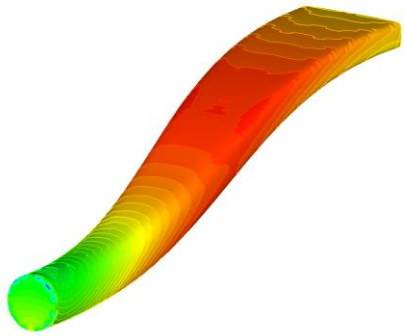
Coupling CAESES® and OpenFOAM: S-Duct Example

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



The CFD setup for OpenFOAM was done with the HELYX OS, an Open Source Graphical User Interface for OpenFOAM. With the help of HELYX OS all required OpenFOAM files can be generated used in CAESES® for geometry variation and optimization.

All you need is a basic understanding of how geometry gets created in CAESES®, but it is not necessary. 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 OpenFOAM 2.2.0. The integration part can be done with CAESES® versions > 3.1.1.

For any further questions you can use the CAESES® forum.

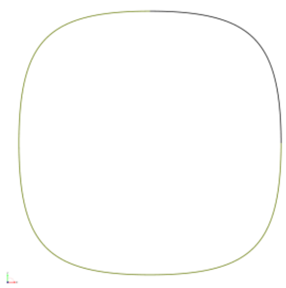
Open the S-Duct Model

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

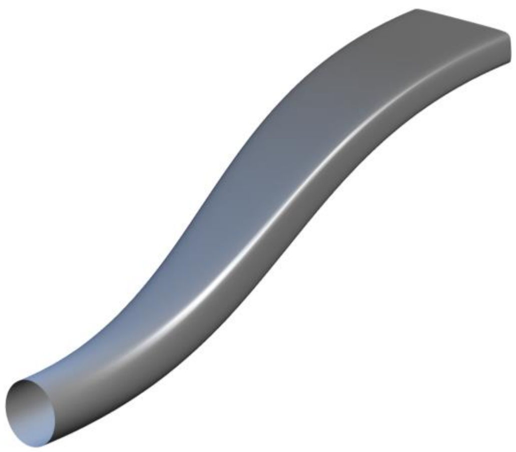
- Open the project *samples > design Engines > s-duct geometry variation.*
- Save it as a new project.

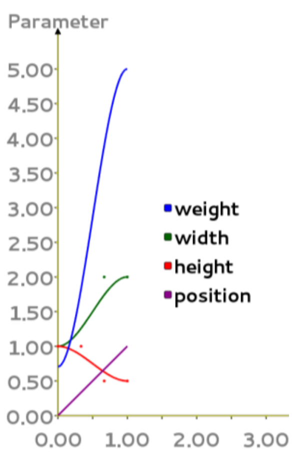
Here is a very brief overview of this model, before we continue with the OpenFOAM integration:

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. 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.



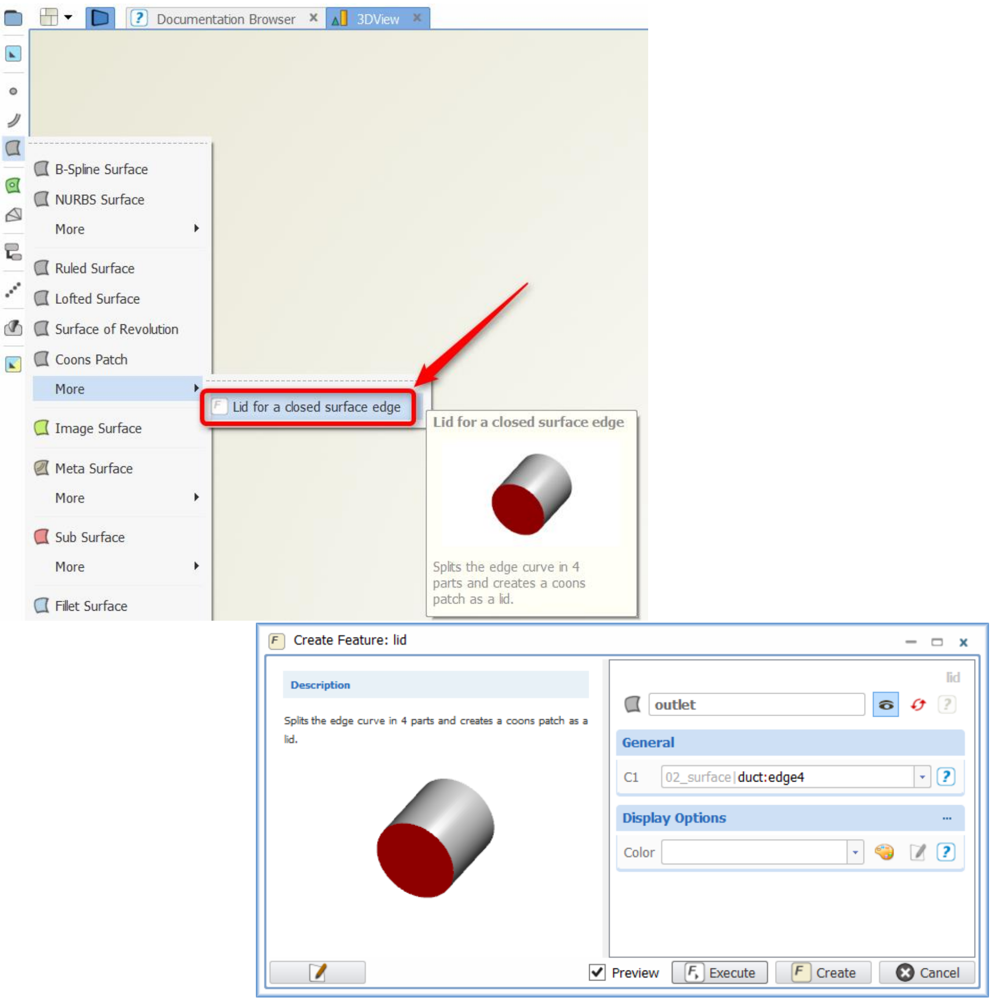


2

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 “outlet”.
- Select the circular edge of the duct surface as the source for *C1*
- Click “Create”. It is also possible to create the feature first, and then select the input curve.
- Repeat this step for the inlet surface.

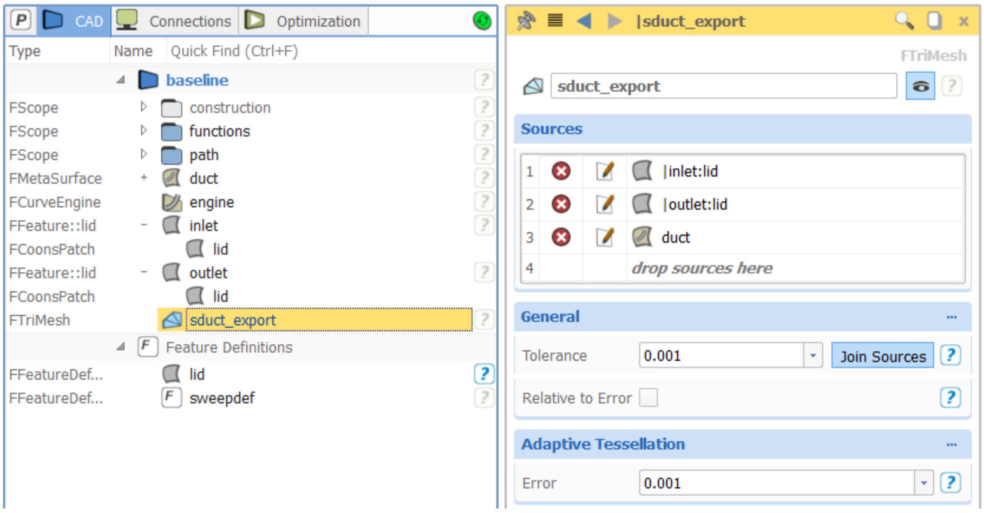


3 Prepare Geometry for Export: Trimesh Part 1

Now we can create a trimesh from the geometry components.

- Create a trimesh via *CAD* > *meshes and solids* > *trimesh* and call it “sduct_export”.
- 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.

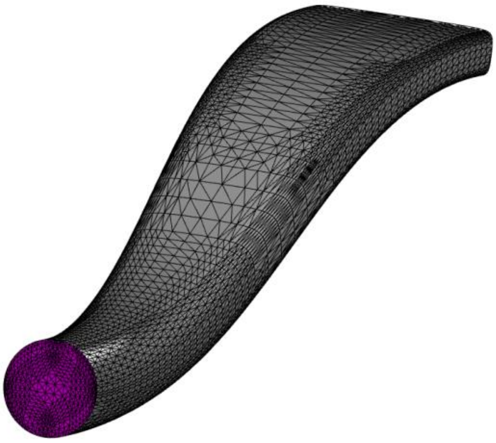


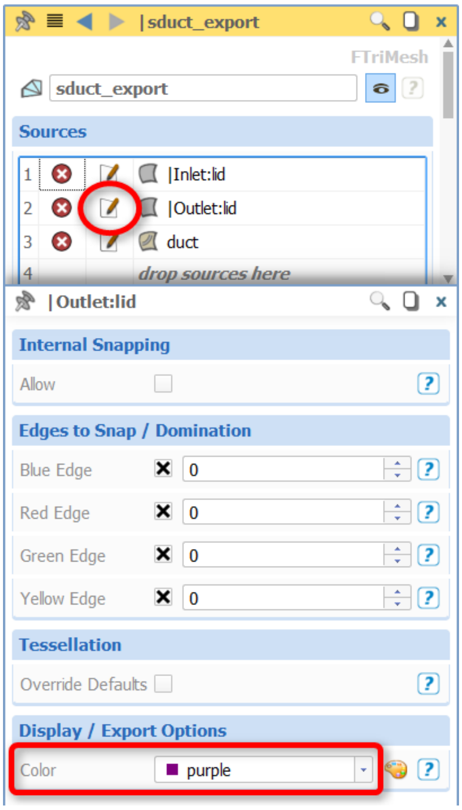
4

Prepare Geometry for Export: Trimesh Part 2

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 "|outlet:lid" to "purple".
- Change the color of the duct to "darkgray".



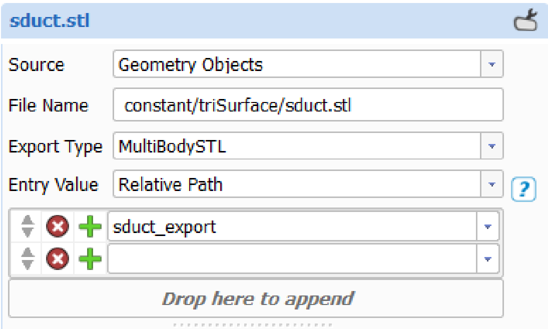


5

Software Connector: Import

Now we can import a prepared *software connector* which includes the OpenFOAM 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.

- Go to the *connections* tab in the pull down menu and choose “Software Connector From File”.
- Select the file “01_SDuct_with_OpenFOAM_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 connector.
- Click on this file, and specify the export name and the type, respectively. Set the export type to “MultiBodySTL”. Since the STL location is in a subfolder of the OpenFOAM project directory, we have to specify the file names as follows: *constant/triSurface/sduct.stl*



✓ By specifying the path *constant/triSurface/myGeometry.stl*, CAESES® will automatically create this folder in the current design directory.

6 Allrun File

We modify the "Allrun" file in this step. Note that we work on a copy of the original file.

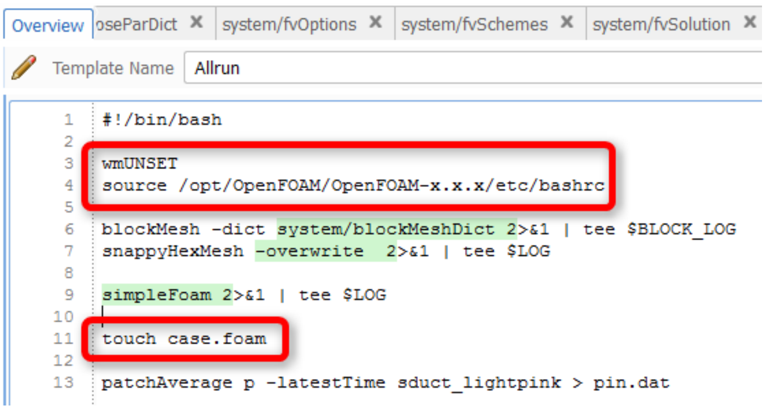
- Double-click on "Allrun" in the top right section "Input Files" of the software connector.
- If you don't want to start CAESES® from the terminal and you want to run OpenFOAM, we need to source the OpenFOAM environment variable. This could be done for example in the following way:

```
wmUNSET

source /opt/OpenFOAM/OpenFOAM-x.x.x/etc/bashrc
```

- For postprocessing reasons, we need to create the "case.foam" file by adding the following command into the Allrun script:

```
touch case.foam
```

- Switch back to the overview (click on "Overview" in the top left corner of the software connector).

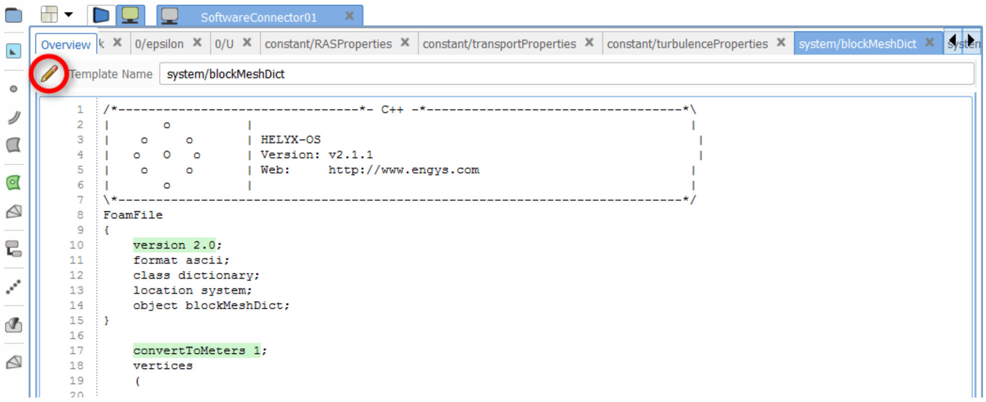


7

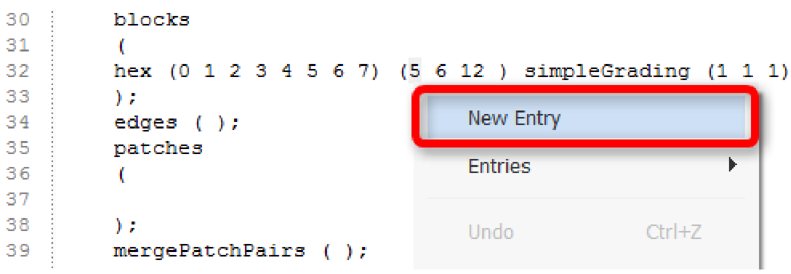
Introduce Parameters in Dict Files: Part 1

In this step, we introduce a parameters which controls OpenFOAM settings. First, it is useful to add a parameter which scales the mesh uniformly in all directions.

- Double-click on the “blockMeshDict” file.
- Switch to plain mode by clicking on the little pen on the top left of the file.



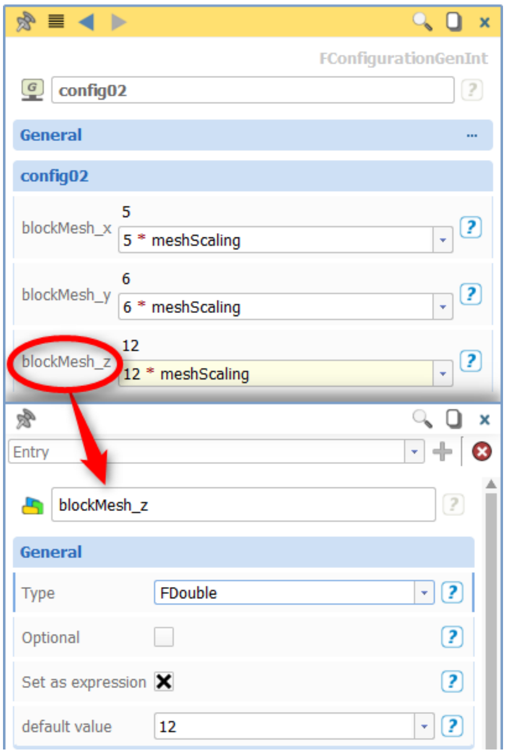
- Mark the number “5” which specifies the direction of cells in x-direction and choose *right click > select* “new entry”.



8 Introduce Parameters in Dict Files: Part 2

In this step, we introduce an additional parameter in order to be able to scale the block mesh settings.

```
blocks
(
hex (0 1 2 3 4 5 6 7) (<entry>blockMesh_x</entry> <entry>blockMesh_y</entry> <entry>blockMesh_z</entry> )
);
edges ( );
```



- In the object editor. Click on entry name “hex_” and rename these entries to blockMesh_x, blockMesh_y and blockMesh_z.
- Change the value to 5*1.
- Then select the value “1” in the editor, do *right click > create parameter*.
- Change the name to “meshScaling”.

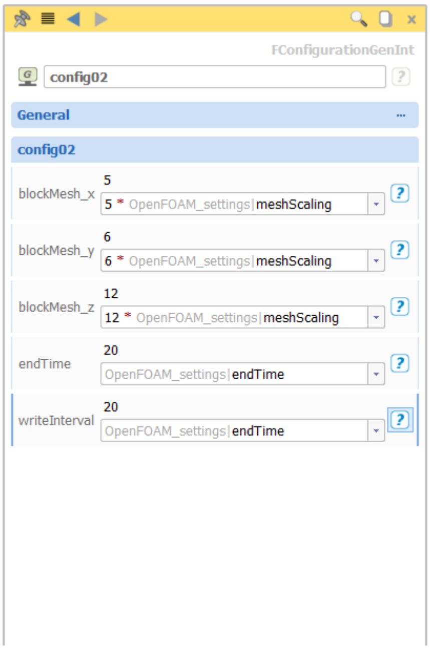
You can find the parameter in the *CAD* tab of the object tree.

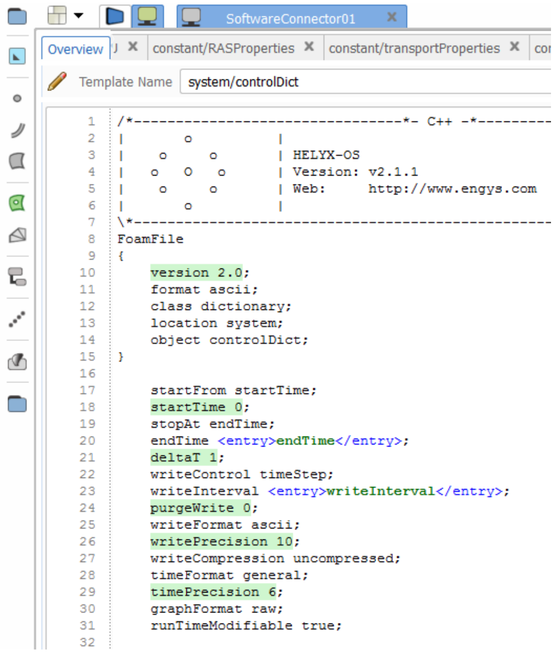
- Do the same for the other directions, while using the “meshScaling” design variable for all three directions.

9 Introduce Parameters in Dict Files: Part 3

We can also create a parameter which controls the number of iterations.

- In the software connector, select the "controlDict" file and create a new entry for the item "endTime" and "writeIntervall".
- Create a parameter for the "endTime" entry similar to what we did for the "meshScaling", and reduce the number to "20".
- Use the parameter "endTime" also for "writeIntervall".
- Put the parameter "endTime" and the design variable "meshScaling" into a scope called "OpenFOAM_settings".

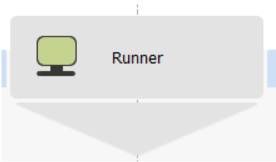




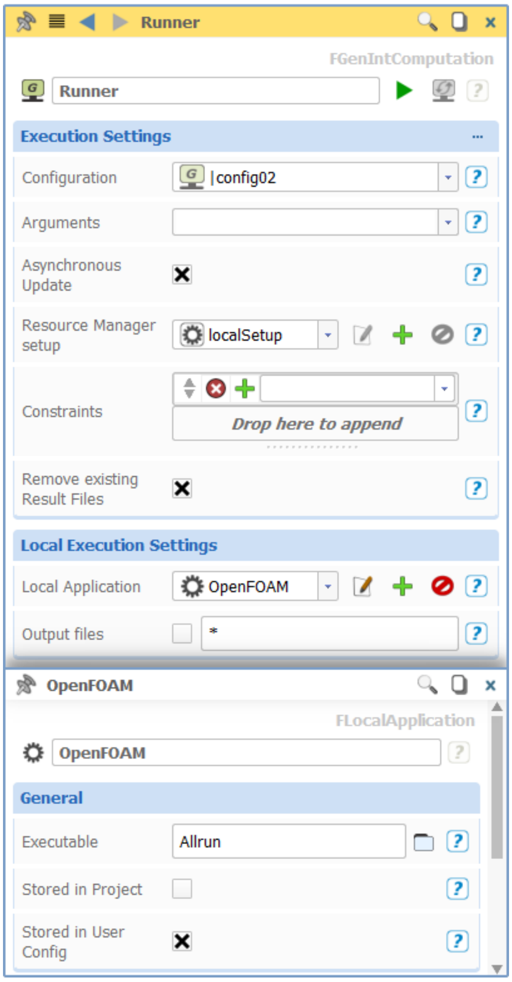

```
/*---------------------------------*- C++ -*----------
|        o              |
|    o     o            | HELYX-OS
|   o   O   o           | Version: v2.1.1
|    o     o            | Web:     http://www.engys.com
|        o              |
\*----------------------------------------------------
FoamFile
{
    version 2.0;
    format ascii;
    class dictionary;
    location system;
    object controlDict;
}

    startFrom startTime;
    startTime 0;
    stopAt endTime;
    endTime <entry>endTime</entry>;
    deltaT 1;
    writeControl timeStep;
    writeInterval <entry>writeInterval</entry>;
    purgeWrite 0;
    writeFormat ascii;
    writePrecision 10;
    writeCompression uncompressed;
    timeFormat general;
    timePrecision 6;
    graphFormat raw;
    runTimeModifiable true;
```

10 Software Connector: Computation

For postprocessing, we need to have results files and values, which will be inserted in the *Results Files* and *Results 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 want to use CAESES® to run OpenFOAM. Therefore we have to set up the *computation* "Runner":



- 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 makes sure that with each run the previous simulation files get deleted automatically, and is similar to the "Allclean" command.
- Create a new "Local Application" by clicking on the plus-button next to the attribute, and call it "OpenFOAM".
- As executable write "Allrun". This will trigger the "Allrun" script.
- Run the computation by clicking on the green run button (▶).
- While the computation is running, you can check the output in the *Task Monitor*.



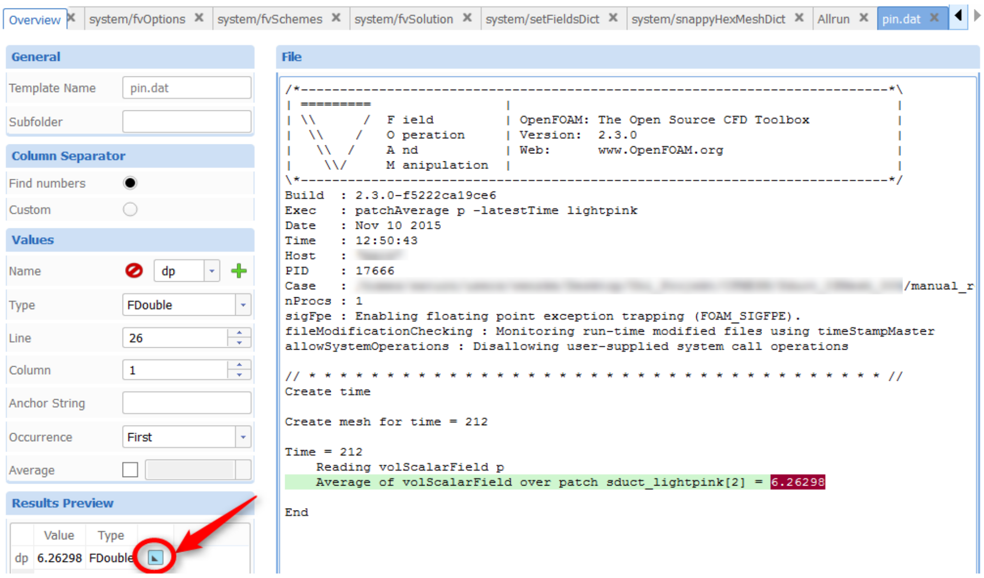
When the computation is done, check the results on your hard disc, to see how the results are handled by CAESES®: A new folder was created, with the name of the current project file (*.fdb). In this folder all results can be found.

11

Result Values

In order to assess the simulation, we need to have result files which provide for example 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 “pin.dat” file.
- Add the “pin.dat” file to the *Results Values* window of the software connector by using either drag & drop, or by using the plus button at the window.
- Double-click on “pin.dat” in the software connector to see the content.
- Add a value by clicking on the plus button and name it “dp”.
- Set *line* to “26” (in this case, this means: always the last row of the file).
- Set column to “1”.
- Now create a parameter for the value by clicking on the blue parameter symbol in the results preview table. This parameter will be the objective of the simulation at a later stage.



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

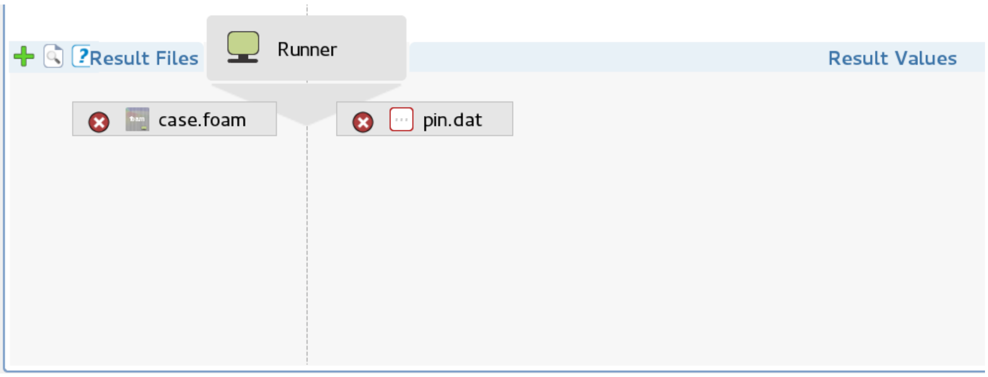
✓ Remember that this parameter is the result value of the last iteration or time step. If you have a strong oscillating result, you might average the values over a specific time.

12

Result Files

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

- Add the "case.foam" file from *manual_results/baseline/Runner* to this window, either by using drag & drop, or by using the plus button in the corner.

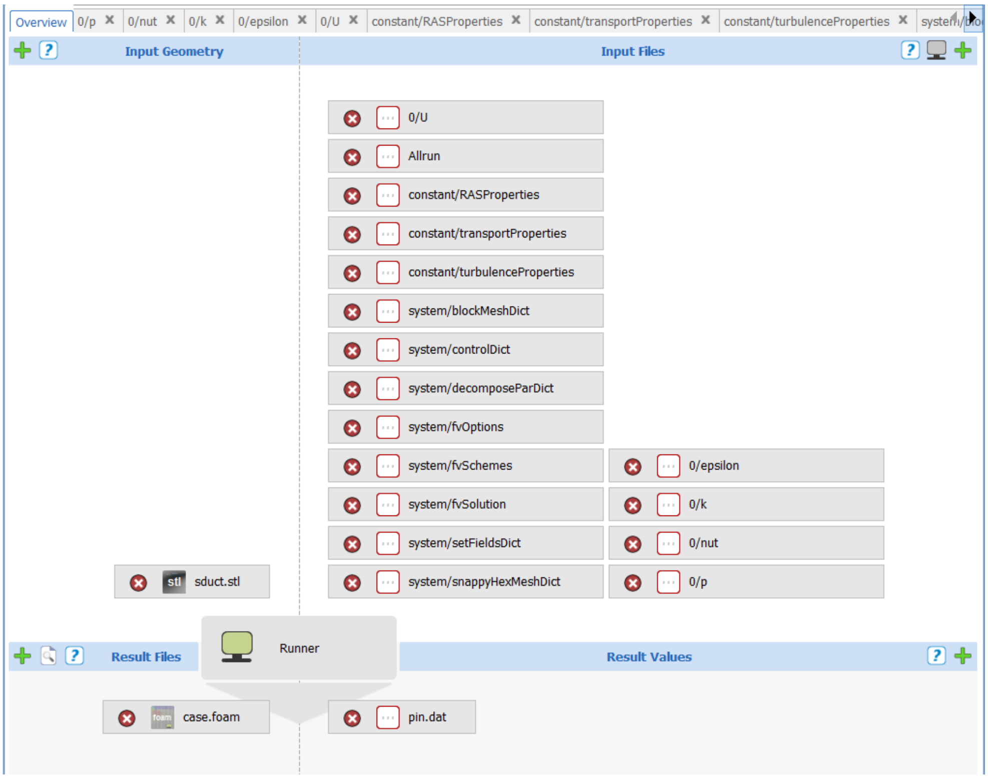


13

Running the Case

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

- Set the parameter "iterations" to "1000".
- You can also increase the "meshScaling" factor, which will increase the computational time.
- Go back to the software connector and run the simulation (again by using the ▶ button).

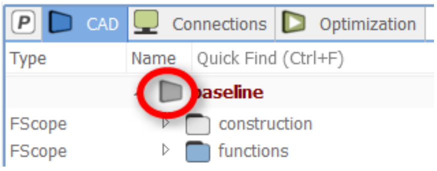


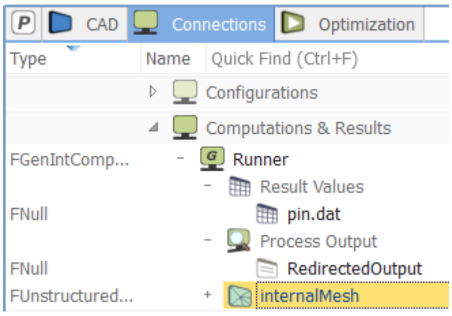
14

Postprocessing Part 1

When the computation has finished, we can do the postprocessing and visualize the results.

- Go to *CAD* and hide the baseline for the time being, by clicking on the baseline icon.
- Go to the *Connections* tab and open *Computations & Results > Runner*.





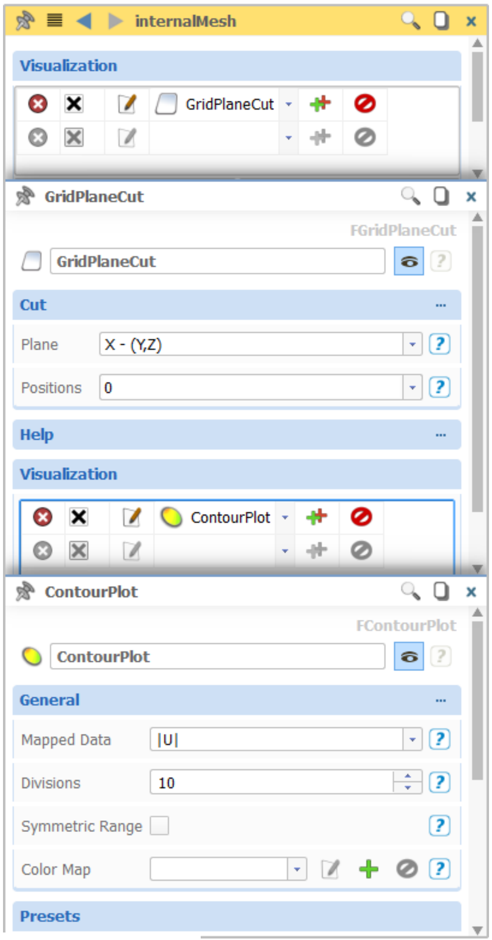
- Select the “internalMesh” object.
- Create a new *Visualization* (1).
- A new window appears, select “FGridPlaneCut” (2).
- Set the position to “0”.

✓ If you want to learn more about the postprocessing capabilities of CAESES® we recommend you to take a look at the postprocessing tutorial. (*Documentation Browser > Tutorials > Getting Started > Postprocessing)*

15

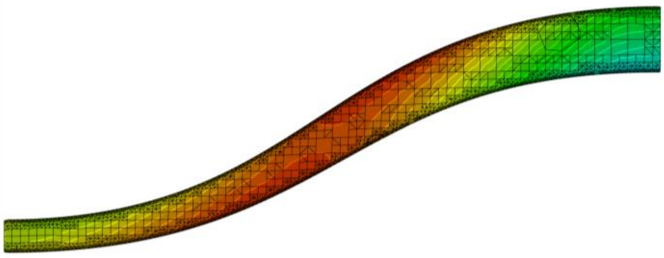
Postprocessing Part 2

Continue to set up the post processing.



- Create a new visualization at the bottom of this window.
- Select "FContourPlot".
- Choose the mapped data you want to visualize, for example "U". In the 3D window the visualization will appear.
- Increase the number of divisions to "20".

You could also create a vector visualization, streamlines, iso-surfaces or mesh visualizations.

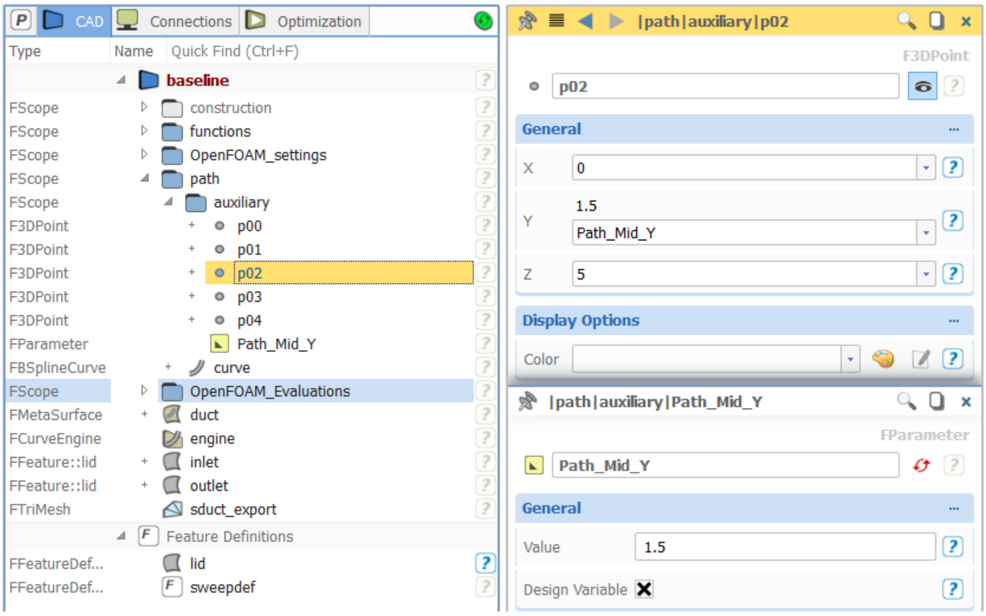


16

Quick Geometry Variation Part 1

Now we can do a quick geometry variation. Therefore we will create a design variable.

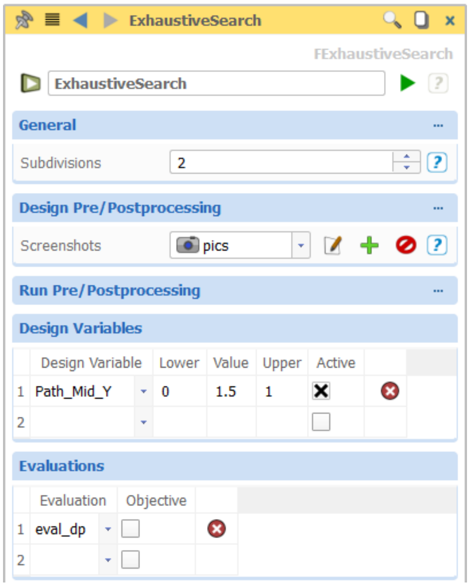
- Go to the path scope, select the auxiliary scope and create a design variable for the y-position of the point “p02”. This can be done by a right-click on the value.
- Change the name to “Path_Mid_Y”.



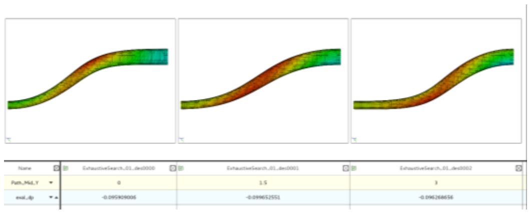
17 Quick Geometry Variation Part 2

In this step we will create and setup 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.



- 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" and the upper range to "3".
- Set "eval_dp" as an *evaluation.*
- Set up screenshot. Make sure you can see some scalar fields in your 3DView. Press the plus 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 *Exhaustive Search*. Note that this will trigger 3 parallel runs by default. If you want to change this setting, you have to edit the *Local Application* in the *Connections* tab.
- 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.