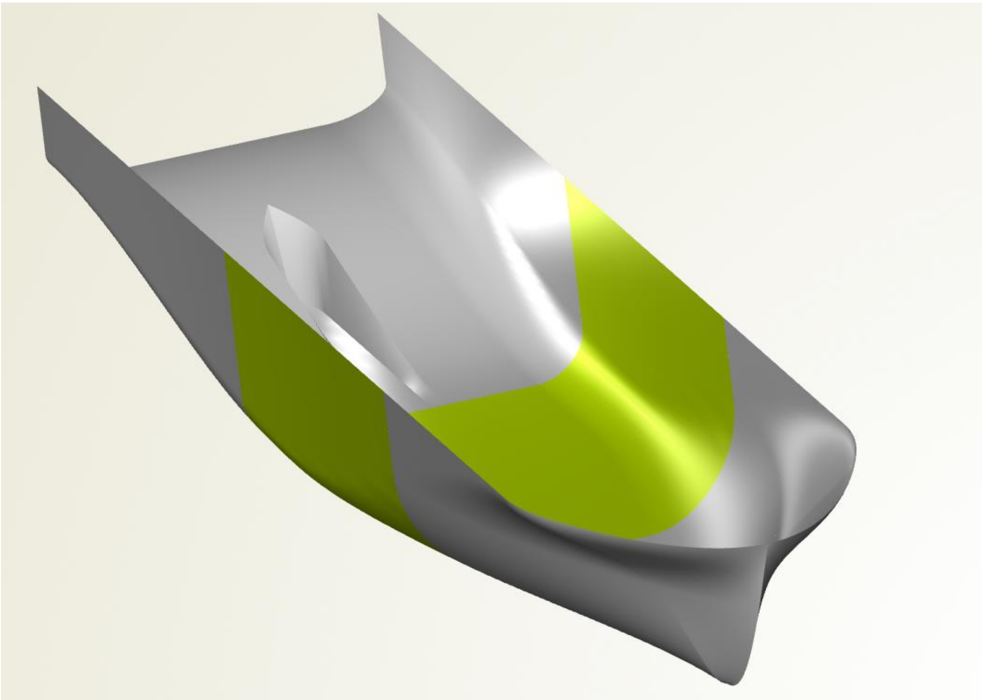
Creating a Fast Monohull Vessel

Fast monohull vessels such as RoRo or RoPax ferries and cruise ships are typically driven by two propellers with open shafts or podded propulsors.

Normally they have a central skeg to ensure better course stability, while a bulb can or cannot be present.



Before you start with this tutorial it is recommend going through the tutorials Introduction to general modelling, Introduction to meta surfaces and the Yacht Hull.

This tutorial will guide you through the process of the creation of a simple fast monohull vessel without bulb.

This tutorial includes the following functionalities curve parametrization, feature definition, curve engine, meta surface and sub surface modelling.

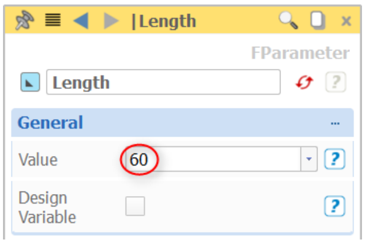
CAESES Project

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

1 Create the Center Plane Curve

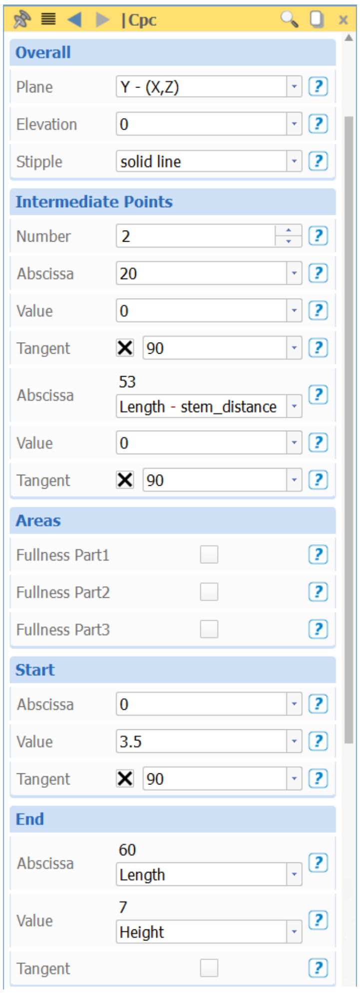
We will first create the center plane curve, which defines the keel and stem contour.

- Create a parameter via *CAD > Parameters > Parameter.*
- Name the *parameter* "Length" and set its *value* to "60".
- Create a second *parameter* with the *value* "7" and call it "Height".
- Create a third *parameter* also with the *value* "7" and call it "stem_distance".
- Create a *multi-segmented smooth curve* via *CAD > curves> more (under F-Spline curve) > Multi-Segmented Smooth Curve.*
- Click “Create” in the appearing window.



There now is a new item in the object tree called "f1".

- Select "f1" and rename it to "CPC" by changing the entry in the top box.
- Change the values as shown in the picture.



After the values are set we can change the curve color to distinguish it better from curves we create later.

- Change the *color* of “CPC” to "firebrick" by selecting it from the drop down menu under *display options.*

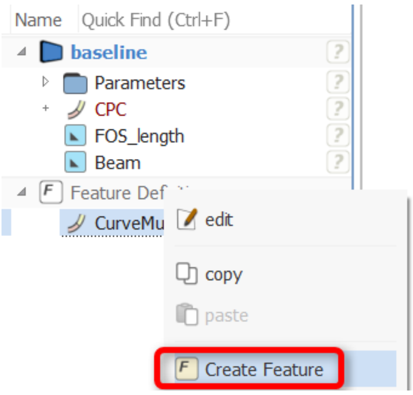
Now we can clean up the object tree a bit by using *scopes.*

- Create a *scope* via *CAD > Scope* and name it "Parameters".
- Move all *parameters* into the *scope* via drag and drop.

2 Create the Flat of Side (FOS)

We now want to create a curve which defines the flat sides of our ship.

- Create a *parameter* called "FOS_length" and set its *value* to "50".
- Create another one called "Beam" with a *value* of "12".
- Create a *multi-segmented smooth curve*, but this time by expanding the node "Feature Definitions" in the *object tree* and right clicking on "CurveMultiSegmentedSmooth" > *Create Feature.*
- Rename the new curve to "FOS".
- Change the *plane* to "Y- (X,Z)" and the number of *intermediate points* to "1".
- Set the *value* at the start "4.5".
- Create a *parameter* called "xMainFrame" with the *value* "25" and another called "bilgeHeight" with the *value* "1.3".
- Set the *abscissa* of the *intermediate point* to "xMainFrame" and the *value* to "bilgeHeight".



- Set the end *abscissa* to "FOS_length" and the end *value* to "Parameters|Height".
- Enable the *tangent* for every point and set the *angle* to "90".
- In the first tab *overall* change the elevation to "Beam/2".
- Change the *colour* to "yellow".
- Move "FOS_length" and "Beam" into the "Parameters" *scope.*

✓ It is important to create *multi-segmented smooth curves* this way from now on, as we want all curves to refer to the same *feature definition*. If we create them as we did first it will always create another *feature definition* along with the curve.

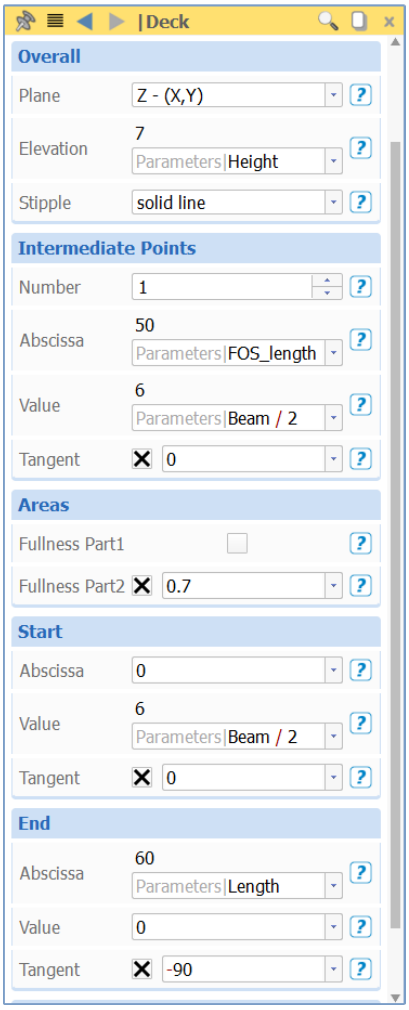
3 Create a Deck Curve

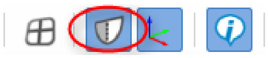
To give the ship an upper end and a clearer contour we will now create a deck curve.

- Create a *multi-segmented smooth curve* again, using the *feature definition* "CurveMultiSegmentedSmooth".
- Change the name to "Deck".

This time we will work in the "Z - (X,Y)" *plane* which will make our *elevation* the height of our curve.

- Set the *elevation* to "Parameters|Height".
- Change the start *value* to "Parameters|Beam/2" and the end *abscissa* to "Parameters|Length".
- Set the start *tangent* to "0" and the end *tangent* to "-90".
- Choose one *intermediate point* with the *abscissa* "Parameters|FOS_length" and the *value* "Parameters|Beam/2".
- Also enable the *tangent* and set it to "0".
- To get a sharper ending of the deck enable *fullness part 2* in the tab areas and give the value "0.7".
- Change the *colour* to "lime".

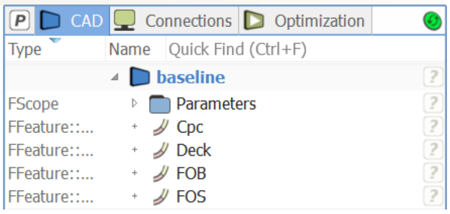


You can now enable *mirroring* by clicking  in the 3D viewer and change into a 3D perspective to see how the ship shape starts to develop.

4 Creating the Flat of Bottom Curve (FOB)

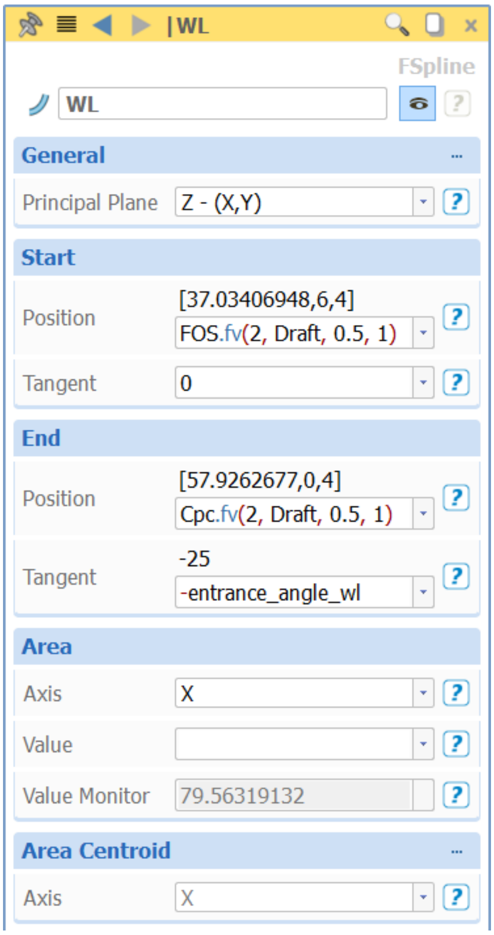
We will now create a curve to define where the bottom will be a flat surface.

- Create a *parameter* called “bilgeWidth” with the *value* “4.2”.
- Create a *multi-segmented smooth curve* and name it “FOB".
- Set the start *value* to "bilgeWidth * 0.9".
- Select one *intermediate point.*
- Change the *abscissa* of the *intermediate point* to "Parameters|Length/2.5" and the *value* to "bilgeWidth".
- Enable a *tangent* at the *intermediate point* and set it to "0".
- Set the end *abscissa* to "Parameters|Length - Parameters|stem_distance".
- Enable the *tangent* at the end and set it to "0".
- Change the *colour* to "orange".



5 Create a Water Line (WL)

As we already created a stem contour (within the center plane curve) and a flat of side, we can now create a curve to define the waterline at the forebody of the ship. This waterline shall be connected to the stem and to the flat of side and be controlled with a draft.



- Create a *parameter* called "Draft" and set its *value* to "4".
- Create an *f-spline* via *CAD > Curves > F-Spline Curve.*
- Change the name to "WL".

We will now define the start and end point of the *f-spline* with a command called ".fv()". This will allow us obtain the intersection of a curve with a certain coordinate elevation.

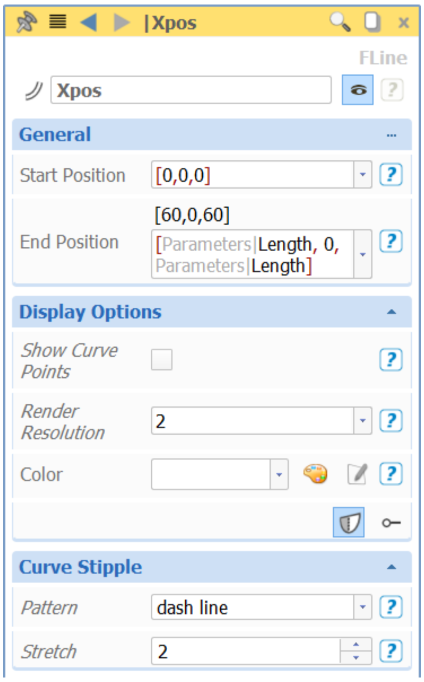
- As a start *position* set "FOS.fv(2,Draft,0.5,1)"
- We now do the same for the end *position* but this time using the "CPC" curve.
- Create a *parameter* called "entrance_angle_wl" and set its *value* to "25".
- Set the start *tangent* of "WL" to "0" and the end *tangent* to " - entrance_angle".
- Change the *colour* to "dogerblue".
- Move the *parameters* into the "Parameters" *scope.*

✓ You can see which input the ".fv()" command needs if you press ctrl + space while your curser is in between the brackets. In this case we define the axis direction as "2" which represents the "Z-axis" and the elevation as the draft *parameter*. We then restrict the range for the intersection to the second half of the curve (between the t value "0.5" and "1").

6 Creation the X-Position Curve

We will now create a simple line with a slope of 1. This line will be needed to supply the x-position to the section generation.

- Create a *line* via *CAD > Curves > Line.*
- Rename the *line* to "Xpos".
- Give as *start position* a vector that is located in the origin of the coordinate system by typing "[0,0,0]".
- Set the *end position* to the end of the ship by typing the vector "[Parameters|Length, 0, Parameters|Length]".
- Expand the *display options* by clicking on the three dots on the right.
- Expand *curve stipple* and select the *pattern* "dash line".

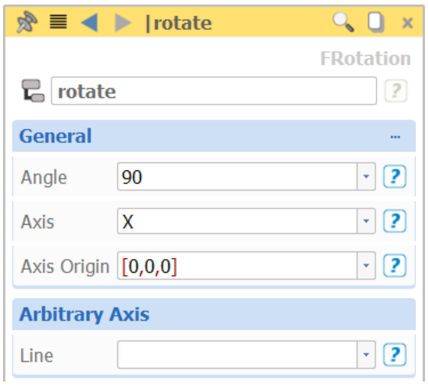


7 Rotating Curves into the Y-Plane

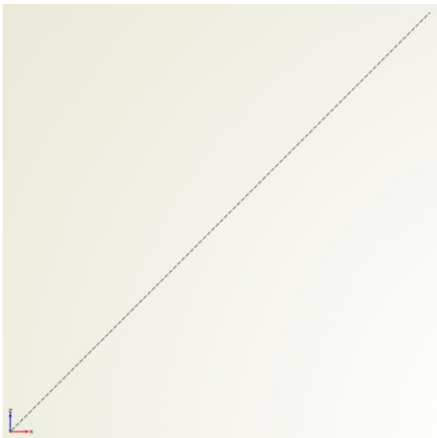
We will now create image curves for all functions in the z-plane and rotate them into the y-plane. This needs to be done, because for the curve engines to work all functions need to be located in the same coordinate system.

- Select the curve "Deck" and create an *image curve* via *CAD > Curves > Image Curve.*
- Rename the *image curve* to "Deck_rot" and give it a "dash line" *pattern.*

To be able to rotate the image curve we need to create a *transformation.*



- Create a *rotation* via *CAD > Transformations > Rotation.*
- Rename the *rotation* to "rotate".
- Set the *angle* of "rotate" to "90", the *axis* should already be set to "x" as well as the *axis origin* "[0,0,0]".
- Select the *image curve* and insert "rotate" into the *image transformation.*
- Create an *image curve* of "WL" and call it "WL_rot".
- Set the *image transformation* of "WL_rot" to "rotate" and the *pattern* to "dash line".
- Do the same with "FOB".
- Create a *scope* and call it "Functions".
- Select everything in the CAD tree except the "Parameters" *scope* and put it into the "Functions" *scope.*



8 Feature Definition for the Aft Surface

We will now create a feature definition for a hull section with which we can create a curve engine and a surface later on.

- To create a *feature definition* select *Features > New Definition* at the top of the GUI.
- In the new window set the *type name* to "Aft_section" and the *label* to "section".
- In the *arguments* tab open the drop down menu under *type* and select *fdouble* .
- Set the name to "xpos".
- Do the same for "Cpc", "Fos", "height", "Fob" and "Beam".

	Type	Name
1	FDouble	xpos
2	FDouble	Cpc
3	FDouble	Fos
4	FDouble	height
5	FDouble	Fob
6	FDouble	Beam

Now we have specified which inputs the *feature* will get. We can now use these inputs to write the *feature definition's function*.

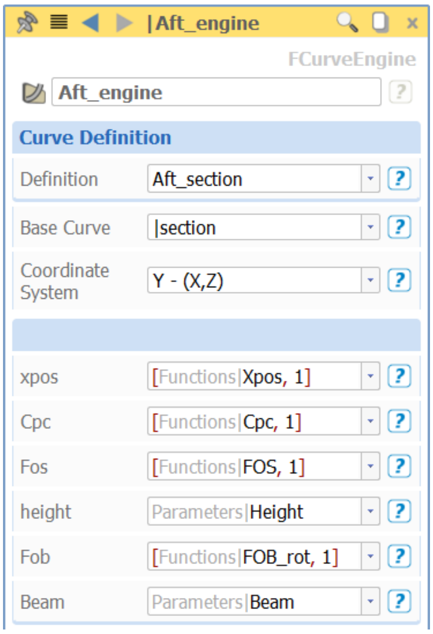
- We write the command sequence that creates a frame via points and curves.

```
point p1([xpos,0,Cpc])
point p2([xpos,Fob,Cpc])
point p3([xpos,Beam/2,Fos])
point p4([xpos,Beam/2,height])

line c1(p1,p2)
line c3(p3,p4)
fsplinecurve c2(p2,p3){
        .setActivePlaneYZ()
        .setStartTan(0)
        .setEndTan(90)
}
polycurve section([c1,c2,c3]).setParametrization("unit speed")
```

- You can now click on *apply* and then close the *feature definition editor* if no warnings are shown.

9 Curve Engine and Meta Surface for the Aft Surface

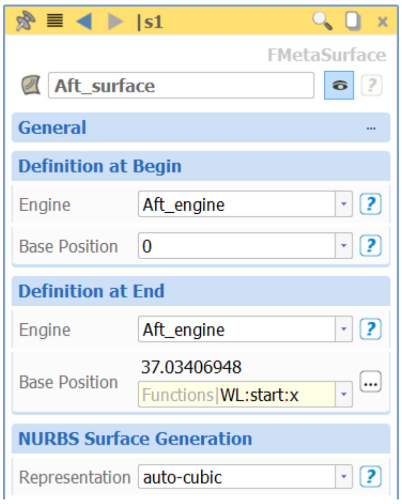
The next step is to create a *curve engine* that will connect the section *feature definition* to the curves we created earlier. With the *curve engine* it will be possible to create a *meta surface* that will model the aft part of the ship following our *feature definition* “aft section”.

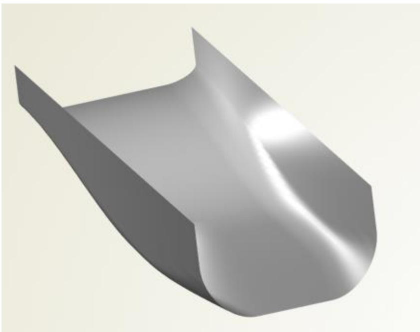


- ▶ Create a *curve engine* via *CAD > Curves > Curve Engine.*
- ▶ Name the *curve engine* "Aft_engine".
- ▶ Select "Aft_section" as the *definition.*
- ▶ Set "section" as the *base curve.*
- ▶ Select "Y - (X,Z)" as the *coordinate system.*
- ▶ Drag and drop the "xpos" line into "xpos".
- ▶ Set the "CPC" curve as "Cpc".
- ▶ Set the "FOS" curve as "FOS".
- ▶ Use the "Height" *parameter* as height.
- ▶ Select the "FOB_rot" *image curve* for "Fob".
- ▶ As "Beam" select the "Beam" *parameter.*

Based on the *curve engine* it is now possible to create a meta surface.

- ▶ Create a *meta surface* via *CAD > Surfaces > Meta Surface.*
- ▶ Change the name to "Aft_surface".
- ▶ Use "Aft_engine" as the *definition at begin* and *end*
- ▶ Set "0" as the start *base position.*
- ▶ The end position shall be at the start of the WL. Therefore write "Functions|WL:start:x" as an end *base position.*





10 Feature Definition for the First FWD Surface

We now want to create another *feature definition* to be able to generate a second *meta surface* for the first part of the fwd ship.

- Create a new *feature definition* named "Fwd_section1" and labelled "section"
- As *arguments* set the following as *fdouble*

  xpos, Cpc, Fob, Fos, deck, height, WL, draft, tanwl.

General | Feature Definitions | Arguments

		Type	Name
1	⊗	FDouble	xpos
2	⊗	FDouble	Cpc
3	⊗	FDouble	Fob
4	⊗	FDouble	Fos
5	⊗	FDouble	deck
6	⊗	FDouble	height
7	⊗	FDouble	WL
8	⊗	FDouble	draft
9	⊗	FDouble	tanwl

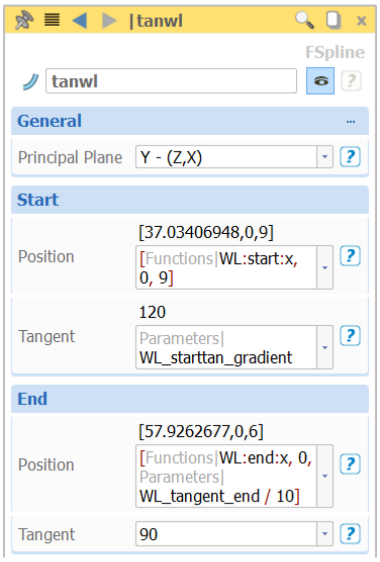
- Write the function as follows.

```
point p1([xpos,0,Cpc])
point p2([xpos,Fob,Cpc])
point p3([xpos,WL,draft])
point p4([xpos,deck,Fos])
point p5([xpos,deck,height])

line c1(p1,p2)
line c4(p4,p5)

fsplinecurve c2(p2,p3){
        .setActivePlaneYZ()
        .setStartTan(0)
        .setEndTan(tanwl)
}
fsplinecurve c3(p3,p4){
        .setActivePlaneYZ()
        .setStartTan(tanwl)
        .setEndTan(90)
}

polycurve section([c1,c2,c3,c4]).setParametrization("unit speed")
```

11 Curve Engine and Meta Surface for the First FWD Surface

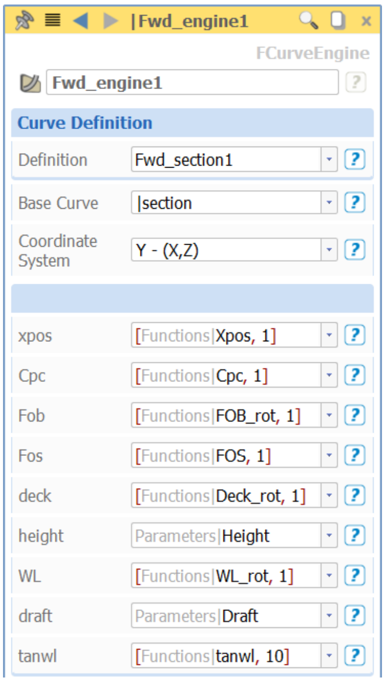
We now want to create the *curve engine* and *meta surface* for the start of the fwd ship. Before we begin we want to create another curve that will control the tangent of the sections at the WL.

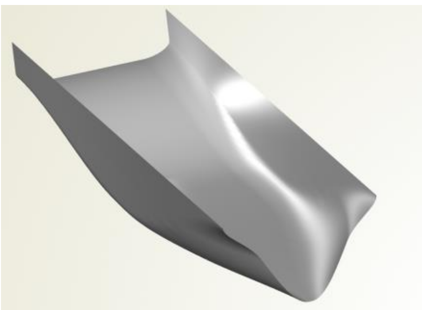
- Create a *parameter* called "WL_starttan_gradient" with the *value* "120".
- Create another *parameter* called "WL_tangent_end" and set the *value* to "60".
- Create an *f-spline* and name it "tanwl".
- Change the entries as can be seen in the picture.
- Set the *colour* to "dodgerblue" and the *pattern* to "dash line".
- Move "WL_starttan_gradient" and "WL_tangent_end" into the "Parameters" *scope* and "tanwl" into “Functions”.



Now we got everything to create another *curve engine* and *meta surface.*

- Create another *curve engine* named "Fwd_engine1" and set all inputs as in the picture.
- Change the number after "tanwl" to "10" instead of "1", because we want to multiply the value that the "tanwl" curve provides by ten.
- Create a meta surface named "Fwd_surface1".
- Set the *base position* at the start to "Functions|WL:start:x".
- Set the *base position* at the end to "Parameters|FOS_length".





12

Feature Definition for the 2nd FWD Surface

We will now create the next *meta surface*, which will be the connection between the previous created forebody surface and the stem surface.

- Create a new *feature definition* named "Fwd_section2" and labelled "section".
- Define the following *arguments* as "FDouble"

    xpos, Cpc, Fob, deck, height, WL, draft, tanwl, tandeck.

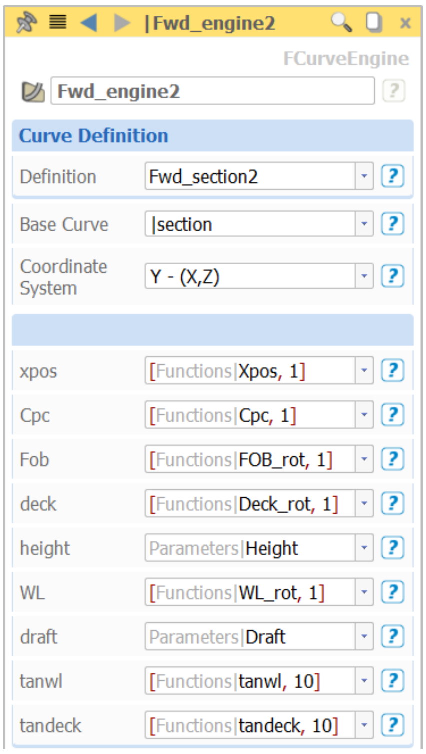
- Write the function as follows.

```
point p1([xpos,0,Cpc])
point p2([xpos,Fob,Cpc])
point p3([xpos,WL,draft])
point p4([xpos,deck,height])

line c1(p1,p2)

fsplinecurve c2(p2,p3){
        .setActivePlaneYZ()
        .setStartTan(0)
        .setEndTan(tanwl)
}
fsplinecurve c3(p3,p4){
        .setActivePlaneYZ()
        .setStartTan(tanwl)
        .setEndTan(tandeck)
}

polycurve section([c1,c2,c3]).setParametrization("unit speed")
```

13 Curve Engine and Meta Surface for the 2nd FWD Surface

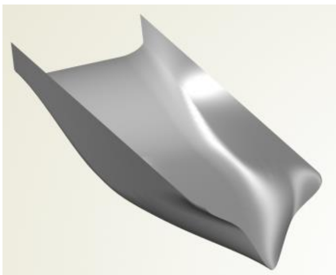
Before we create the *curve engine* and *meta surface* we again need a new control curve, that this time will vary the tangent of the section at the deck.

- ▶ Create a *parameter* called "Deck_starttan_gradient" and set the *value* to "110".
- ▶ Create another *parameter* called "Deck_tangent_end" and set the *value* to "50".
- ▶ Create a *f-spline* named "tandeck".
- ▶ Change the x coordinate at the start to "Parameters|FOS_length" and the z coordinate to "9".
- ▶ Change the x coordinate at the end to "Parameters|Length", the y coordinate to "0" and the z coordinate to "Deck_tangent_end/10".
- ▶ Set the start tangent of "tandeck" to "Deck_starttan_gradient" and the end *tangent* to "90" referring to the "Y - (Z,X)" *plane.*
- ▶ Set the *colour* to "lime" and the *pattern* to "dash line".
- ▶ Move "tandeck" into the "Functions" *scope.*
- ▶ Move the *parameters* "Deck_starttan_gradient" and "Deck_tangent_end" into the "Parameters" *scope.*



We can now finish the *meta surface* after creating a *curve engine.*

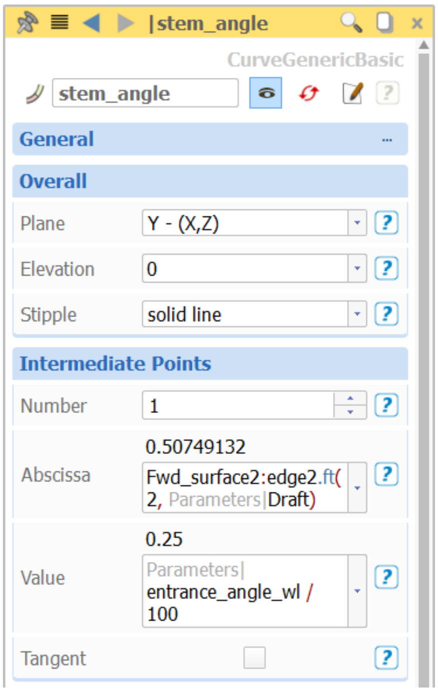
- ▶ Create a curve engine named "Fwd_engine2" and set the inputs as can be seen in the picture above.
- ▶ Create a *meta surface* named "Fwd_surface2".
- ▶ Set the start *base position* to "Parameters|FOS_length".
- ▶ Set the end *base position* to "Functions|CPC:inter2:x", which is the x position of the second *intermediate point* of "CPC".

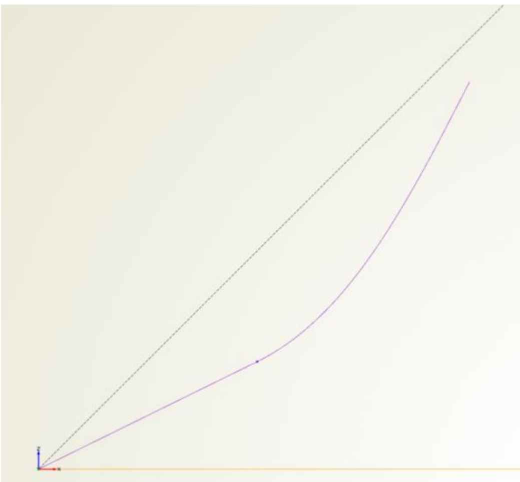


14 Stem Angle Distribution

We will now create a function to control the distribution of the entrance angle along the stem. In this curve, we want to have a control point that represents the WL position, so we can set the angle to the same parameter as we did for the WL curve.

- Create a *multi-segmented smooth curve* named "stem_angle".
- Enable one *intermediate point.*
- Set the *value* of the *intermediate point* to "Parameters|entrance_angle_wl/100" in order to get the entrance angle we need.
- Set the *abscissa* to "Fwd_surface2:edge2.ft(2, Parameters|Draft)"; this command will find the specific t value of the *surface edge* from Fwd_surface2 corresponding to the z position "Draft".
- Set the *abscissa* at the end to "1" and the *value* to "0.9".
- Change the *colour* of "stem_angle" to "darkorchid".
- Move "stem_angle" into the "Functions" *scope.*





15

Stem Surface

We can now create a surface at the stem. For this purpose we will use an existing feature.

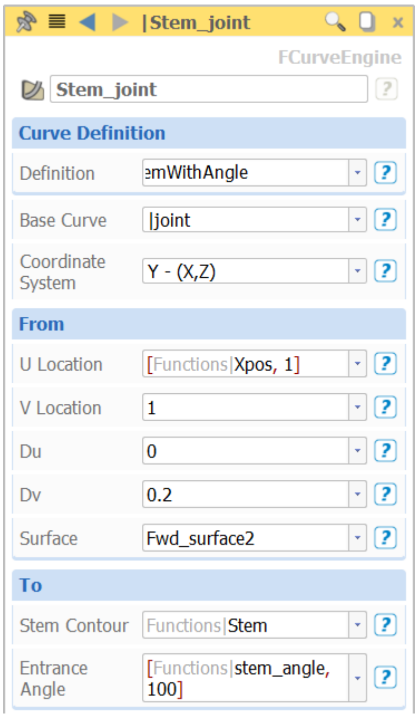
- Create new feature definition.
- Click on the *Reload*.
- Choose the feature definition “05_CurveDerivativeJointToStemWithAngle” which is located in *CaesesDirectory/tutorials/ 06_Hull_Design*.
- Change the *Type name* to CurveDerivativeJointToStemWithAngle.
- Select *Apply* and *Close.*

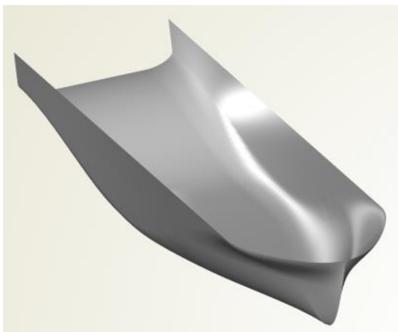
To be able to use the *feature* we will need a stem contour. As we only have a curve describing the complete center plane contour we need to create a shortened curve from Cpc.

- Create an *image curve* and name it "Stem".
- As *source* select "Functions|CPC".
- Change the *domain* start from "0" to "Functions|CPC.ft(0, Functions|CPC:inter2:x)" (this will find the t value of "CPC" for the given x position).
- Move "Stem" into the "Functions" *scope.*

Now we can use the *feature definition* for a *curve engine* and create a *meta surface.*

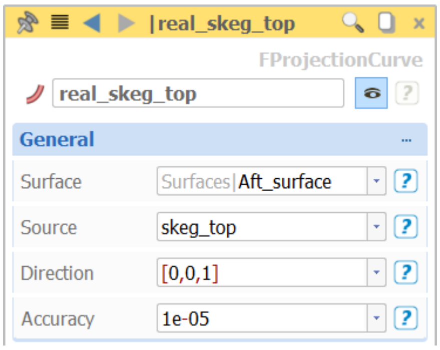
- Create a curve engine that uses the formerly created *feature definition* and name it "Stem_joint".
- Again, set the inputs shown in the picture above.
- Create a *meta surface* that uses "Stem_joint" as *definition* and call it "Stem_surface".
- To fit "Stem_surface" and "Fwd_surface2" together they need a similar number of curve points in the z direction, therefore change the number of *interpolated points in surface direction* of "Stem_surface" to "30".





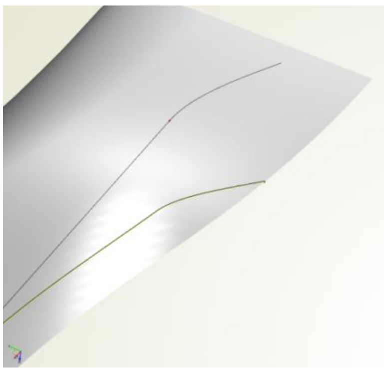
16 Creating the Upper Skeg Contour

As the basic hull is now complete, we can start to create a skeg. We will start by creating a upper contour, which will be projected onto the aft surface.



- First, create a *scope* called "Engines" and put all *curve engines* into that *scope.*
- Do the same for all surfaces with a *scope* called "Surfaces".
- Create a *parameter* called "skeg_startx_top" and set its *value* to "5".
- Create a second *parameter* with the *value* "2" and name it "skeg_width_top".
- Create a *multi-segmented smooth curve* named "skeg_top" and set the *elevation* to “-1”.
- Set the number of *intermediate points* to "1".
- Set the start *abscissa* to "skeg_startx_top".
- Set the *intermediate point abscissa* to "skeg_startx_top +3" and *value* to "skeg_width_top/2".
- Enable the *intermediate point tangent* and give it the value "0".
- Set the end *abscissa* to "Parameters|Length/3" and the end *value* to "skeg_width_top/2".
- Create a *projection curve* via *CAD > Curves > Projection Curve.*

- Rename the *projection curve* to "real_skeg_top".
- As the surface use "Surfaces|Aft_surface".
- Set the source to "skeg_top".
- The direction should be set to z (“[0,0,1]").



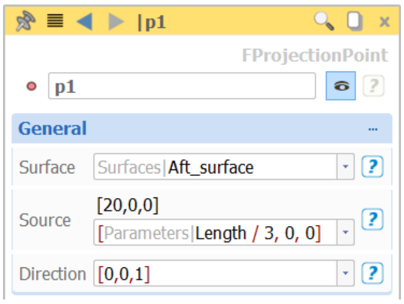
17

Creating the Lower Skeg Contour

Now we will create the bottom skeg contour similarly to the top, but this time not projected. Also, we will create a *projection point* that will be used for an *image surface* later.

- Create a *parameter* "skeg_startx_bottom" with the value "6.5".
- Create a *parameter* "skeg_width_bottom" with the value "1".
- Create a *multi-segmented smooth curve* named "skeg_bottom".
- Set the number of *intermediate points* to "1".
- Set the start *abscissa* to "skeg_startx_bottom".
- Set the *intermediate point abscissa* to "skeg_startx_bottom +2" and *value* to "skeg_width_bottom/2".
- Enable the *intermediate point tangent* and give it the value "0".
- Set the end *abscissa* to "Parameters|Length/3" and the end *value* to "skeg_width_bottom/2".
- Create a *line* and name it "mid_skeg".
- Set the start of the *line* to "[skeg_startx_bottom, 0, 0]".
- Set the end of the *line* to "[Parameters|Length/3,0,0]".

We now want to project the end point of the *line* onto our "Aft_surface".

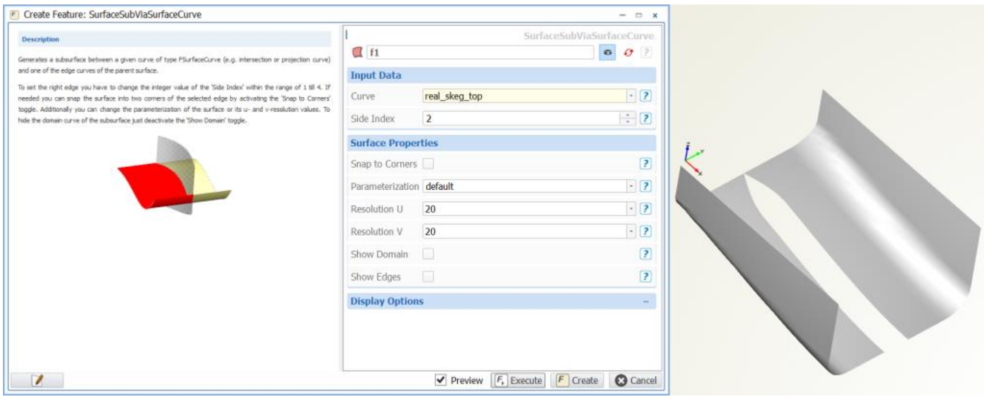


- Create a *projection point* via *CAD > Points > Projection Point.*
- As *surface* set "Surfaces|Aft_surface".
- As *source* type "[Parameters|Length/3,0,0]".
- Give z as the direction via "[0,0,1]".
- Set "Aft_surface" invisible.
- Move all *parameters* into the “Parameterss” *scope.*
- Create a *scope* and call it “skeg”.
- Move all skeg lines and “p1” into the “skeg” *scope* and then move “skeg” into the “Functions” *scope.*

18 Cutting out the Skeg Contour from the Aft Surface

As we want the skeg to be an integrated part of the ship we will have to trim the aft surface. We will do this with *sub surfaces*.

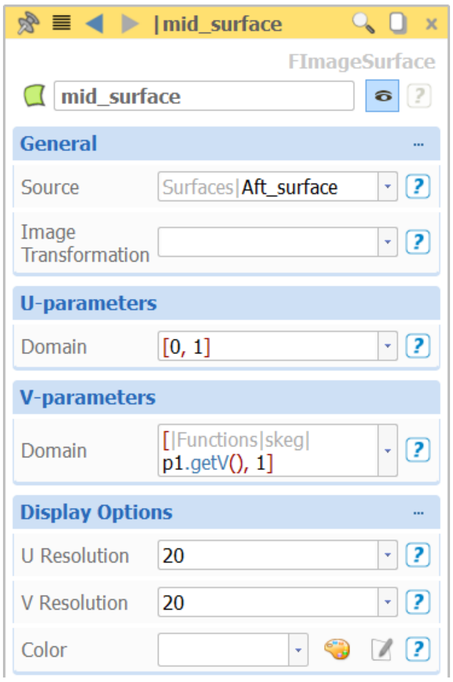
- Create a *sub surface* from a *surface curve* via *CAD > Surfaces > More (under Sub Surface) > From Surface Curve.*
- As Curve use the *projection curve* “real_skeg_top” and as *side index* "2".
- Press the create button at the bottom.



- A new *feature* "f1" was created. Rename it into "aftsub1" and move it into the "Surfaces" *scope.*
- Create another *sub surface* from a *surface curve* (use the new *feature definition* “SurfaceSubViaSurfaceCurve”), but this time use the the first edge of the first *sub surface* "aftsub1:D1" as the *projection curve* and as the *side index* "1".
- Rename the *feature* into "aftsub2".
- Move "aftsub1" and "aftsub2" into the "Surfaces" *scope.*

19 Connecting the Aft Ship with the FWD Ship

Now we have cut out the skeg contour, but we need to connect the aft and fwd part again. We can use the *projection point* we created earlier together with an *image surface.*



- Create an *image surface* via *CAD > Surfaces > Image Surface.*
- Change the name to "Mid_surface".
- Set the *source* to "Surfaces|Aft_surface".

To define the starting position of the surface, we use the projection point in the *v-parameters domain.*

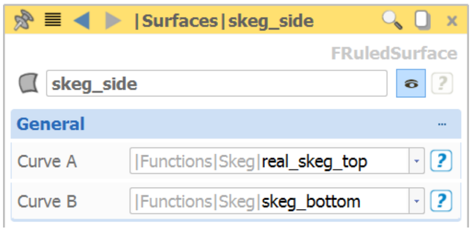
- Put the *projection point* into the first entry of the *v-parameters domain* and set the command ".getV()" behind it.
- Move "Mid_surface" into the "Surfaces" *scope.*

20

Finishing the Skeg

Now we have trimmed the hull with the skeg. As a final step we can create the skeg surface to finish the model.

- Create a *ruled surface* via *CAD > Surfaces > Ruled Surface.*
- Name the surface "skeg_side".
- As *curve a* select the *projection curve* "real_skeg_top".
- As *curve b* select the bottom skeg contour.



✓ The surface will look a little deformed. If you enable the *wireframe* in the 3D-view, you can see that the surface is not created evenly. The reason is that we use a *polycurve* in the "CurveGenericBasic" feature, which is unevenly parameterized. To change this, you can open the feature definition of “CurveGenericBasic” and select the *create function* tab. Scroll down to the last line and add ".setParametrization("unit speed")" behind "curve.setStipplePattern(stip)".

Now we will close the bottom of the skeg with another *ruled surface.*

- Create a *ruled surface* and connect the bottom "skeg_bottom" with the mid line "mid_skeg".
- Rename the *ruled surface* into "skeg_bottom_surface".
- Move "skeg_bottom" and "skeg_side" into the “Surfaces” *scope.*