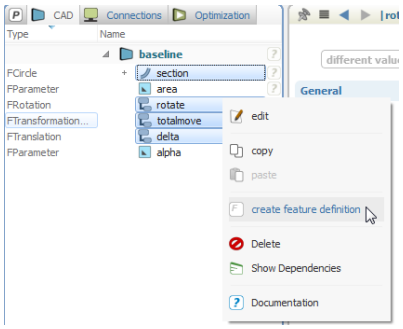




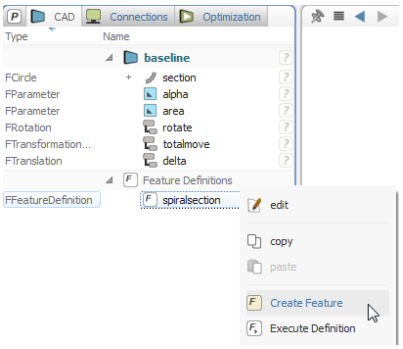
3 Feature Definition

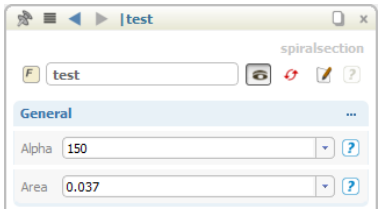
The circle section can be rotated around the y-axis and the area value controls the radius of the circle. Let's put this curve into a feature definition which will be the input object for the upcoming *curve engine*.

- Select all objects in the tree – apart from the parameters "area" and "alpha" (these values are variable so they will be inputs for the curve definition).
- From the context menu, select *create feature definition*.
- Set the *type name* of the definition to "spiralsection" (see the *general* tab).
- Press the *apply* button and close the dialog.



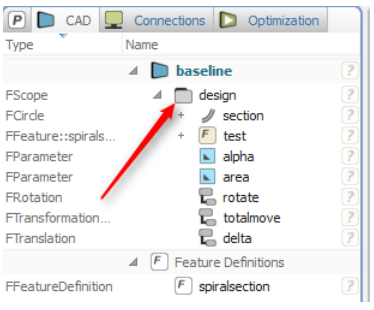
- Create a feature from the new definition by right-clicking and name it "test": Check whether everything works fine and enter some test values for alpha and area.





Now that we have created the feature definition, we no longer need the initial geometry. Let's put it into a scope and hide the geometry for the time being. Note that we could also delete the initial geometry.

- Select all objects and create a scope via CAD > scope.
- Name the scope "design" and set it invisible by clicking at its scope icon in the tree.

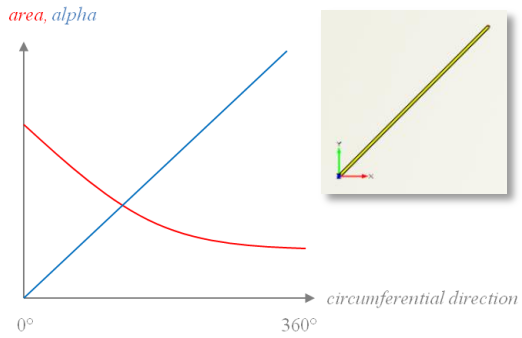


4 Parameter Distributions: Alpha

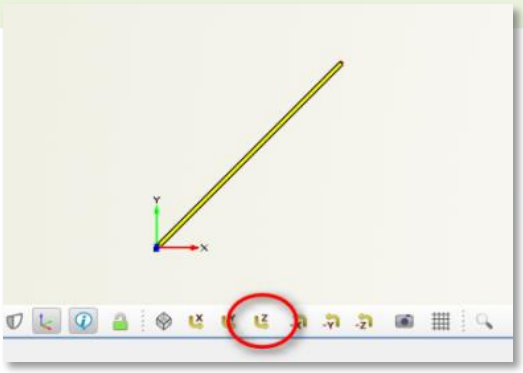
We create parameter distributions for the variable input of the section definition, namely “area” and “alpha”. Remember: The goal is to vary the area in the circumferential direction. Angle “alpha” shall run between 0° and 360°, say, in a linear manner.

For this example, we want to define normalized functions in the xy-system i.e. the x-axis is the abscissa and y-axis the ordinate, both in the interval [0,1]:

- Create a line via *CAD > curves > line* and name it “alphafunction”.
- Set *start position* to “[0,0,0]”.
- Set *end position* to “[1,1,0]”.



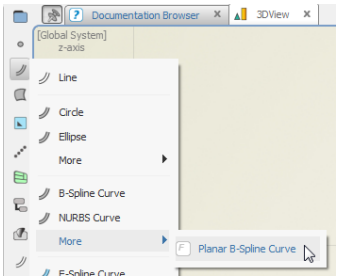
✓ Use the “z”-button at the bottom of the 3D view in order to have an xy-view.



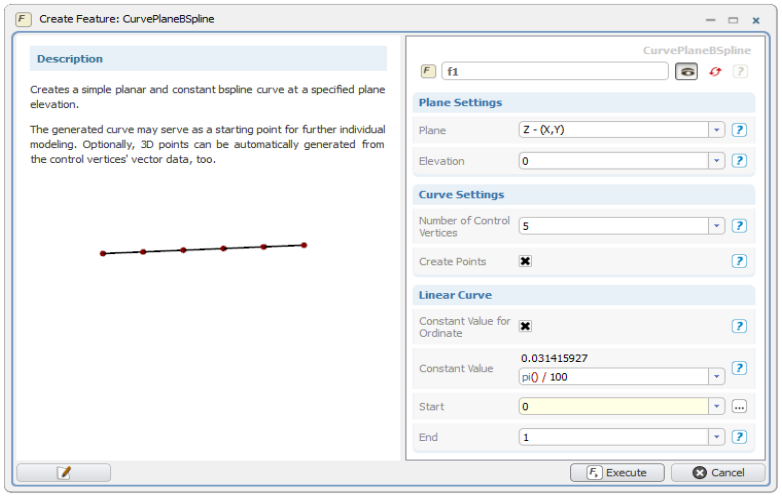
5 Parameter Distributions: Area

For the area distribution we want to have a simple b-spline curve that is defined by a set of 5 points.

- Choose *CAD > curves > more > planar b-spline curve* from the b-spline menu, see the screenshot.
- Set *number of control points* to "5".
- Set *constant* value to "pi()/100" again for a reasonable initial constant function.



Note that there is a preview in the 3D view of your entered values.



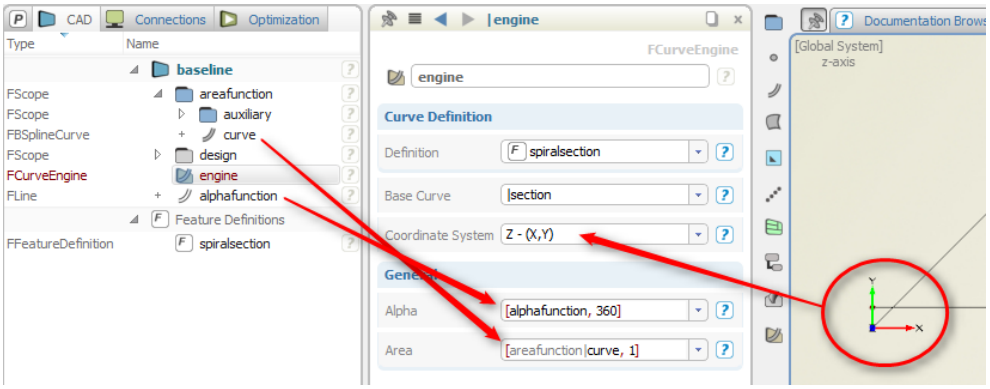
- Press the execute button of the dialog so that the b-spline curve gets created in the tree in the scope "f1".
- Rename "f1" to "areafunction".

✓ Any curve can be utilized as a function. For instance, the *generic curve* allows you to define pure mathematical functions. The *fspline* can also be nicely used as a smooth function with reduced degrees of freedom, perfectly suited for automated optimizations when changing functions automatically using design engines.

6 Curve Engine

In this step, the 2D section definition and the parameter distributions are connected using a curve engine.

- Choose *CAD > curves > curve engine* and name it “engine”.
- Set “spiralsection” at the attribute *definition* of the curve engine. Now, the available feature arguments *alpha* and *area* are shown in the interface and the inputs can be set.



Use either drag & drop or the pull-down menu of the attributes:

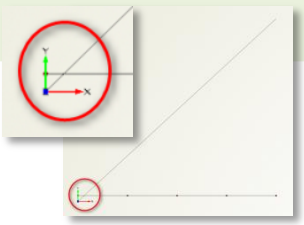
- Set “alphafunction” for attribute *alpha*.
- Set the curve from scope “areafunction” for attribute *area*.

We modeled the distributions in a normalized system. The function for alpha will therefore return values from 0° to 1° (ordinate range) but we need to have a range of 0° to 360°:

- Insert a multiplication factor right after the curve i.e. “[alphafunction, 360]”, see the screenshot above. The values from this function will then be scaled by “360”.

✓ The attribute *base curve* is automatically set to “section” since this is the only curve in our feature definition “spiralsection”. There may have been more curves available (e.g. auxiliary curves combined to a final *polycurve* etc). The base curve is the one that is taken for surface generation.

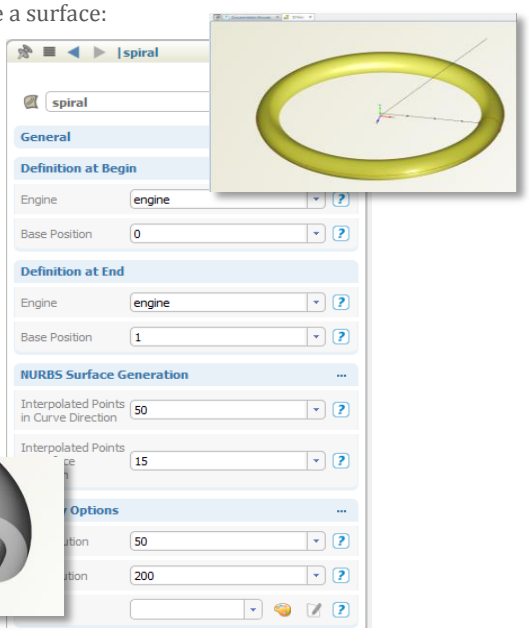
Also, the default *coordinate system* of the curve engine for given functions is the xy-system which is correct in our tutorial since we modeled the parameter distributions in the xy-system (step 4+5). In some situations it is more convenient to use other systems (e.g. for ship hulls in the zx-system etc.).



7

Meta Surface

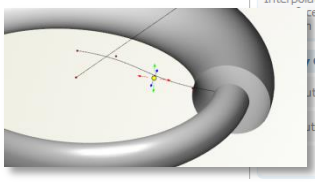
So far, the curve engine has all the information for returning a curve in the abscissa range [0,1] since the 2D circle definition and its distributions are defined within this range (Note: there is also a curve engine command *getCurve()* for this). From this information, we can generate a surface:



- Select the curve engine (not really required but convenient since the next step will include this selection by setting the engine for the meta surface).
- Choose *CAD > surfaces > meta surface* and name the new surface "spiral".

Let's modify the area distribution as a demonstration:

- Select the points of the area function (in scope "areafunction|auxiliary") and move them in y-direction in the 3D view to change the area distribution of your surface.



Note that this is typically what a *design engine* does in an automated process via *design variables* for the x- and y-coordinates of curve points.

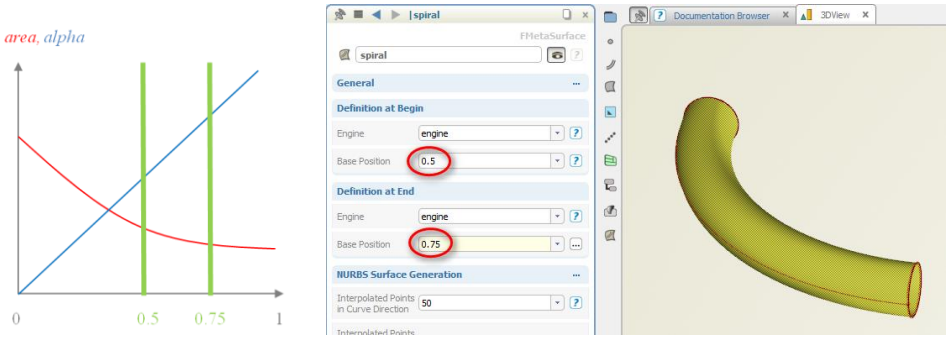
✓ In the display options of the surface, set the u-resolution to "50" and the v-resolution to "200". These settings are only for rendering and do not affect the surface definition. Note that "u" corresponds to the section definition while "v" represents the sweeping direction of the surface.

If you want to have the surface and the parameter distributions in different 3D views, then open another one via *menu > view > windows > 3DOverView* and/or use the point-, curve- and surface-filters at the bottom of the 3D views (i.e. to hide points and curves in *3DView* and show them only in *3DOverView* where the surface filter then needs to be activated).

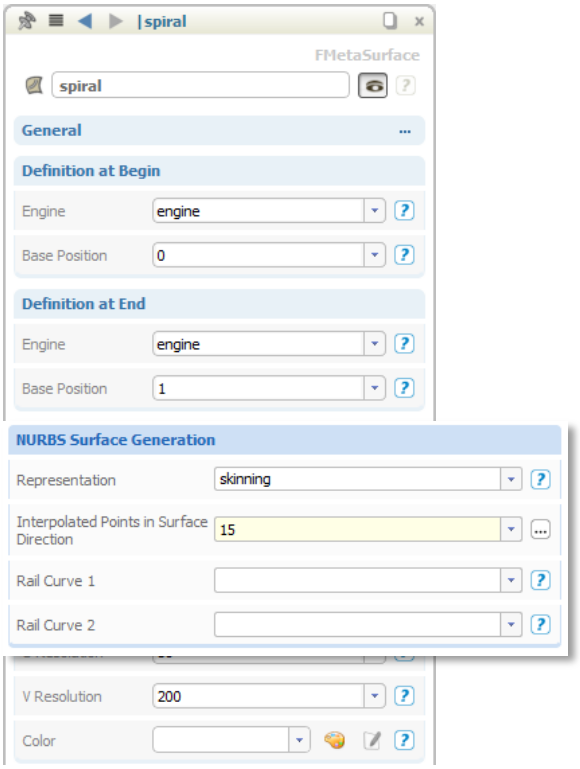
8 Meta Surface: Some Remarks

Let's have a look at the *base positions* and some finetuning attributes of the meta surface:

- Base positions: Identify the surface generation range that corresponds to the abscissa values of your coordinate system. In our case, our definition range (where the distributions are given) is from "0" to "1" which is also set in the two base positions, respectively. Change the values e.g. from "0.5" to "0.75" in order to generate a surface only in this range.



- NURBS Surface Generation: The meta surface entity needs to be converted into a common CAD entity so that it can be exported, for instance in IGES format. Meta surface positions get interpolated and a NURBS surface is generated (this is what you finally see in the 3D view). You can control the surface interpolation by using the two input attributes for *number of points in curve direction* and *surface direction* (here: "sweep" direction).

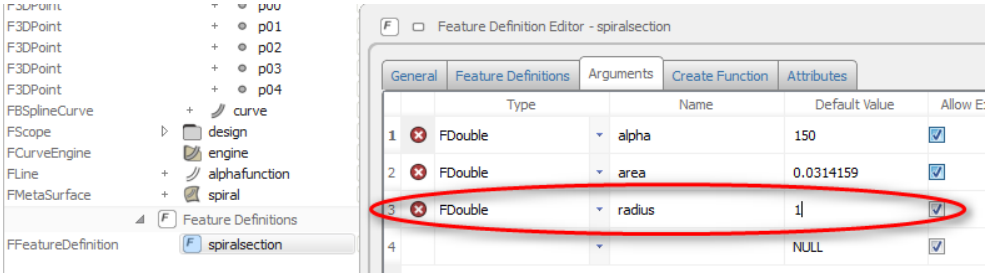


If you click at the *NURBS Surface Generation* category header, the attribute *representation* is shown. "Skinning" is the recommended option. This creates *n* curves (e.g. n=15) as defined in your feature definition, and applies a surface skinning in sweep direction.

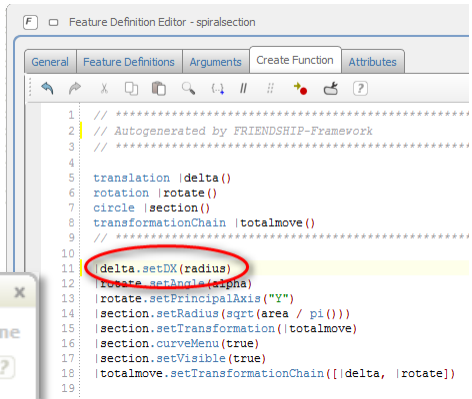
9 Adding the Circumferential Radius

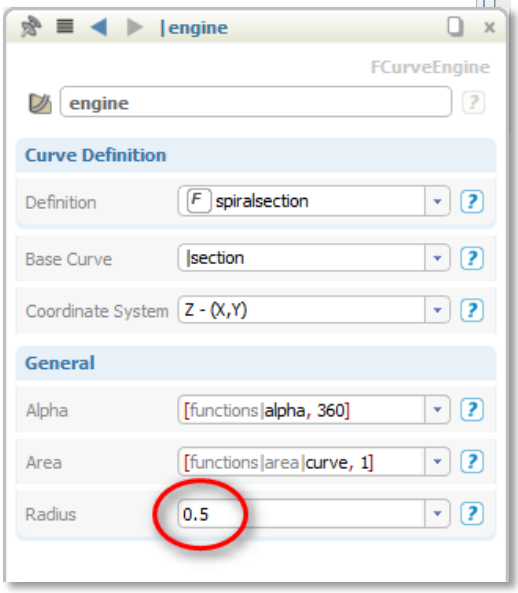
As a final task, we want to introduce another parameter which controls the circumferential radius so that we receive a typical spiral shape:

- Double-click the feature definition “spiralsection” in the tree (node *feature definitions*) in order to edit the section description.
- Add another argument with name “radius”, choose *FDouble* to be the type and a default value of “1”.



- In the *create function*, replace the fixed value “1” of “delta.setDX(1)” with the radius.
- Press apply and close the dialog.
- Select “engine” in the tree and try different values for the new input *radius*.



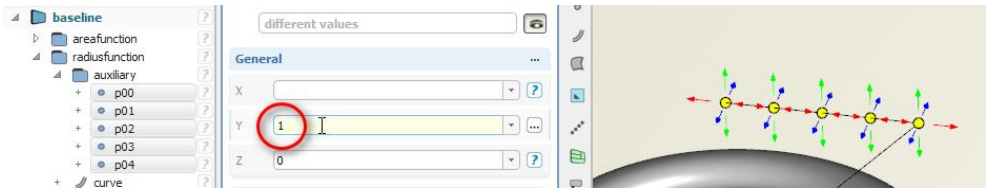


10

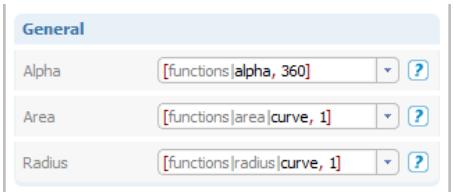
Function for the Circumferential Radius

The new radius parameter is introduced so we still need to create a function and connect it to the spiral:

- Copy and paste the scope “areafunction” and name it “radiusfunction”.
- Select all points in there and set the y-coordinate to “1” as a starting point.



- Select “engine” and set the new input curve for *radius* e.g. by using drag & drop.



- Select “p03” and “p04” and set the y-coordinate to “0.5”.
- Interactively: Pick a point of the radius function and drag it in the y-direction.