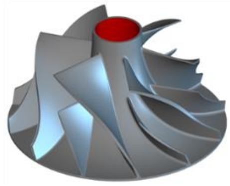
Meridional Camber Curve

For the customized design of centrifugal blades for e.g. pumps and turbochargers, CAESES provides a specialized curve, called *meridional camber curve*. Basically, this curve expects the following main input:



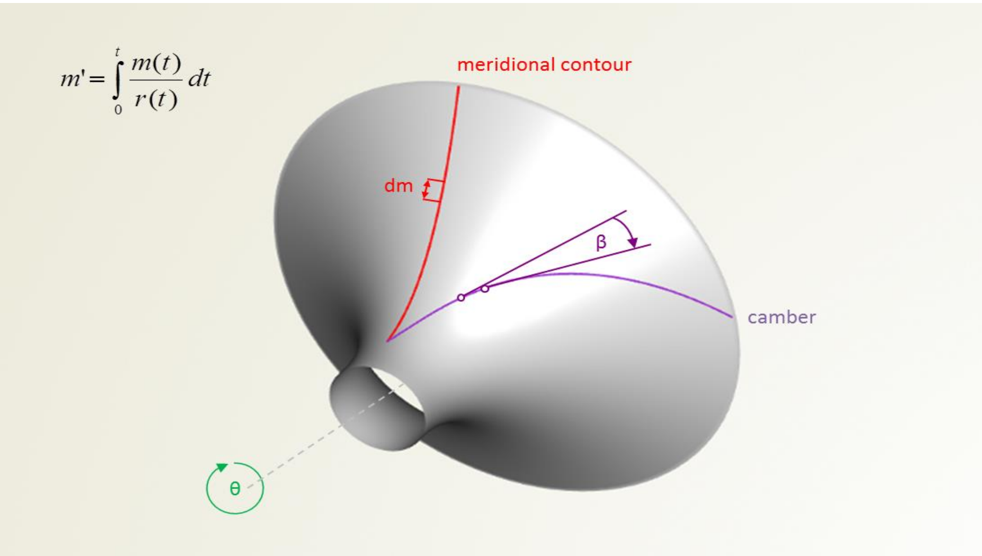
- 2D meridional contour that defines a surface of revolution
- Beta function that defines the angle distribution along the camber curve.

In this tutorial, this camber curve gets created and explained in more detail. In addition, check out this video which shows an impeller model based on this curve.

Meridional Camber Curve vs. Stream Section:

In addition, there is a curve type called “FStreamSection”, which is the main curve type for the design of turbomachinery blades. The *stream section* can be used to map custom profiles onto a stream surface. It readily allows you to use a camberline definition and a thickness distribution. In contrast to the stream section, the *meridional camber curve* from this tutorial is more a basis for own designs with high customization level. It provides the meanline curve on a stream surface, and you can then start to define a camber surface in order to apply own thickness definitions normal to the camber surface.

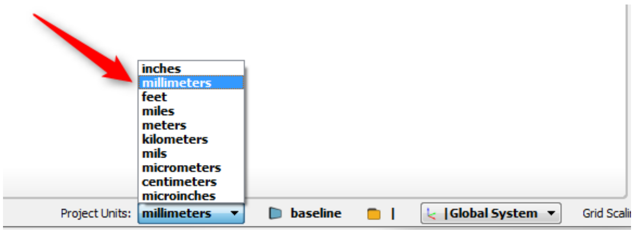
Please also check out the impeller tutorial, which describes the capabilities of the stream section curve type!



Getting Started

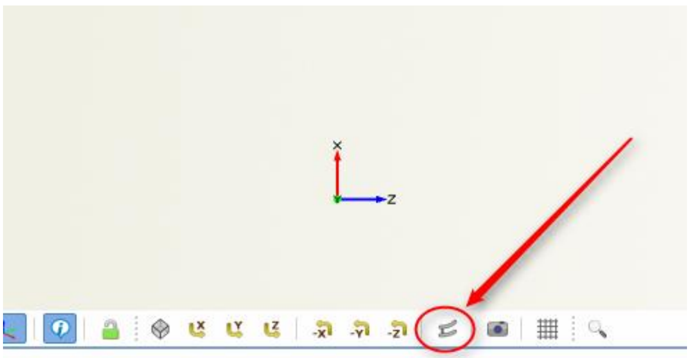
We need a 2D meridional curve that defines a surface of revolution Therefore we make sure that we use the correct project units. In addition, we get in touch with the zx-view which is the main 2D view for designing the blade's hub and shroud contours.

- Switch the project units to millimeters.



✓ Projects units are relevant when it comes to geometry exports at a later stage.

- Check out the zx-view button at the 3D view which always brings you back to the main hub and contour view in the upcoming steps of this tutorial.



2

Meridional Curve

Let’s create a simple 2D contour that shall define our hub stream surface. This contour needs to be designed in the **zx-plane** as shown below. For instance, for compressors, the positive z-axis indicates the flow direction.

- Choose *Planar B-Spline Curve* from the curve menu.

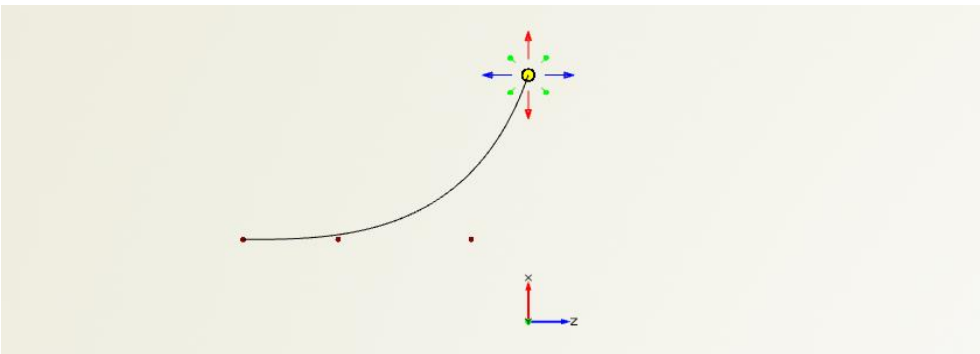
Now, a dialog pops up. Do the following settings:

- Select the plane “Y – (Z,X)”.
- Set number of control vertices to “4”.
- Set the constant value to “60”.
- Set the start position to “-200”.
- Set the end position to “0”.
- Press *Execute* in order to generate the curve.

The curve and its control vertices are now given in the object tree.

Now, we change the shape of the curve:

- Set the z-coordinate of point “p02” to “-40”.
- Set the x-coordinate of point “p03” to “180”.

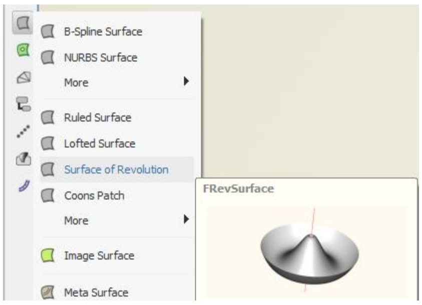


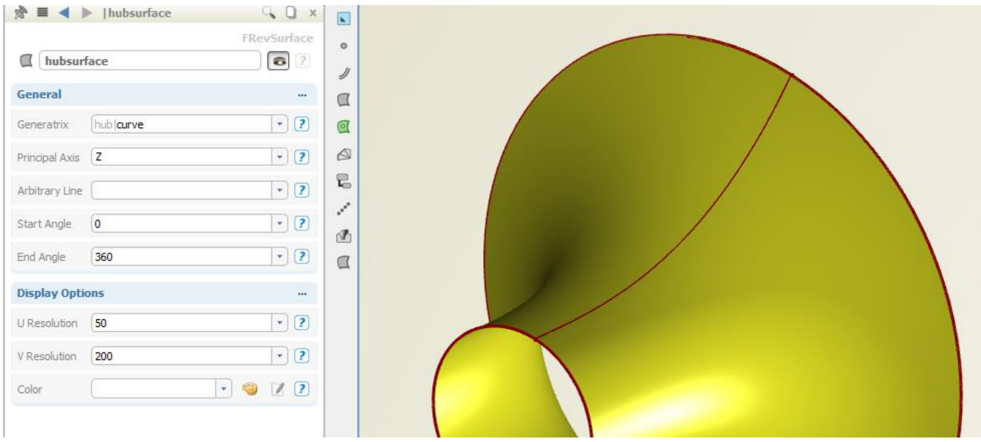
- Reset the scope name from “f1” to “hub”.

3 Hub Surface

For visualization purposes, we create a surface of revolution.

- Choose *Surface of Revolution* from the surface menu.
- Set the meridional curve “hub|curve” as *Generatrix*.
- Set “Z” to be the principal axis.
- Set End Angle to “360”.
- Optionally, increase the rendering resolution for a smoother surface (display options).
- Rename the surface to “hubsurface”.



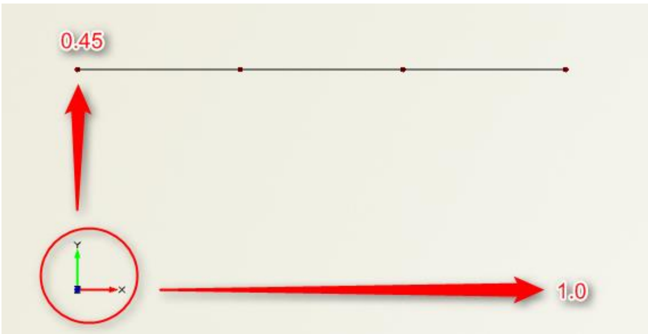


4

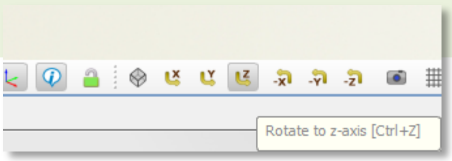
Beta Function

Let's define a function that describes the angle distribution of our upcoming camber curve. For this, we use a simple b-spline curve again.

- Choose *Planar B-Spline Curve* from the curve menu.
- Select the plane "Z – (X,Y)".
- Set number of control vertices to "4".
- Set constant value to "0.45".
- Press *Execute* in order to generate the curve.
- Reset the scope name "f1" to "beta".



✓ This curve is now created in the xy-plane, and it is a curve that will be interpreted as a normalized curve i.e. it runs in the x-interval [0,1]. In order to see this curve, you need to switch to the z-view (xy-plane) by clicking the button at the bottom of 3D view, and zoom in. It is much smaller than the meridional curve from the previous steps.

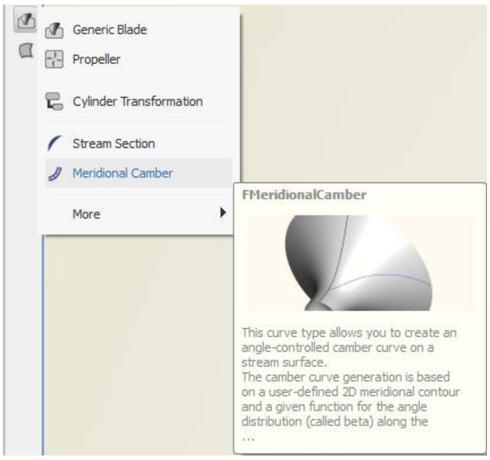


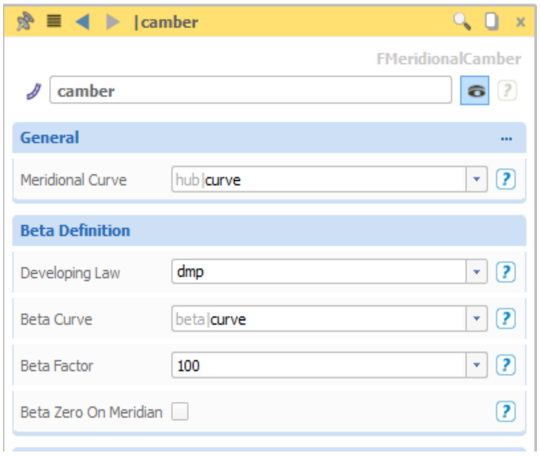
5

Meridional Camber Curve

With the meridional 2D curve and the beta distribution, we can now create a meridional camber curve that is basically mapped to the hub surface and which gets controlled by the beta curve.

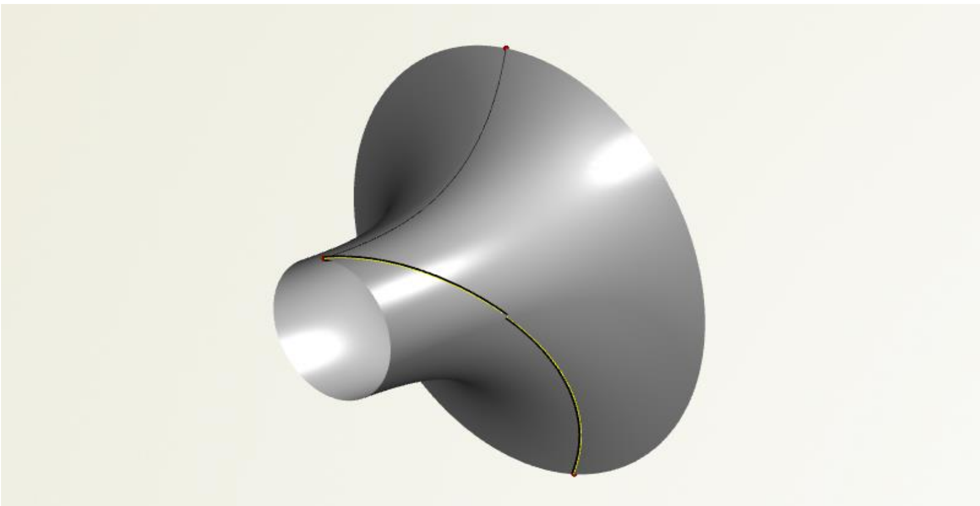
- Choose *menu > CAD > blade > meridional camber*, see the left screenshot:





- Set the hub curve as *Meridional Curve*.
- Set the beta curve from our previous step as the input for *Beta Curve*.
- Set the name of the curve to “camber”.

Now you can see the camber curve on your hub surface:

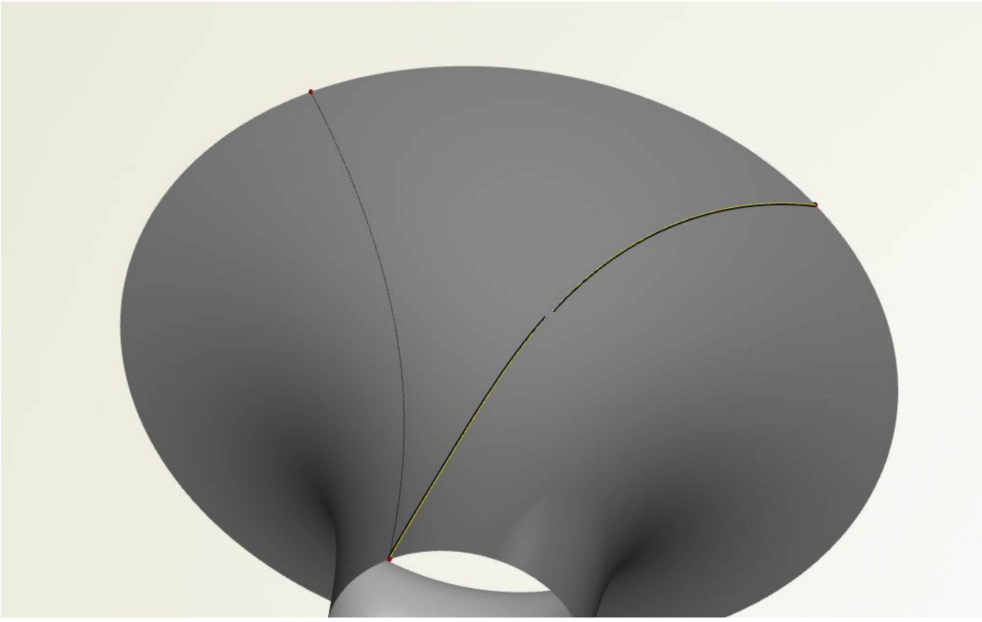


6

Changing the Beta Function

You can now change the control vertices of your beta function in order to change the shape of the camber curve.

- Simply select the control vertices e.g. "|hub|auxiliary|p03" and change the y-coordinate.
- You can also drag the point interactively in the 3D view.



- Of course, you can additionally change the hub curve by changing its control vertices, too.

7

Camber Curve Settings

Here are some brief comments about the different settings, see also the documentation for more details (press “F1” while camber is selected).

Developing Law

dmp: The beta curve is interpreted as a function of m’, based on the input curve from step 2,

$$\beta = \beta\,(\Delta m').$$

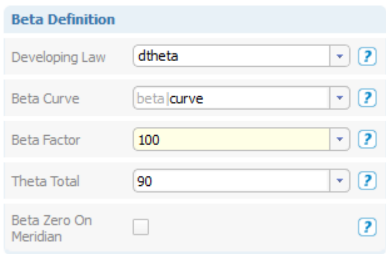
dtheta: Alternatively, you can switch to

$$\beta = \beta\,(\Delta\theta).$$

In this case, beta gets interpreted as a function of the wrap angle theta. You have to specify a total theta, see the second screenshot (“90” in this example).

Beta Factor

The ordinate values of the normalized beta function are multiplied by this value. For instance, a value of “0.45” will yield 45 degree in our example, using a beta factor of “100”. This is convenient for modeling these functions in normalized coordinate systems.



Beta Zero On Meridian

You can switch the reference for the zero-angle. If this option is active, e.g. a constant beta angle of “0” will result in a camber curve that coincides with the meridional curve.

Delta Theta

This is an initial rotation for the camber curve i.e. a rotation around the z-axis.

Apply at Curve End

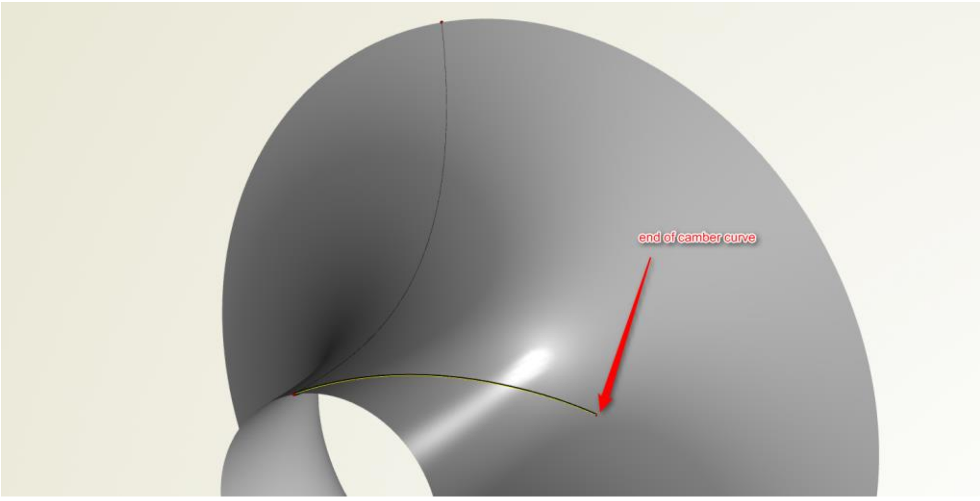
You can start at the meridional curve start or end position when developing the camber curve.

Extrapolate at End

If active, the camber curve gets extrapolated at the end which is helpful for cutting blades in a later step.

8 Curve Information for dtheta Law

Some designers of e.g. pumps might use the developing law *dtheta*. In this case, the camber curve does not necessarily end up at the end of the hub surface. See the screenshot for an example:



More detailed, matching the end of the hub is an optimization task where the beta function needs to be found that leads to the desired camber curve. In order to set up these kinds of optimizations, the camber curve provides information about its length as a fraction of the total arc length of the meridional curve. In addition, the theta angle error at the total meridional arc length can be received as well for use in optimizations. The sample (as described in the next step) contains parameters that hold these values, plus an optimization engine that corrects the beta function so that the camber curve ends up at the end of the hub surface.

Functions		
FCurve	**getBetaCurve** ( )	Returns the beta curve.
FBSplineCurve	**getConformalMapping** ( )	Returns the curve (m',theta). For instance, use this command as input for an image curve in order to use and visualize this curve information.
FDouble	**getErrorMeridionalLength** ( )	For the "dtheta" developing law, this value returns the error of the iterated meridional length as a fraction of the total length. If the length is reached, this value is zero.
FDouble	**getErrorThetaAtMeridionalLength** ( )	For the "dtheta" developing law, this value returns the theta angle error at the iterated meridional length as a fraction of theta total. If this value is zero, the theta value at the beta function's end position has been reached.
FCurve	**getMeridionalCurve** ( )	Returns the meridional contour curve.

9 What's Next?

This short tutorial should give you a general idea of how you can control the camber curve of a blade. The blade itself is not modeled in here. In order to create a blade, you can make use of the feature definitions that are provided in CAESES. You can find them in the *menu > features > blade design*. They help you to define a thickness distribution, and to define a 3D section based on a camber surface.

In addition, there is a sample "Meridional Camber" which is based on this camber curve, and the feature definitions in the blade design category. Just browse through it to get an idea of how everything is combined.

Again, check out [this video] which also explains the impeller model. And, finally, see the *stream section* curve type of CAESES, e.g. by browsing through the axial compressor example.