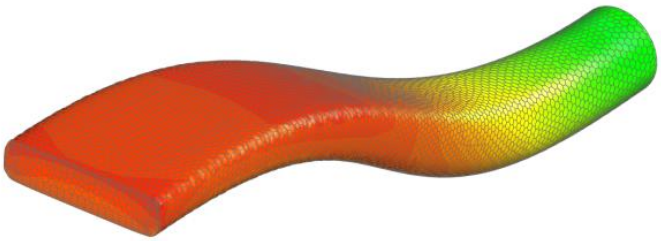
Coupling CAESES and STAR-CCM+: S-Duct Example

The purpose of this tutorial is to learn the integration of a 3rd party CFD software, in this case STAR-CCM+. You will use an existing s-duct geometry, prepare it for the export and connect it with STAR-CCM+.

The CFD setup in STAR-CCM+ is created in batch mode with JAVA macros. These macros can be recorded in STAR-CCM+ and modified with e.g. NetBeans.

All you need is a basic understanding of how geometry gets created in CAESES. Some feature definitions are used as well in this model; see the feature definitions tutorials (and videos) for more information.



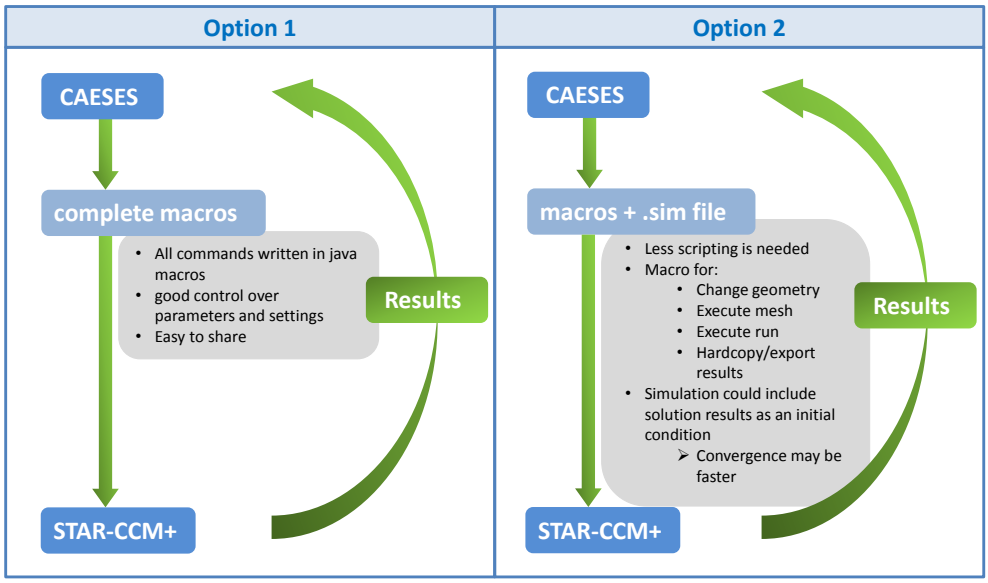
The coupling is done with STAR-CCM+ 9.06. The integration part can be done with CAESES versions > 3.1.1.

For any further questions please use the CAESES forum.

1

Options to couple CAESES with STAR-CCM+

To control STAR-CCM+ in batch mode it is necessary to write or record java macros, which have to be included in the software connector of CAESES. The following flowchart shows the two ways of coupling STAR-CCM+ with CAESES.



One way is creating the complete setup with java macros. Therefore, you have to record the macros in STAR-CCM+, while setting up the complete case. This could be difficult in very complex cases. That's why another option of coupling STAR-CCM+ to CAESES is to set up the whole case in STAR-CCM+ without recording macros. When the case is completed, you only have to record or create macros which change the geometry, create the mesh, start the simulation and export the results.

In the software connector you have to provide the macro files and the simulation file. Another advantage of this way is that you could have a solution in the simulation file. This solution can be used as an initial condition hence the simulation will converge faster.

In this tutorial we explain option 1 with complete macros. In step 13 you can see the outlook of the software connector with macros and a simulation file (option 2).

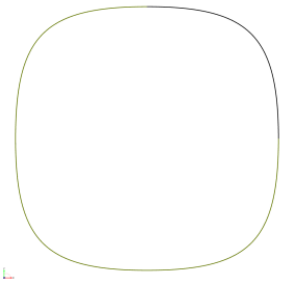
2 Open the S-Duct Model

The model from which we will start can be found in the documentation browser:

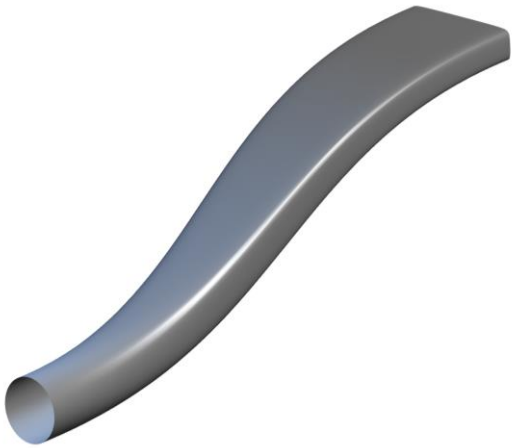
- Open the project *samples > s-duct geometry variation*.
- Save it as a new project.

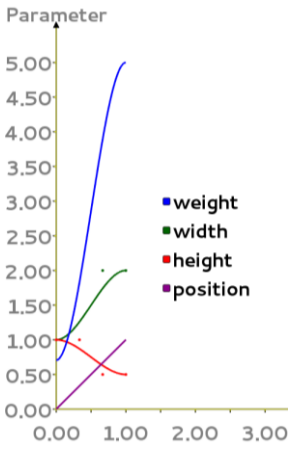
Here is a very brief overview of this model, before we continue with the integration:

The first step of modeling the s-duct is to define the shape of the cross section. This is done in the first scope by using a NURBS curve with different transformations. The position of the cross section depends on the path. From these objects a feature definition is created, which will be the input for the curve engine. From the curve engine, a meta surface gets finally created, which sweeps the cross section along the path.



Different functions define the parameters of the cross section along the path, which are additional inputs for the curve engine. The functions are usually smooth B-Spline curves, where the x-position defines the curve path's parameter position (0 to 1) and the y-position the value of the parameter (weight, width, height and position). These y-values can be controlled by design variables. You can move the points of the functions in y-direction, to see how the model changes.

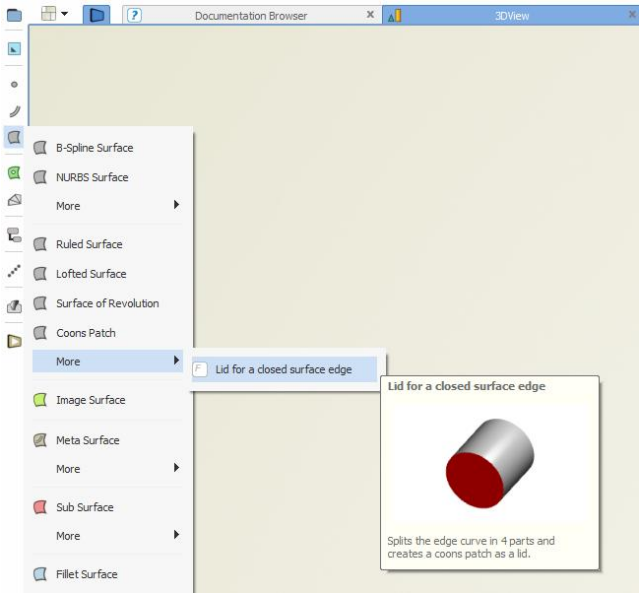


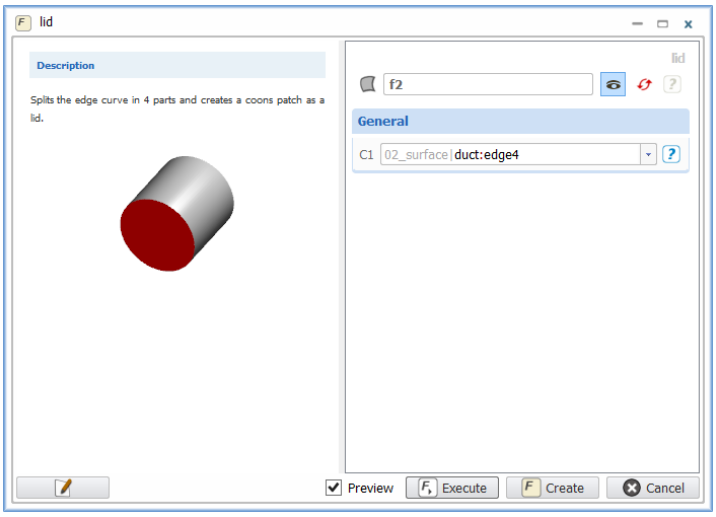


3 Prepare Geometry for Export: Lids

In order to export the geometry as a watertight STL file later on, we have to close the inlet and outlet and create a trimesh.

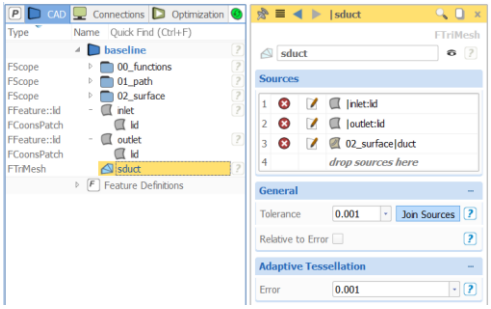
- Go to menu and select *CAD > surfaces > more (Coons Patch) > lid for a closed surface edge*.
- Rename the feature to “inlet”.
- Select the circular edge of the duct surface as the source for *C1.*
- Press *create.* (It is also possible to create the feature first, and then select the input curve.)
- Repeat this step for the outlet surface.





4 Prepare Geometry for Export: Trimesh

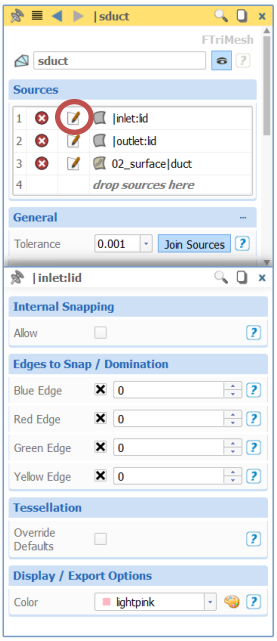
Now we can create a trimesh from the geometry components.



- Create a trimesh via *CAD > meshes and solids > trimesh* and call it "sduct".
- In the object tree, expand the inlet surface and drag & drop the "lid" surface into the trimesh sources.
- Do the same for the outlet surface.
- Drag & drop the duct surface into the trimesh sources as well.
- Set the *Tolerance* and the *Adaptive Tesselation* of the trimesh to "0.001".

Now you can see the mesh for the STL export. By increasing or decreasing the *Adaptive Tesselation*, you are able to capture more or less details of the geometry in the resulting STL file.

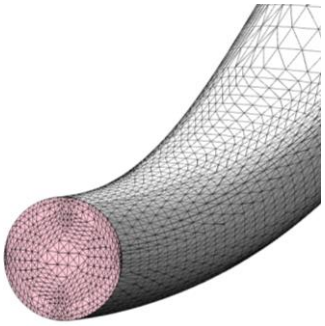
Now we want to insert different colors for different surfaces, so that the STL file consists of different patches.



- Press the edit button  next to the "|inlet:lid" entry of the trimesh sources.
- Change the color to "lightpink".
- Change the color of the outlet lid to "purple".
- Change the color of the duct to "darkgray", and turn on internal snapping!

Since the geometry is created in a normalized way (the diameter of the circle is 1), we have to scale the mesh to the real dimensions.

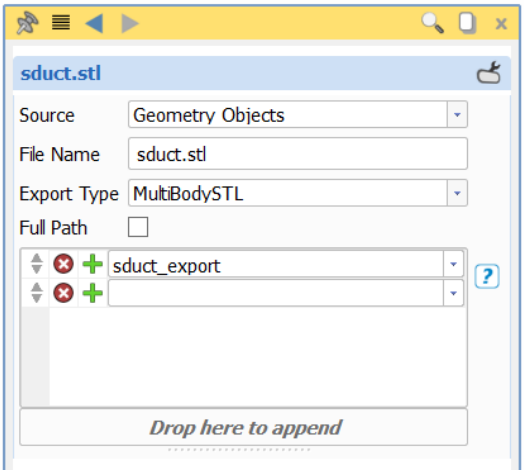
- Select the *trimesh* "sduct".
- Create an *image trimesh* via *CAD > meshes and solids > image trimesh* and set the name to "sduct_export".
- Create an *image transformation* by *CAD > transformations > scaling.*
- Use the scaling factor 0.1 for all dimensions.
- Set the scaling as image transformation at "sduct_export".



5 Software Connector: Import

We can import a prepared *software connector* which includes the STAR-CCM+ files. It was configured before (for the purpose of this tutorial) and saved for re-use. The software connector is the widget where external tools can be plugged-in.

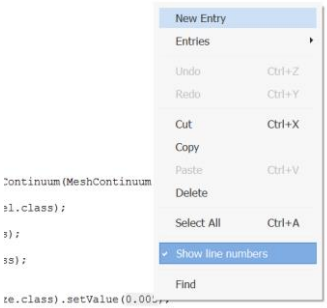
- Choose *connections > software connector from file*.
- Select the file "05_SDuct_with_STAR-CCM+_softwareconnector.xffl" that you will find in the installation directory *> tutorials > 08_integrations*.
- Drag and drop the trimesh "sduct_export" into the *Input Geometry* window of the software connector.
- Click on this file, and specify the export name and the type, respectively. Set the export type to *MultiBodySTL*. Set the name to "sduct.stl".



6 Introducing Parameters

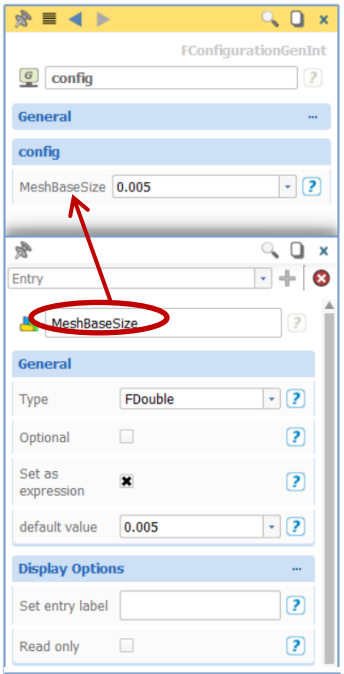
In this step, we introduce some parameters, which control STAR-CCM+ settings. First, it is useful to add a parameter which scales the mesh uniformly in all directions.

- Double-click on the "macro_02_mesh.java" file.
- Switch to plain mode by clicking on the little pen  on the top left of the file.
- Go to the "Base Size" value (line 38). Mark the value "0.005" which specifies the base size and choose *new entry* from the context menu.




```
36
37  //Base size
38  meshContinuum_0.getReferenceValues().get(BaseSize.class).setValue(<entry>meshContinuum_</entry>);
39
```

- In the properties tab (click on entry name, see screenshot), rename this entry to "MeshBaseSize".
- Select the value "0.005" in the object editor and *right click > create parameter*. Change the name to "MeshBaseSize".

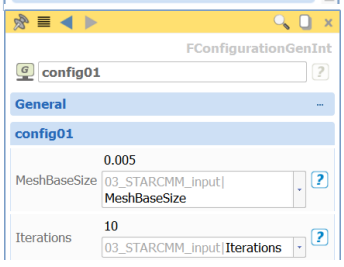


You can find the parameter in the *object tree*.

- Select the tab "macro_05_run.java".
- Go to maximum steps. Mark the number "200" (line 25) which specifies the number of iteration.

```
23
24  //set maxiumum steps
25      stepStoppingCriterion_0.setMaximumNumberSteps(200);
26
```

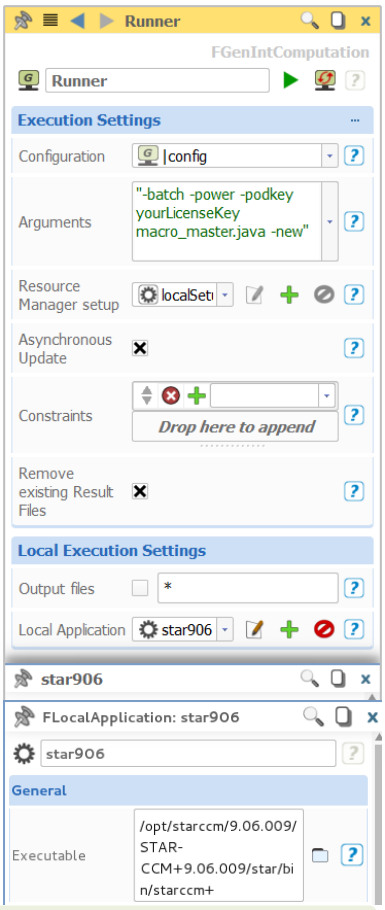
- Do *right click > new entry* and rename this entry to "Iterations".
- Select the value "200" in the object editor and *right click > create parameter*. Change the name to "Iterations".
- Set the value of "Iterations" to "10".
- In the object editor, mark both parameters and create a new scope.
- Rename this scope to "03_STARCCM_input".



7 Software Connector: Computation

For post-processing, we need to have result files and values, which will be inserted in the *Result Files* and *Result Values* window of the software connector. In order to get these, we have to trigger a first run – either with CAESES, or externally. Since we have already changed parts of the input files in the previous steps, we trigger the computation from within CAESES. For this, we have to set up the *computation*:

- Click on "Runner" in the center of the software connector.
- In the object editor of the selected object "Runner", activate "Remove existing Result Files". This ensures that with each run the previous simulation files get deleted automatically.
- For Linux set the arguments to: *"-batch -power -podkey yourLicenseKey macro_master.java -new"* ,also shown in the picture. For Windows see the green box at the bottom. In this case we use a POD (Power on Demand) license option.
- Create a new "Local Application" by clicking on the "+" button next to the attribute, and give it a meaningful name.
- As *executable* select the executable of STAR-CCM+.
- Run the computation by clicking on the run button (▶).
- While the computation is running, you can check the output in the *Task Monitor*.



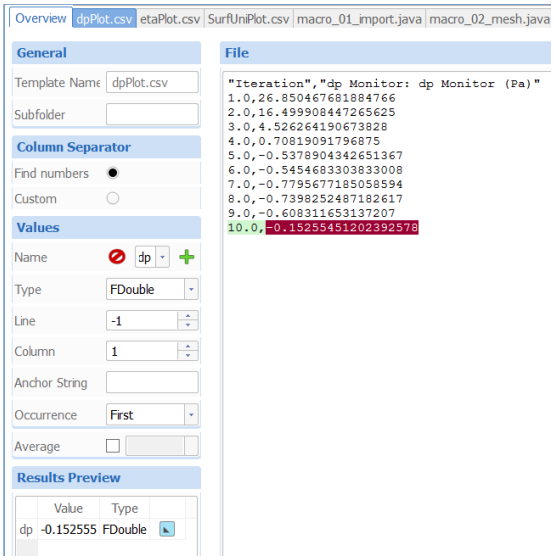
When the computation is done, check the results on your hard disc, to see how the results are handled by CAESES: A new folder was created, with the name of the current project file *.fdb. In this folder all results can be found.

✓ The order of the arguments of the *Runner* is very important. For further information see the STAR-CCM+ documentation for the batch mode. Arguments settings for Windows: "-batch macro_master.java -power -podkey YourLicenseKey -new".

✓ If the computation does not run, check the error message in the task monitor or the "stderroroutput.redirect" files in your baseline folder.

8 Result Values

In order to assess the simulation, we need to have result files which provide e.g. the pressure drop. These values can be extracted from any text files, but usually from *.csv or *.dat file formats. Since CAESES needs to know where the desired values are located in each design directory, we have to provide an example file.



- In your file explorer, go to the baseline directory which was created during the last run and check for the "dpPlot.csv", "etaPlot.csv" and "SurfUniPlot.csv" file.
- Add these files to the *Result Values* window of the software connector by using either drag & drop, or by using the "+" button at the window.
- Double-click on "dpPlot" in the software connector to show the content.
- Add a value by clicking on the "+" button and set the name to "dp".
- Set *line* to "-1" (in this case, this means: always the last row of the file).
- Set column to "1".
- Now create a parameter by pressing the blue parameter symbol in the results preview. This parameter will be the objective of the simulation at a later stage.
- Do the same for the "etaPlot.csv" and the "SurfUniPlot.csv" files.

✓ Remember that this parameter is the result value of the last iteration or time step. If you have a strong oscillating result, you might average the values over a specific time. This could be done by selecting the average button in the table.

- Select the evaluation parameters, and create a scope. Set the name of this scope to "04_STARCCM_evaluation" (note, names can be user-defined i.e. arbitrary).

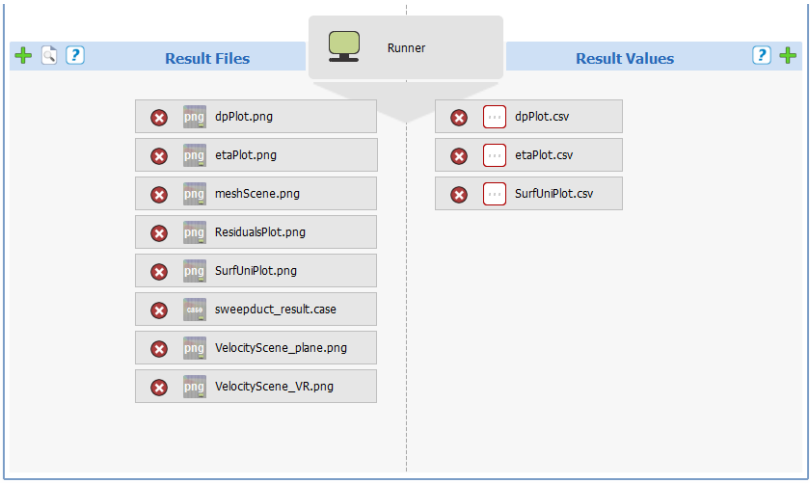
See the following table for a description of the parameter. Note that the v_{in} is the average velocity at inlet; the areas of the inlet and outlet are computed in STAR CCM+ with reports.

Value	Equation
Total pressure drop:	$dp = p_{out} - p_{in}$
Efficiency:	$\eta = \frac{C_p}{C_{pi}}$
Static pressure recovery coefficient:	$C_p = \frac{d_p}{0.5\ \rho\ v_{in}^2}$
Ideal pressure recovery coefficient:	$C_p = 1 - \left[\frac{A_{in}}{A_{out}}\right]^2$

9 Result Files

In the *Result Files* window of the software connector, output files from the CFD calculation are referenced. Typical output files are pictures, tables, text files and CFD solutions.

► Add the following files from *manual_results/baseline/Runner* to this window, either by using drag & drop, or by using the "+" button in the corner.



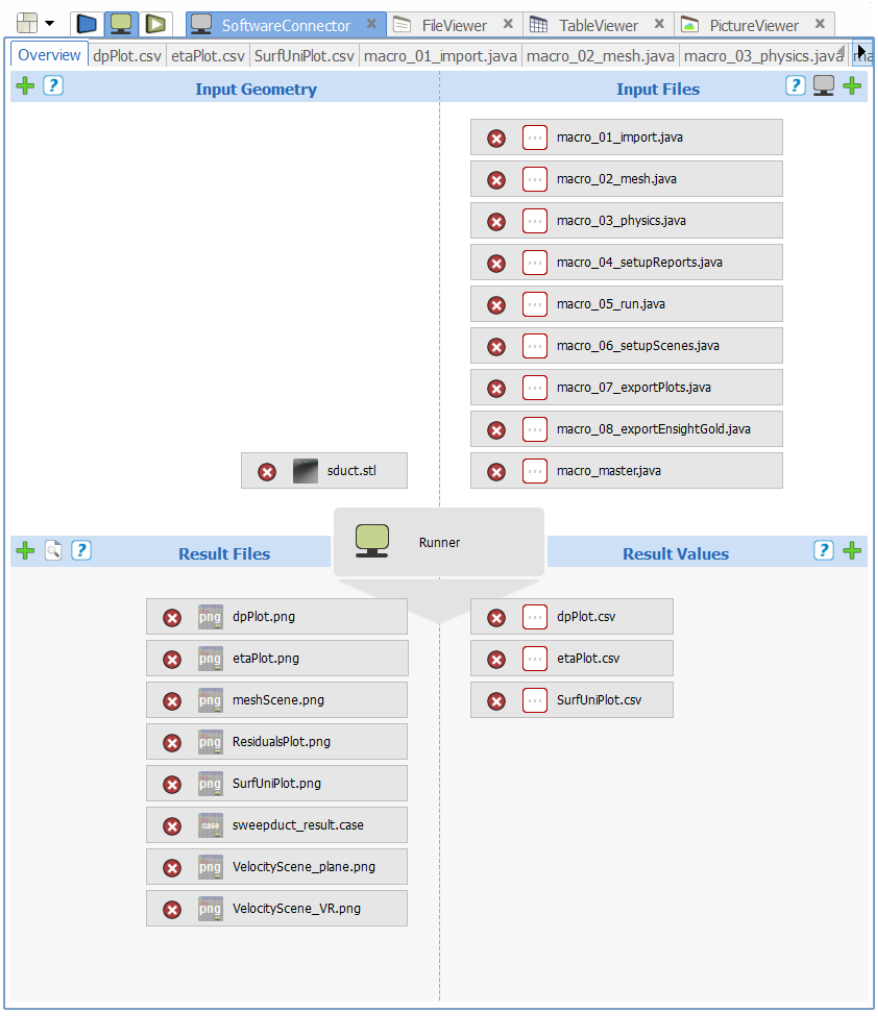
✓ For the visualization of the mesh, scalar and vector fields the supported exporting format from STAR-CCM+ is the *EnsightGold* format. These files have the ending *.case.

You can import the plots which are produced by STAR-CCM+ or you could also create plots in CAESES. Therefore, you have to use the .csv tables also in the *Result Files* window of the software connector.

10 Running the Case

Now the software connection is completed, and a first test case can be started.

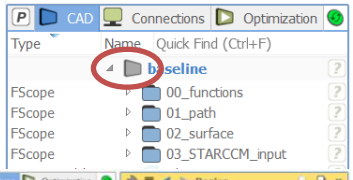
- ▶ Set the parameter “Iterations“ to “200”.
- ▶ Go back to the software connector and run the simulation (again by using the ▶ button).

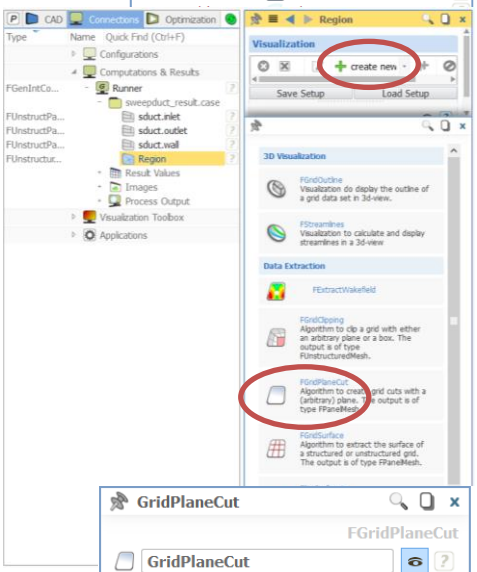


11

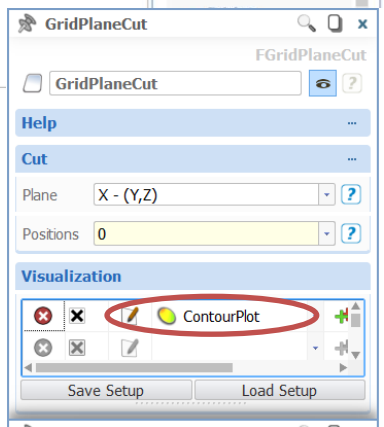
Post-Processing

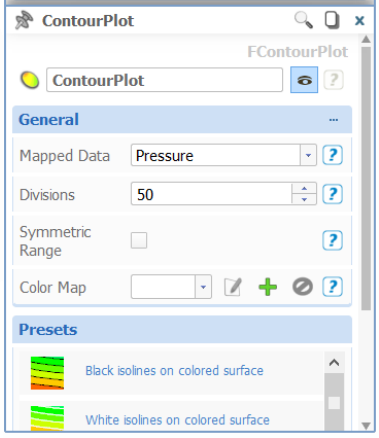
When the computation has finished, we can do some post-processing and visualize the results.



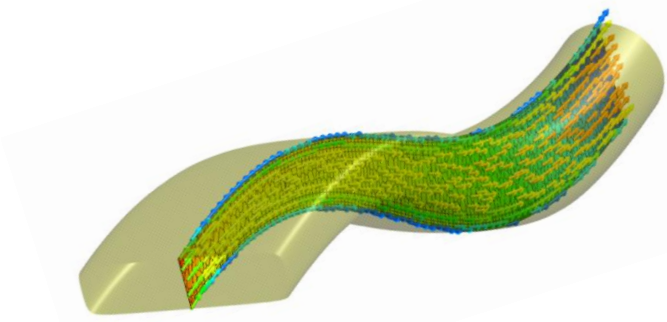


- Go to *CAD* and hide the baseline, by clicking on the baseline icon.
- Go to the *Connections* tab and expand *computations & results > runner*.
- Select sweepduct_result.case > *Region*
- Create a new *Visualization*.
- A new window appears, select *FGridPlaneCut*.
- Set the position to "0".
- Create a new visualization at the bottom of this window.
- Select *FContourPlot*.
- Choose the mapped data you want to visualize, for example *Pressure*. In the 3D window the visualization will appear.
- Increase the number of divisions to "50".





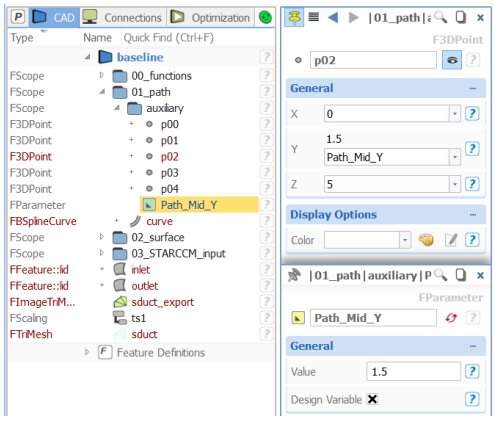
You could also create vector visualization, iso-surfaces or mesh visualizations. Note that it is currently not possible to visualize streamlines from STAR-CCM+ data.



12 Quick Geometry Variation

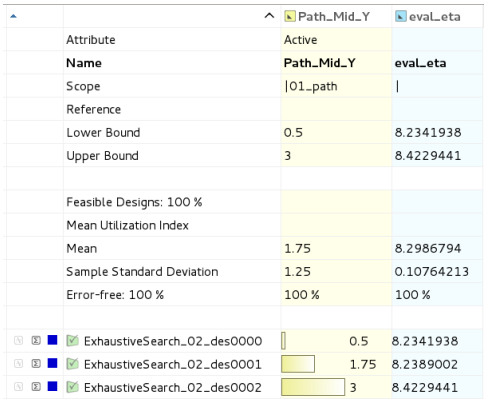
Now we can do a quick geometry variation. Note: with *CAESES Free* it is only possible to create up to 3 designs in a design engine.

First we have to introduce a design variable.



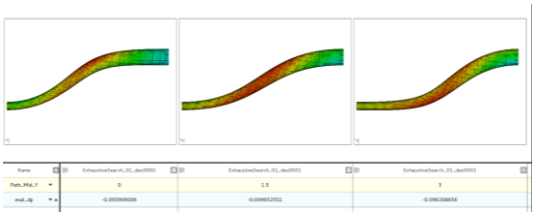
- Go to the scope 01_*path*, select the auxiliary scope and create a design variable for the y-position of the point "p02". This can be done by a right-click on the value.
- Change the name to "Path_Mid_Y".

For geometry variations we have to create a *design engine*. In this case we use the *exhaustive search* algorithm which simply divides the design space of a variable into a specific number of subdivisions.



	Path_Mid_Y	eval_eta
Attribute	Active	
Name	**Path_Mid_Y**	**eval_eta**
Scope	|01_path	|
Reference		
Lower Bound	0.5	8.2341938
Upper Bound	3	8.4229441
Feasible Designs: 100 %		
Mean Utilization Index		
Mean	1.75	8.2986794
Sample Standard Deviation	1.25	0.10764213
Error-free: 100 %	100 %	100 %
ExhaustiveSearch_02_des0000	0.5	8.2341938
ExhaustiveSearch_02_des0001	1.75	8.2389002
ExhaustiveSearch_02_des0002	3	8.4229441

- Create an *exhaustive search* via *optimization > exhaustive search.*
- Set the number of subdivisions to "2".
- Select the design variable "Path_Mid_Y".
- Set the lower range to "0.5" and the upper range to "3".
- Set "Eval_dp", "Eval_eta" and "Eval_SurfUni" as *evaluations.*
- Set up screenshot. Make sure you can see some scalar fields in your 3DView. Press the "+" button next to *screenshots* (in the engine) and select the 3DView as *window*.
- You can now run the study by pressing the run button of the e*xhaustive search*. Note that this will trigger 3 parallel runs by default. If you want to change this setting, you have to uncheck the *asynchronous update* of the computation.
- You can see the status of the running simulations in the *task monitor* window.
- When the variation is done, you can see the results in the table.
- Create a *design viewer* to compare the designs by clicking the button  of the table.



13 Appendix: Software Connection with Macros and Simulation File

In the following image you can see how the software connector would look if you use macros in combination with a simulation file.

It is recommended that you use the absolute path of the simulation file (do not double click on such a file – this would import a copy), therefore the size of your *.fdb* project does not get too large.

In the argument you now have to include the name of the simulation (sduct.sim) file instead of the *–new* option.