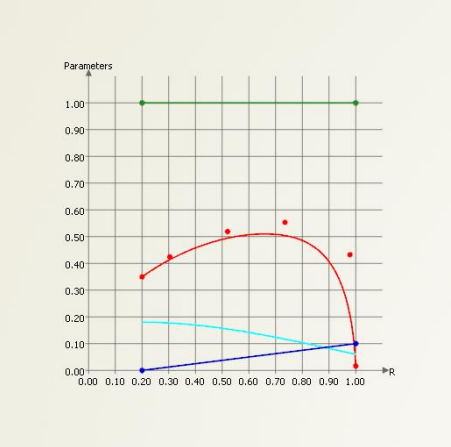
Generic Blade

CAESES provides comprehensive functionality for designing different types of blades for various industries. In this tutorial, the *generic blade* entity is introduced, and as an example we will model a ship propeller. The generic blade transforms a given profile definition onto a cylindrical surface according to characteristic blade functions such as rake, skew and pitch.



For profile design, any CAESES curve type can be used. This also includes mathematical definitions (e.g. via *generic curve*). In this tutorial, a NACA curve will be utilized.

The radial functions of the blade are defined in a normalized system, which is the global xy-system of CAESES. Again, any kind of curve can be used to define functions. Typically, these functions are later on controlled by design variables in optimization processes.

The generic blade can serve as input for a *propeller* entity, which is also briefly illustrated.

Prerequisites for this tutorial are the introductions to basic modeling and feature definitions.

CAESES Project

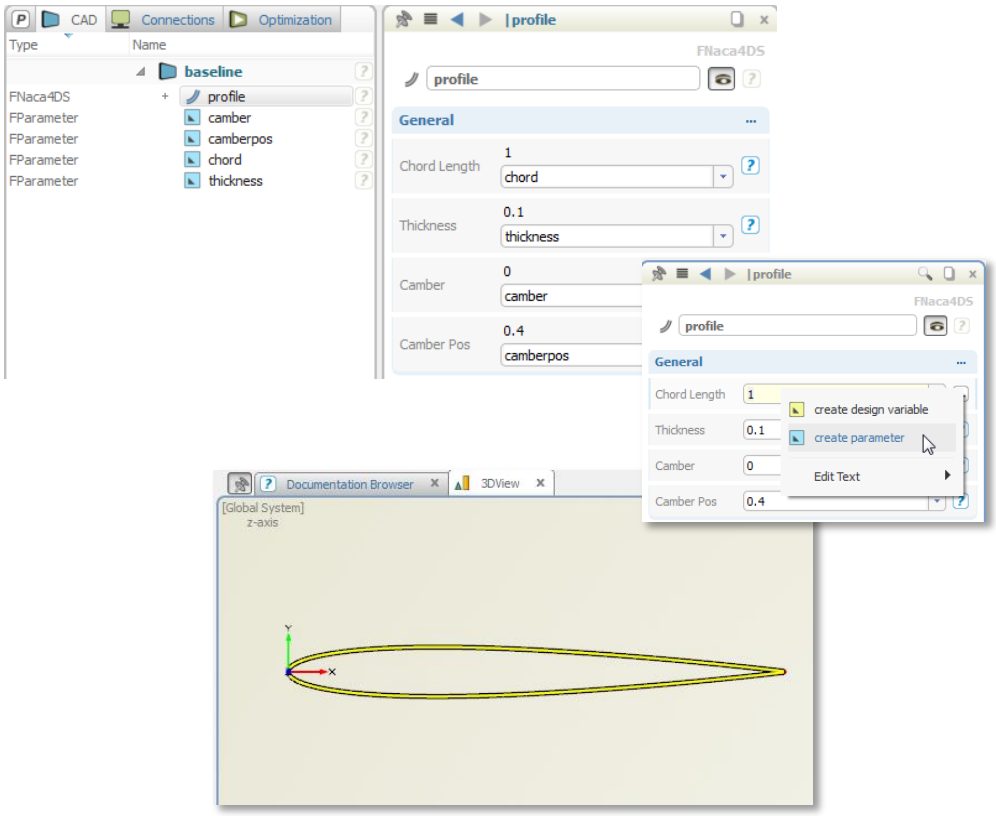
The resulting model can be found in the section *samples* > *tutorials* of the documentation browser.

1

Profile Definition

The user is free to design custom profiles using any curve type that is available in CAESES. Mathematical definitions such as customized airfoil definitions can also be set up for describing a profile. In this tutorial we will utilize a NACA profile that comes with CAESES, so that we can focus on the actual blade generation process. Profile definitions need to be provided in the *xy*-system of CAESES where the leading edge is expected to be located at the origin.

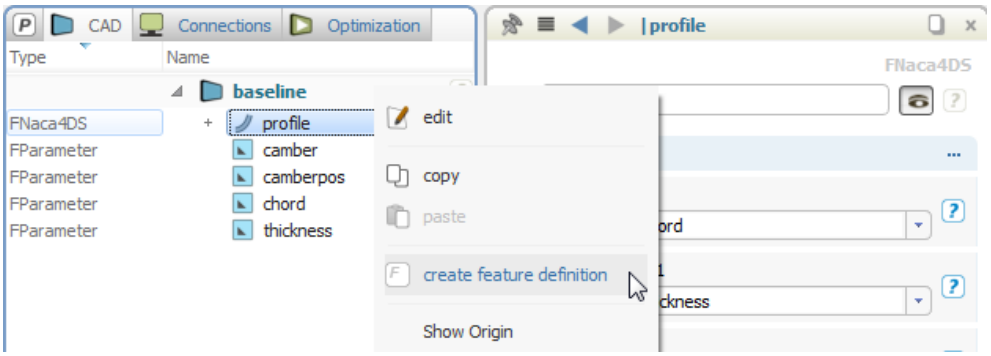
- Create a NACA profile via *CAD > curves > NACA-4DS curve* and name it “profile”.
- Create parameters for the variable input values of the profile (chord length, thickness, camber, camber pos) and set the names to “chord”, “thickness”, “camber” and “camberpos”.



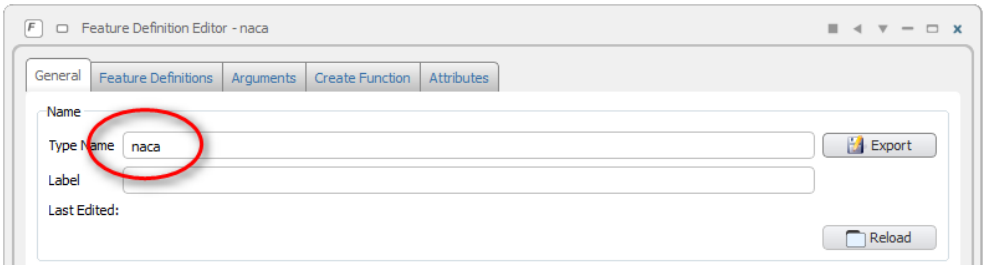
2 Feature Definition

Similar to *meta surfaces*, the generic blade entity is also based on *curve engines*. For this reason, we put the profile description into a feature definition, which will be used by the curve engine.

▶ Select "profile" and choose *create feature definition* from the context menu.



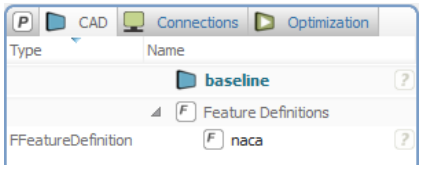
▶ Edit the new feature definition and, set the definition's type name to "naca" (if you like, you can also use "naca4ds" etc.).



▶ Press the *apply* button and close the dialog.

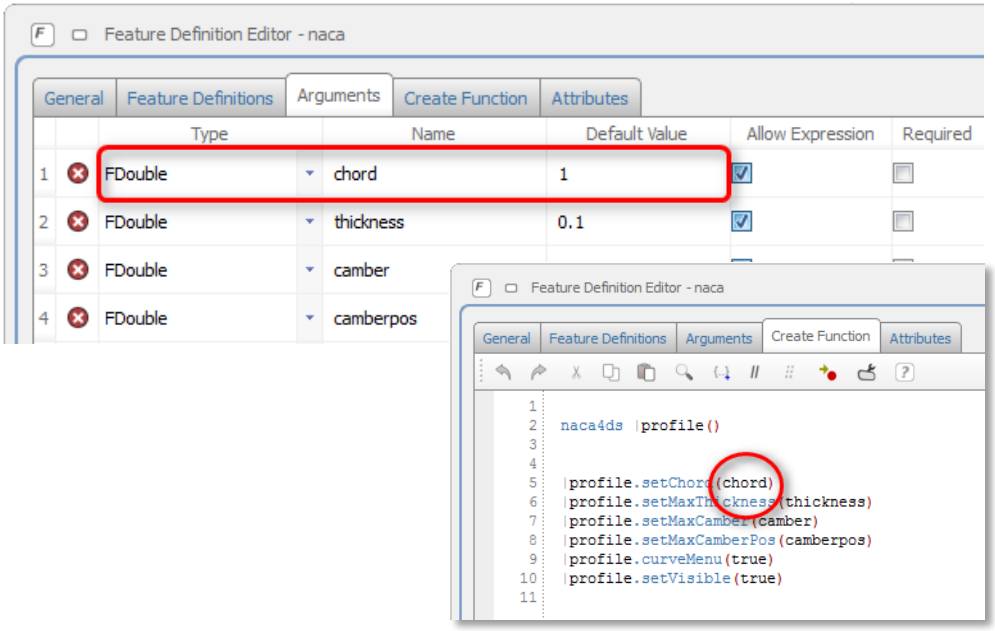
✓ By this step, we have created a profile definition ("naca") that can be used for surface creation. Note that the next steps of this tutorial are applicable to any other profile that is given as a feature definition. Therefore, the initial profile geometry from step 1 is no longer needed:

▶ Select "profile" and the 4 parameters in the object tree and delete them so that only the feature definition "naca" is left.

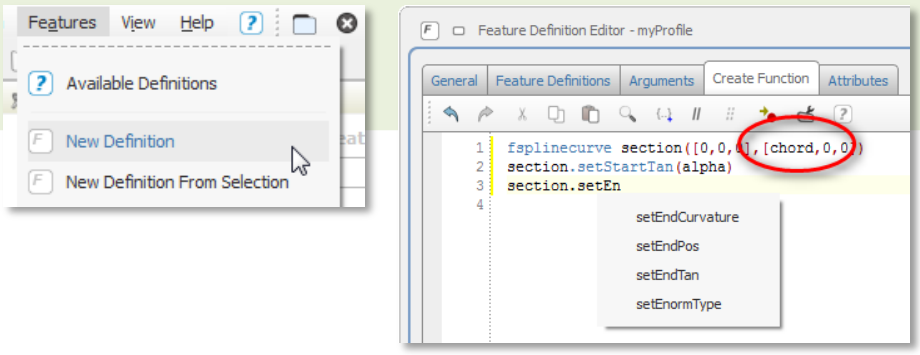


3 Feature Definition: Note

If you would like to use other custom profiles in combination with the *generic blade* (see step 5), your feature definition always needs to have an input argument with name "chord" (type *FDouble*) as shown in the figure below. Otherwise, the surface generation won't work correctly.

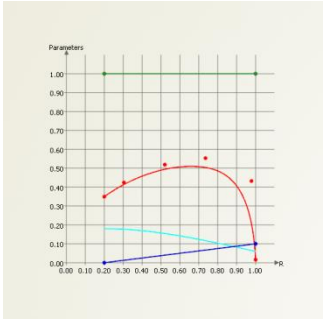


✓ We created a feature definition from a selection of objects in step 2, which is a convenient mechanism. Note that you can also start from scratch with a blank feature definition created from the menu by choosing *features > new definition*. Just type the definition of your custom profile curve as a sequence in the tab "create function". As mentioned above, make sure that you have at least an argument "chord" which defines the length of the profile.



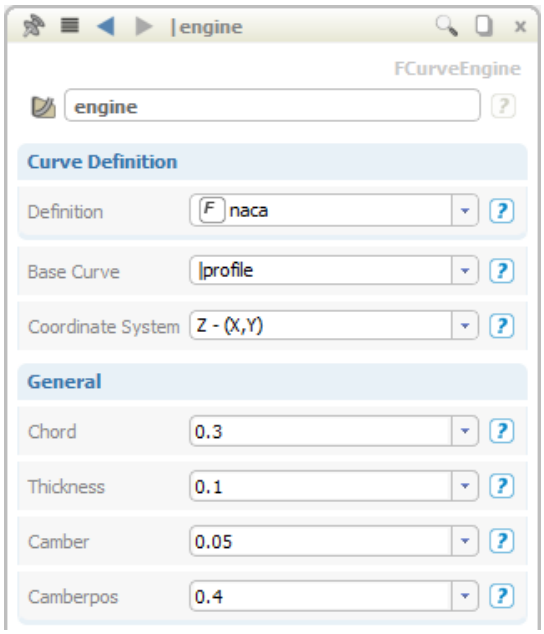
4

Curve Engine



Curve engines connect the curve (i.e. blade profile definition) with functions in a certain range. For blade design, the range is given along the radius of the blade. We consider a normalized system for the profile and blade functions. The radius runs from the normalized hub radius (e.g. R="0.2") to the tip radius R="1". For a start, we will use constant values for the profile parameters. For instance, *chord* will not change in the radial direction. We will then create radial functions in step 6.

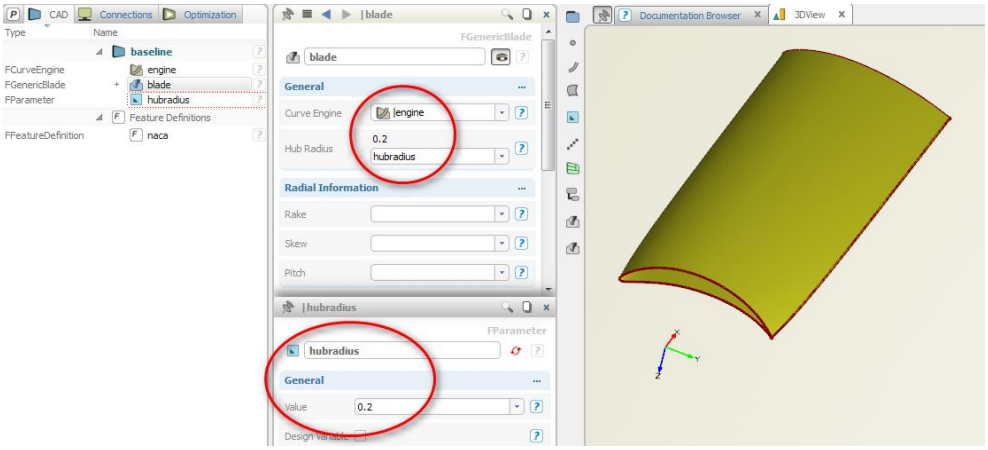
- Create a curve engine via *CAD > curves > curve engine* and name it "engine".
- From the pull-down menu of definition, choose "naca".
- Set initial constant values for *chord, thickness, camber* and *camberpos* according to the following screenshot:



5 Generic Blade

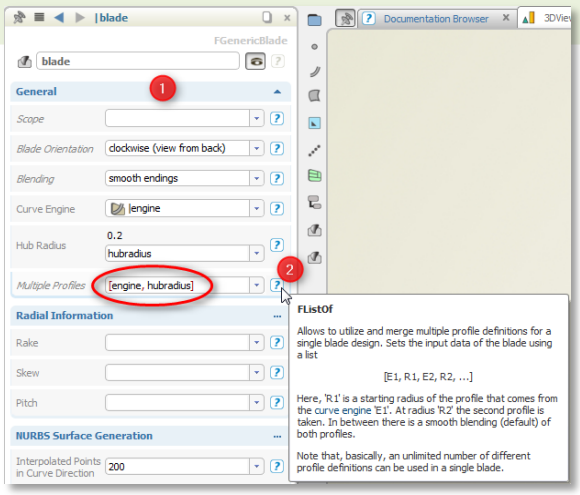
The *generic blade* requires at least one *curve engine* as input along with a (normalized) hub radius.

- Create a blade via *CAD > blade > generic blade* and name it “blade”.
- Choose “engine” from the pull-down menu of the blade attribute *curve engine*.
- Introduce a parameter “hubradius” for the hub radius of the blade (via the context menu).



An initial blade is now visible in the 3D view. Play around with the parameter “hubradius” and the constant input values of “engine”.

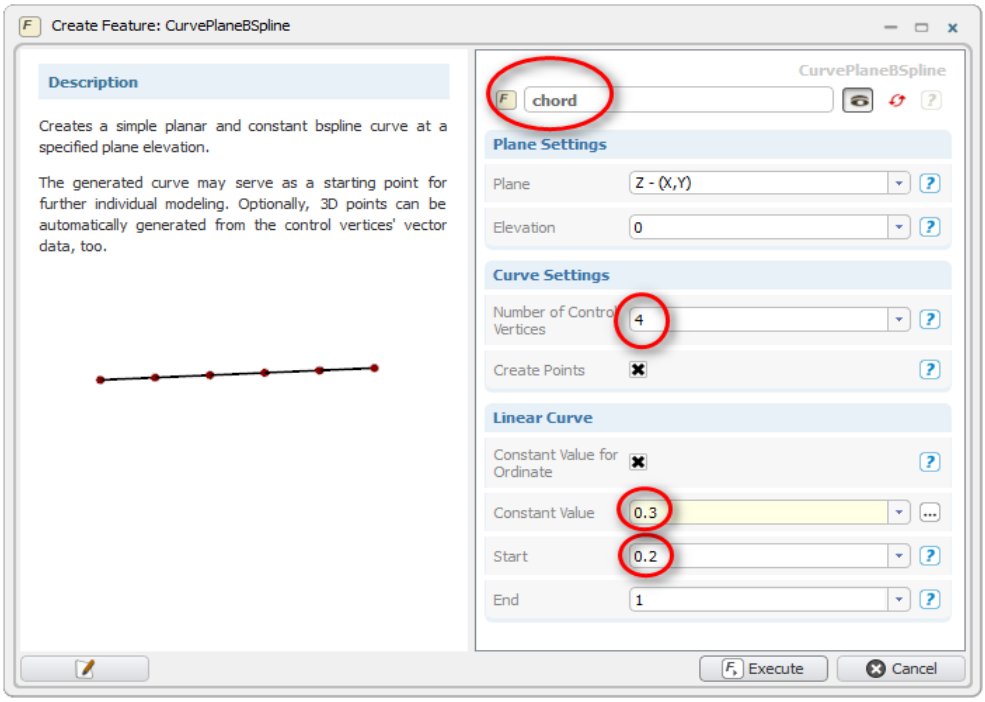
✓ Often only a single profile definition is applied in a blade generation process. If you want to have different profiles in the blade definition, insert them in the attribute *multiple profiles* according to the convention (click on category *general* to see this option). You have to provide a list of curve engines along with their starting radius. See also the attribute documentation for more information.



6 Radial Functions for Profile Parameters

We distinguish *2D profile parameters* such as chord or camber position and *3D blade parameters* such as rake, skew and pitch. For both kinds of parameters we introduce radial functions. As an example, we start with "chord":

- Choose *CAD > curves > more* (menu *more* after *NURBS curve* entry) *> planar b-spline curve*.
- In the dialog, set the name to "chord".
- Set *number of control vertices* to "4".
- Set *constant value* to "0.3".
- Set *start* to "0.2" (starting x-position of the function i.e. corresponds to "hubradius").
- Press the *execute* button.



This creates an initially constant curve (i.e. a function for chord along the radius) in the xy-system. Let's organize our project structure slightly before we continue:

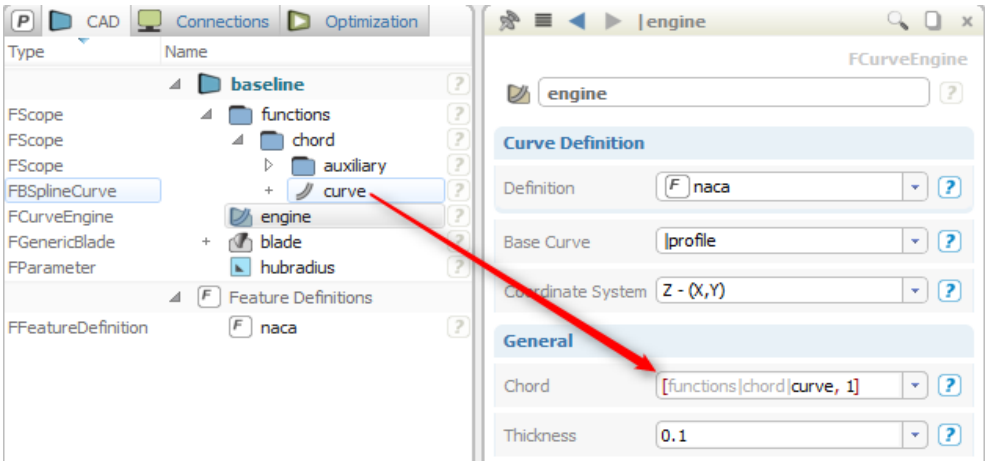
- Select the scope "chord" and create another scope via *CAD > scope* ("chord" is then put into the new scope). Rename the new scope "sc1" to "functions".

✓ Note that you can use any CAESES curve type as a function, not just *bspline* curves.

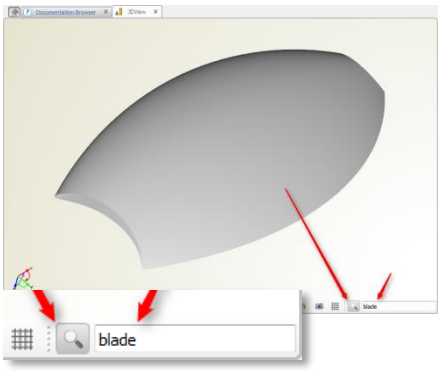
7 Connect Radial Function to Curve Engine

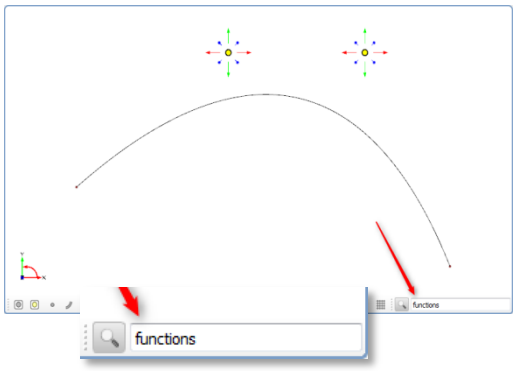
Now we connect the chord function to the blade via the curve engine (remember: the curve engine is the input for the blade).

- Select “engine” and drag & drop the curve from scope “chord” into the corresponding attribute editor *chord* of the curve engine.



- Open another 3D window by choosing *view > windows > 3DOverView.*
- See the screenshots below: Use the name filters of the two 3D windows in order to filter only for the blade in “3DView”, and only for the functions in “3DOverView”.
- Select some curve points of “chord” (from the scope “auxiliary” or in the 3D window) and move them in the y-direction to see how the chord length can be changed in radial direction.

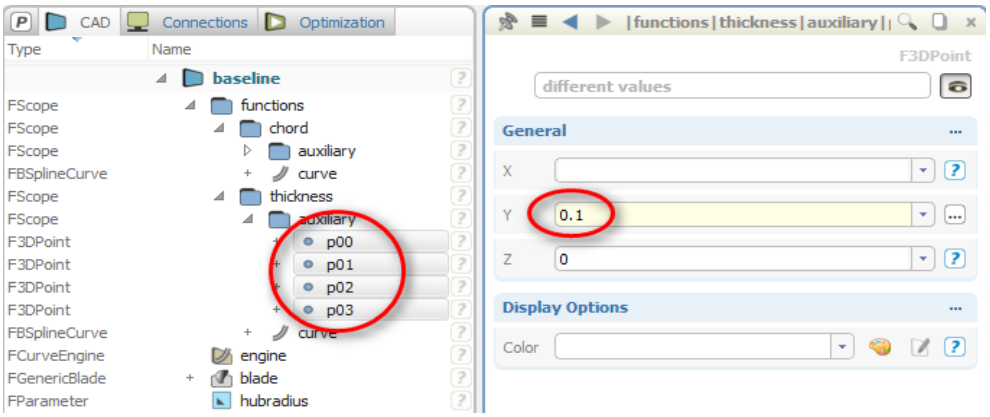




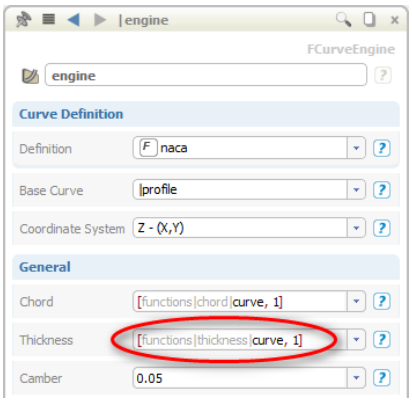
8 Radial Functions for Other Profile Parameters

In order to keep it simple, we just copy & paste the chord function and adjust the curve points according to reasonable values for *thickness*, *camber* and *camberpos*.

- Select the scope “chord” (in scope “functions”) and copy & paste it.
- Set the name of the new scope from “chord01” to “thickness”.
- Select all four points of the thickness function in the tree and set the y-value to “0.1”.

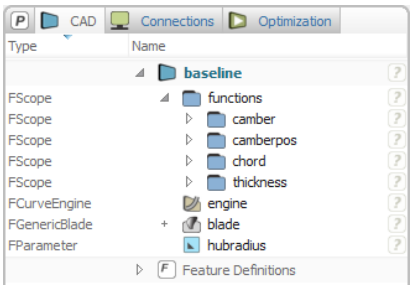


- See also step 7: Select the curve engine “engine” and drag & drop the thickness curve into the *thickness* attribute to connect it to the blade.



In the same manner, create radial functions for *camber* and *camberpos*.

- For *camber*, set the y-value of all four points initially to “0.05”.
- For *camberpos*, set the y-value of all four points initially to “0.4”.

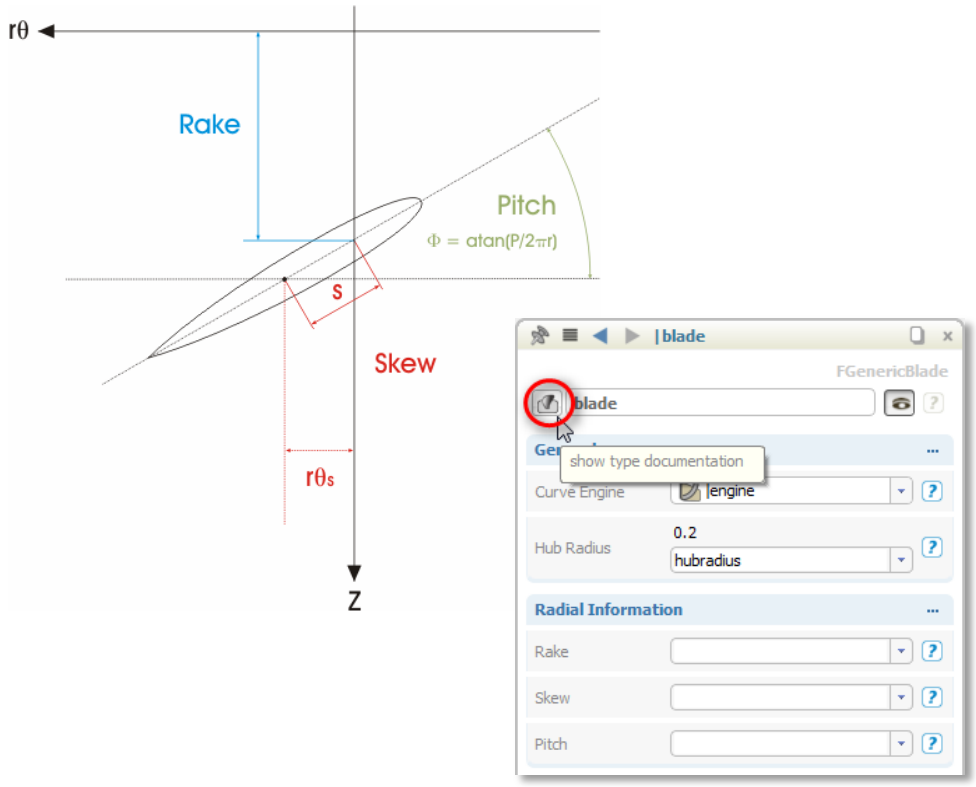


You can now change these functions interactively in the 3D window by moving their points in x- and y-direction. Note that the x-position of the first point (at x=0.2) needs to be untouched since this corresponds to our hub radius. See also step 13 where this issue is addressed again.

9 Radial Functions for 3D Blade Parameters

Functions for rake, skew and pitch can be set in the generic blade. Information about these parameters is given in the type documentation:

- Select the blade object in the tree.
- Click on the type icon of "blade" to show the documentation in the *documentation browser* (see editor screenshot below).
- Rake, skew and pitch are typical parameters used in propeller design. See the documentation browser for the definitions (shown below).



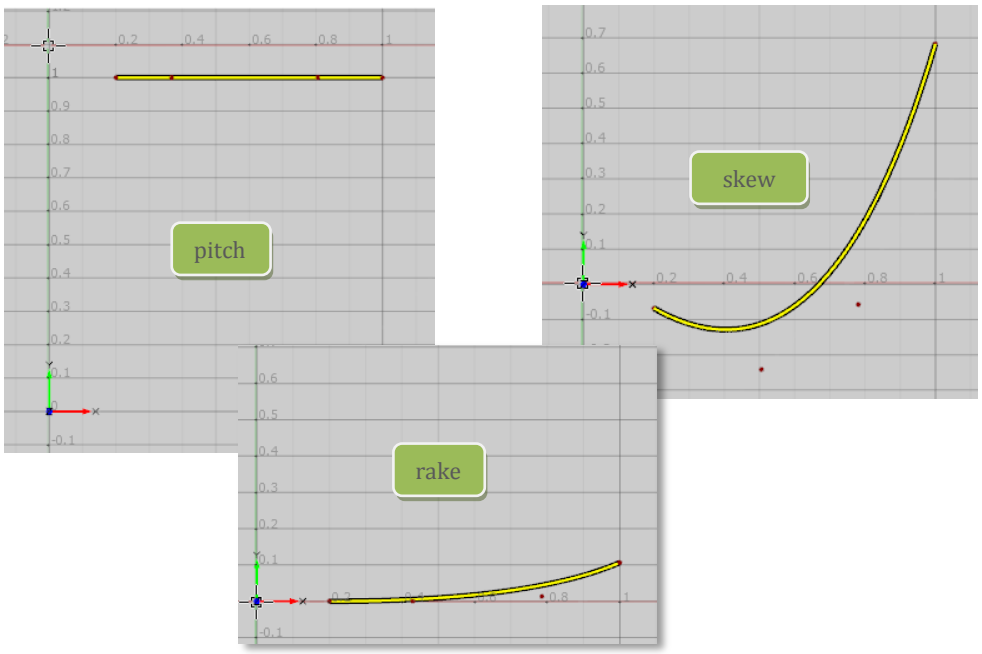
Function curves for these parameters are created as described in step 8, for instance, using an initially constant bspline curve or via copy & paste of an existing function such as "chord". Alternatively, use the smooth *fspline* curve, an *interpolation* curve (interpolate point data) or the *generic* curve (for mathematical definitions). All curve types are given in the menu *CAD > curves*.

Examples for rake, skew and pitch are briefly given in the next step.

10

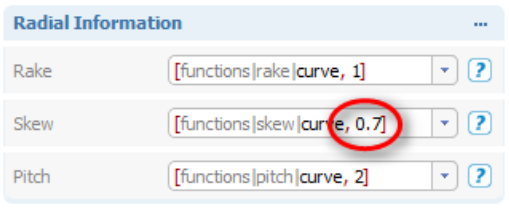
Examples for Rake, Skew and Pitch

Have a look at example functions for rake, skew and pitch. The following curves are also stem from a copy & paste action of one of the existing curves (bspline with 4 points). The values along the abscissa and the ordinate are displayed so that remodeling of such curves is easier.



► Set these three curves at the blade (category *radial information*), see the screenshot below.

We can also use a scaling factor for each function. Hence, these curves can still be designed in a normalized system (e.g. convenient visualization) but the ordinate values get scaled internally. Replace the default value of "1" (no scaling) with a new value for the factor. Note that this might also be a separate parameter (*CAD* > *parameters* > *parameter*) that can be controlled. The function values are then automatically scaled by this value during blade surface generation. Try to change these values and watch the surface generation.



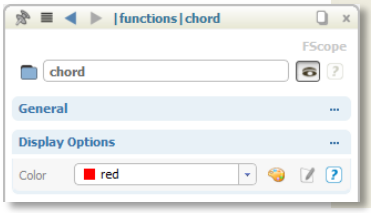
11

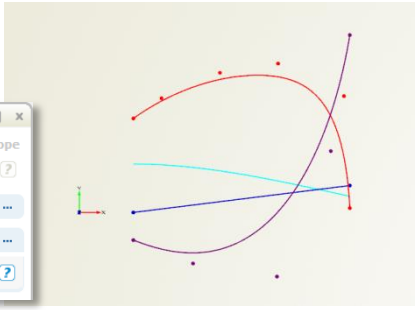
Colors and Labels

So far, an initial blade has been generated and radial functions control the shape. For convenience, it is often useful to assign different colors and labels to the functions in order to manipulate them more easily.

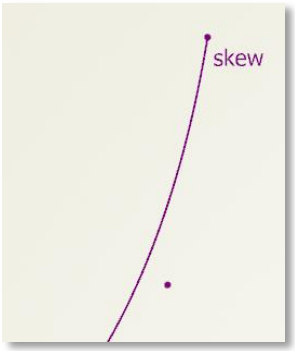
Assign different colors to the functions:

- Select the scope of a function such as “chord” and set a color.



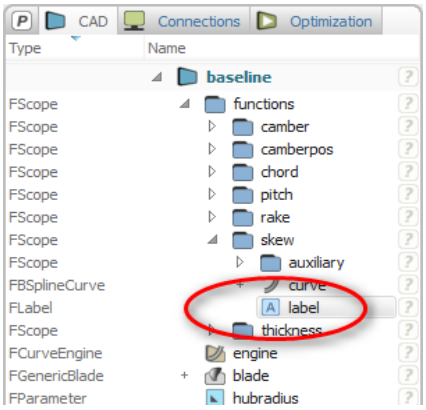


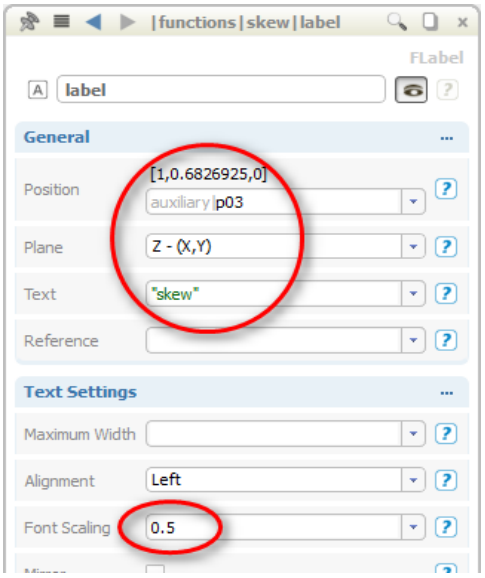
✓ All objects in a scope will then use the scope color, if no individual color has been set before for an object in this scope.



To assign labels for the functions:

- Choose *visualization* > *label* and move the new label into the function scope.
- Set a *position* – for instance, drag & drop the last point of the curve into it the *position* attribute.
- Set a text, such as “skew”.
- Set a *font scaling* of “0.5”.
- Configure background settings (e.g. *draw frame filled* etc.).



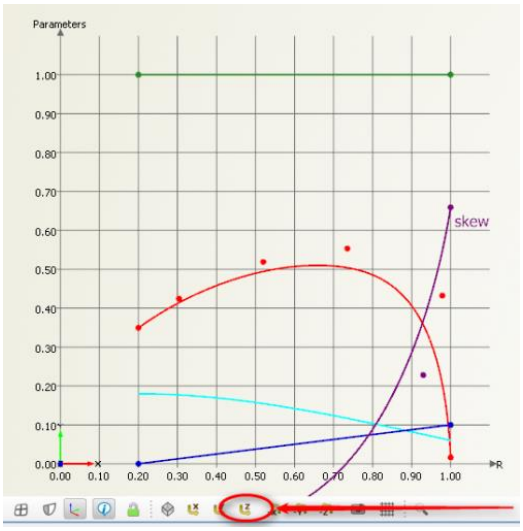


12

Coordinate System

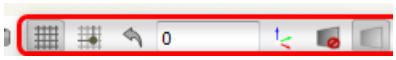
The functions are modeled in the global system. If you want to have a coordinate system with labels for the axes, then try this out:

- Create a coordinate system via *CAD > transformations > coordinate system.*
- Set the attribute *origin* to "[0,0,0]".
- Set *number of divisions* to "10".
- Set the attribute *grid type* to "All".
- Activate the z-view at the 3D view.

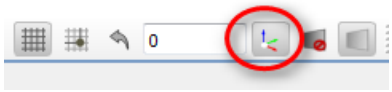


Alternatively, you can also use the grid view of the 3D windows:

- Switch on the grid in the 3D view.



- Activate the plane handle.



- Click on an axis of the handle to activate it and drag the handle.
- Use the handles (white points) in the corners of the grid to resize the grid plane.
- Switch off the plane handle (by deactivating the button) once the plane has been positioned and sized.

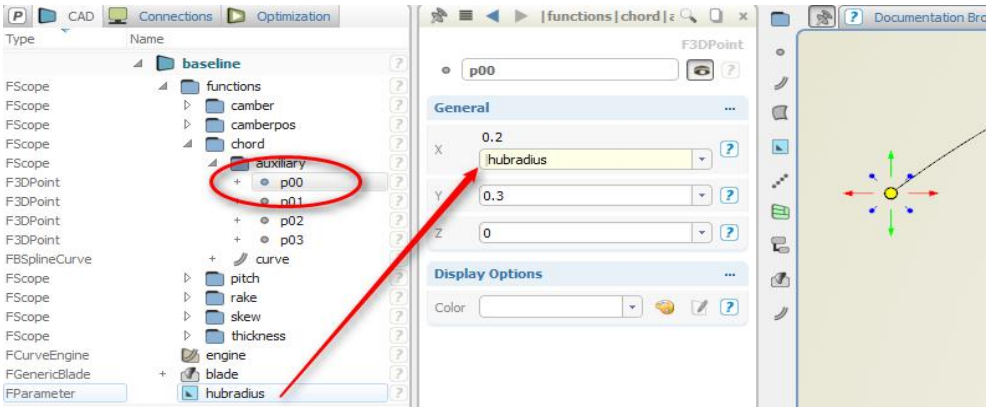
✓ See also the introductory tutorial of the graphical user interface where the plane view and clipping planes are explained.

13

Hub Radius

Usually, the hub radius is fixed and will not change. However, for a proper parametric model, the hub radius might also be a parameter that can be changed: All functions then simply "follow" i.e. they change their range according to the new hub radius. So far, the parameter "hubradius" has been set for the blade but not for the functions which we change in this step:

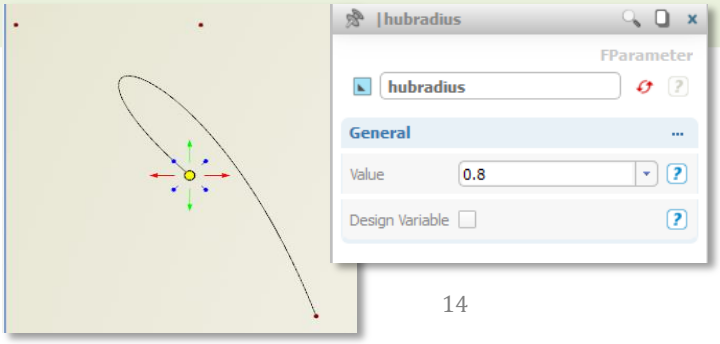
- Select the first curve point of all seven functions (this can also be done all at once using CTRL, or in the 3D window, select all first points with a single selection).
- Drag & drop the parameter "hubradius" into the x-coordinate.



✓ In step 6, only the discrete value of the hub radius ("0.2") was used to set the x-coordinate of the first point. In this step the parameter "hubradius" is used to control the x-coordinate of all functions.

- Try values for "hubradius" such as "0.15" or "0.25.

✓ Make sure that the x-coordinate of the first point is smaller or equal than the second point. For instance, large values of "hubradius" will result in an invalid function since the other curve points will not adapt automatically (they are not dependent on "hubradius" so the x-coordinate of the first point can get greater than the x-coordinates of the other points.).



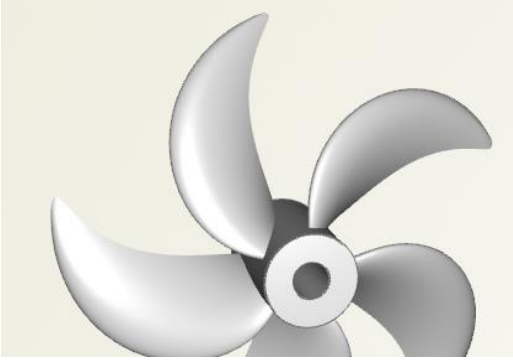
14

Propeller

The blade has a normalized outer radius of “1”. The propeller entity allows scaling of the blade to the real dimensions.

- Choose *CAD > blade > propeller*.
- From the pull-down menu of the attribute *blade*, choose our “blade”.
- Set number of blades (attribute *NOB*) to “5”.

If you cannot see the propeller, check the name filter at the bottom of the 3D view.

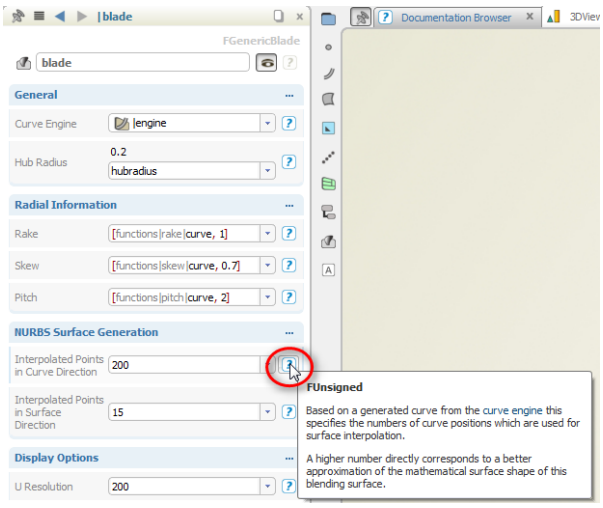


15

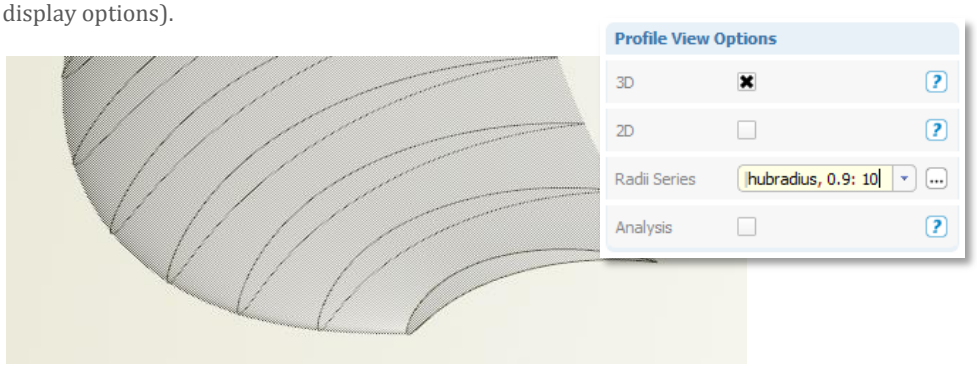
Conclusion

In this tutorial the generic blade entity is presented and a simple blade surface is generated. Similar to *meta surfaces*, the blade is also based on a curve definition that is connected to user-defined function curves via the curve engine.

Note that the blade surface can further be finetuned by using its NURBS attributes. See also the *meta surface* introductory tutorial for more information as well as the attribute documentation.



There are additional options available for the blade – just click on the different categories to view them (for instance, *profile view options* in the category display options).



The generic blade conveniently comes with a set of blade-related attributes and options such as rake, skew, pitch and profile view options etc. A more generalized way of generating a blade surface can be realized with meta surfaces in combination with a *cylindrical transformation*. Such a design process is demonstrated in one of the next blade design tutorials.

For turbomachinery blades, the *stream section* curve is available which allows mapping of 2D profiles onto arbitrary stream surfaces i.e. not just cylindrical stream surfaces.