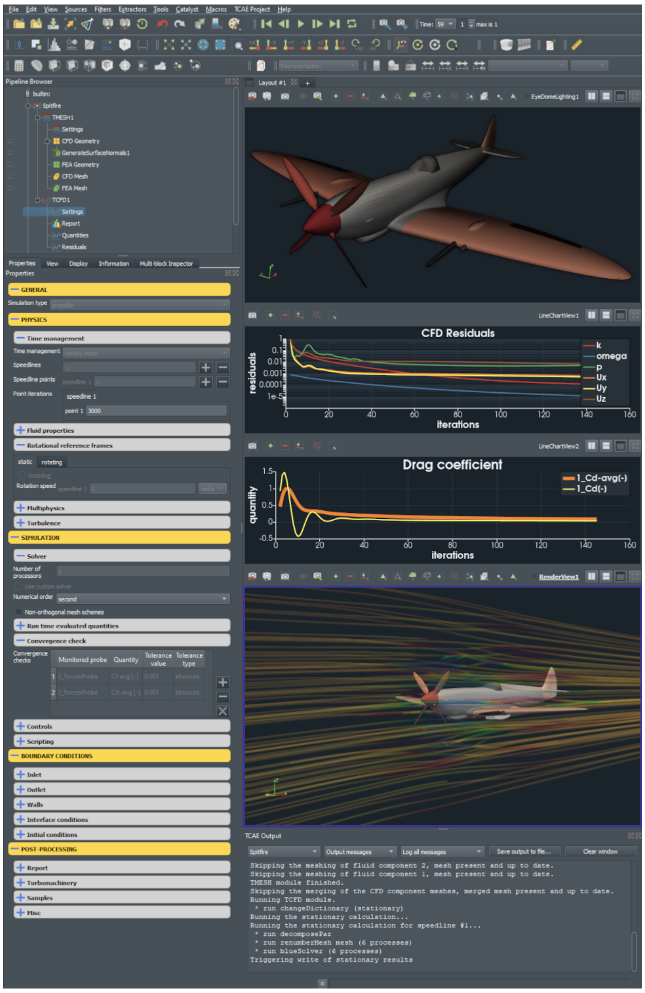
TCAE

Manual

Version 23.10



```
Skipping the meshing of fluid component 2, mesh present and up to date.
Skipping the meshing of fluid component 1, mesh present and up to date.
TMESH module finished.
Skipping the merging of the CFD component meshes, merged mesh present and up to date.
Running TCFD module.
 * run changeDictionary (stationary)
Running the stationary calculation...
Running the stationary calculation for speedline #1...
 * run decomposePar
 * run renumberMesh mesh (6 processes)
 * run blueSolver (6 processes)
Triggering write of stationary results
```

CFD SUPPORT S.R.O.
Sokolovská 270/201, 19000 Praha 9, Czech Republic
info@cfdsupport.com | phone: +420 212 243 883 | www.cfdsupport.com

Copyright

Trademarks

The OPENFOAM® related products and services are not approved or endorsed by OpenCFD Ltd. (ESI Group), producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD® trade marks.

Contents

Part I

TCAE Software

Chapter 1

TCAE Introduction

What is TCAE?

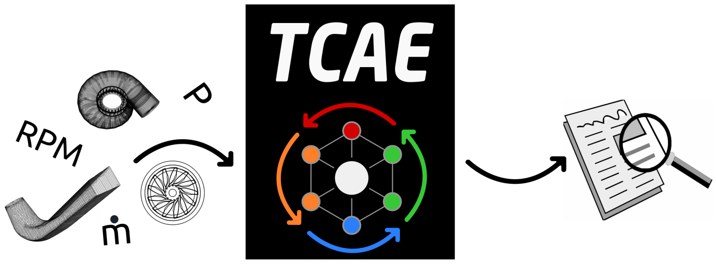
TCAE is a comprehensive software for engineering simulations produced by CFD SUPPORT LTD. TCAE is pretty unique because it enables a smart connection between various open-source codes for engineering simulations.

Both open-source and commercial nature

TCAE successfully merged the benefits of both open-source code and commercial code. Due to its open-source nature, there are no limitations on TCAE users, jobs, cores, and it is very flexible and further customizable. Due to its commercial nature, TCAE is professionally supported, well tested, ready for the industry, robust, accurate, automated, has GUI, has documentation and much more.

Unlimited simulation software

Because TCAE is unlimited, it scales the available resources to the fullest. TCAE is very effective at all engineering project stages. TCAE fits very well with large demanding transient simulations, as well as with many smaller simulations in the optimization loops.

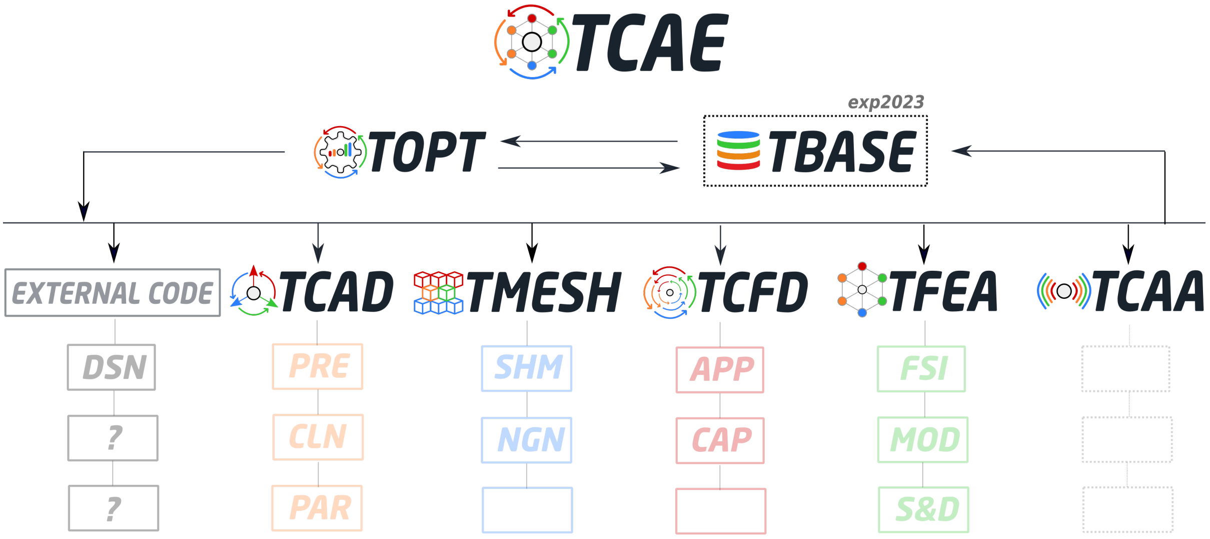


Automated software

TCAE is fully automated and very flexible. Its beauty is, that it is the user who decides how deep to dive into the theory or not at all. TCAE can be used as a black box (data in - data out) or as a highly sophisticated workflow where all the options remain open at the same time.

A system of independent modules

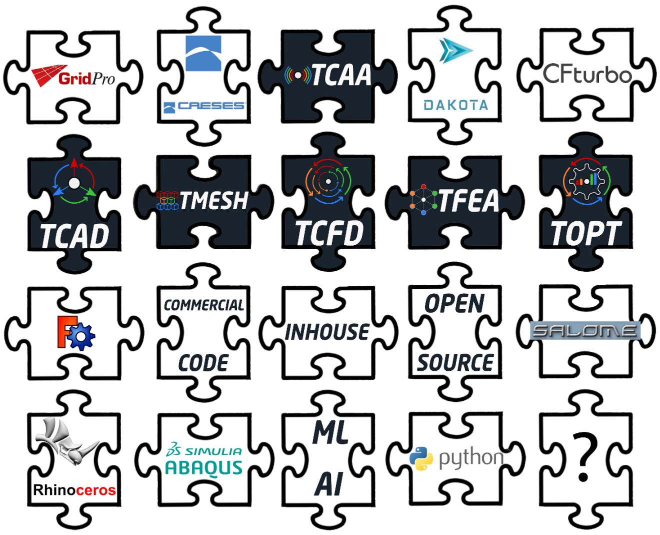
TCAE is structured as a system of independent modules (verticals) which can work together in an automated workflow or they can be used as standalone tools.



The whole TCAE workflow is managed by the master process CAEProcessor, which manages the communication among the individual modules. TCAE always keeps its modularity and is possible to be used inside any other existing workflow. TCAE is very flexible and it is continually developed. There are two main development lines. The horizontal one, which is adding the new tools, and new simulation plugins. And the Vertical development, which is adding new applications, new capabilities, and new hi-tech methods for the existing plugins.

Flexible workflow

TCAE is from the beginning developed to fit any existing CAE workflow. TCAE has a modular (plugin) character. Any part of it can be used within other external applications. It has strong interfaces to cover a wide range of input and output data. Both in GUI and batch mode are available.



TCAE applications

TCAE is completely engineering independent simulation software. TCAE can be used for a wide range of various CFD & FEA simulations. It was originally designed for simulations of rotating machinery like Pumps, Fans, Compressors, Turbines, etc., and it showed itself to be so effective, that later, it was extended with many other applications to cover even a wider range of applications. TCAE shows great performance at the external aerodynamics of various objects. TCAE can be used as a general-purpose code (contact CFD support for the particular questions), nonetheless, TCAE has proven excellent results at the following applications:

- Pumps
- Fans
- Hydro Turbines
- Wind Turbines
- Compressors
- Turbines
- Propellers
- Turbochargers
- Nozzles
- Diffusers
- Steam Turbines
- Ventilators
- Both axial and radial machines
- Both compressible and incompressible flows
- Water Valves
- Hydraulic Valves
- Ship hulls with propellers
- Stators
- Intake Manifolds
- Piping Systems
- Pressure Losses
- Turbulent Flows
- Exhaust Manifolds
- Wind load
- Building Construction
- External Aerodynamincs
- Car Aerodynamincs
- Human Comfort
- Aircraft Aerodynamincs
- Sports Aerodynamincs
- Atmospheric Flows

1.1 TCAE standard delivery includes:

- TCAE GUI software
- CAEProcessor - command line application
- OpenFOAM® based software + source code
- Turbo Blade Post - Visual Postprocessing software for turbomachinery

- All the software components are Perpetual, Unlimited Users, Jobs & Cores
- Real tutorials - preset test cases demonstrating how this workflow works on real examples
- Training - tailored training covering individual needs of the client
- Unlimited technical support
- Updates & Maintenance until technical support is valid

1.2 Technical Specifications & Software Features

- Natively compiled for Windows & Linux
- 100% workflow in GUI (graphical interface)
- Batch mode available
- Fully automated
- Compressible and incompressible fluid flow
- Turbulent flow
- Both subsonic and transonic
- Fully parallel computing
- Robust solvers
- Automated meshing available
- Meshes from 3rd party SW can be loaded
- Advanced turbo postprocessing
- Convergence monitoring
- Special boundary conditions
- Special function objects
- HTML reporting
- Python user defined functions
- Bash scripting

Chapter 2

TCAE – Installation & First run

This chapter summarizes the initial steps to run one of its preset tutorials. TCAE supports operating systems Windows and Linux. TCAE can be run both in graphical user interface (GUI), and also in batch mode in terminal.

2.1 Windows

2.1.1 Software Requirements

Operating system requirements

is compatible with the following Windows versions:

- Windows 11
- Windows 10
- Windows 8.1
- Windows 8
- Windows 7
- Windows Server 2012 R2

Operating system language

TCAE requires "." (dot) as the decimal separator. If you are using a specific non-English version of Windows, please check it in "Regional and Language Options" of your system.

Additionally, TCAE supports a standard ASCII encoding in path names and in all other strings which are used by TCAE, i.e., avoid special characters such as á, ß, ö, è, î, ç, š, % # $ % ˆ & etc. ...

Case sensitivity

As a part of its solver infrastructure, TCAE installs the OpenFOAM software. Due to its design, OpenFOAM requires case-sensitive filesystem. When installing TCAE, your operating system will be automatically checked for case sensitivity and, if needed, the necessary modifications to its setup will be done by the installer at the very beginning of the installation process. On Windows 10 18.04 and newer this amounts to enabling WSL ("Windows Subsystem for Linux"), while on older systems this means disabling the default system-wide case insensitivity ("obcaseinsensitive = 0"). This is performed automatically by the installer.

2.1.2 Installation

1. Start with the TCAE installation package. It is delivered with the TCAE purchase, or the TCAE trial version can be requested here:
   `http://www.cfdsupport.com/tcae-demo.html`

2. Download and install TCAE. Just follow the install wizard and preferably install all the components of TCAE.

3. Copy the license file (*License.dat*) into the TCAE installation directory, typically: `C:\TCAE\23.03\`

4. Ready to run now!

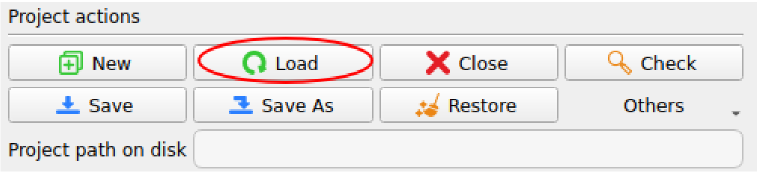
5. A ready-to-run tutorials can be found in:
   `C:\TCAE\23.03\tutorials\`
   Other tutorials can be downloaded from CFDSUPPORT website:
   `https://www.cfdsupport.com/download-cases.html`

When using GUI:

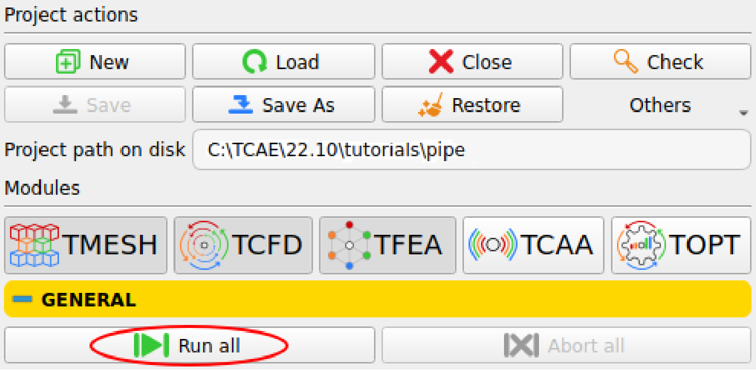
6. Launch ParaView using the `"TCAE 23.03"` desktop shortcut or the corresponding item in the Start menu.

7. In case you see some errors immediately after launching ParaView, which can be caused by a missing or incompatible OpenGL drivers on your system; go to the TCAE installation directory, i.e., `C:\TCAE\23.03\etc` and edit the file `paraview-launcher.ini` by changing "`system`" keyword to "`software`". Launch ParaView again.

8. Open the tutorial project folder e.g. *pipe* using the *Load* button in the *General* section of the *Properties* panel located in the left area of the ParaView window.



9. Click *Run all*.



Project path on disk C:\TCAE\22.10\tutorials\pipe

10. And all the processes are done automatically: the new case is written in project folder into default case name `simulationRun1`, mesh is created, case is set up, case is simulated, results are evaluated and report is written down.

11. Inside the *Line Chart View*, the instant residuals can be watched. For more information about the computation, the report can be updated anytime during the simulation.

12. When the simulation is finished the final results report is written down. It is located inside the project folder:
`.\pipe\simulationRun1\TCFD\report-steadystate-TCFD-efficiency1`

13. Visual postprocessing can be done in ParaView using Turbo Blade Post plugins (included).

Important Technical Notice

The default TCAE installation in Windows (only in Windows) can automatically recognize rendering capabilities to cover both the direct usability of computers and to work on clusters, clouds, remote desktops, etc. The rendering options are set in the file `C:\TCAE\23.03\etc\paraview-launcher.ini` .

There are three possible values of the **opengl** parameter:

- **opengl=auto** - default value (automatically choose between software and system)
- **opengl=system** - uses system OpenGL drivers (usually not available for remote control, etc.)
- **opengl=software** - uses open-source implementation of The OpenGL specification (Mesa, works everywhere, lower rendering performance)

If any problems appear during the start of TCAE, please switch back to the **auto** or **software** mode.

When using Cygwin terminal:

6. Navigate to the TCAE installation folder, i.e., `C:\TCAE\23.03`

7. Run OpenFOAM® for Windows command line by clicking on `TCAE 23.03 OF4Win Shell.lnk`.

8. Navigate to the `tutorials` directory (`mc` command – Midnight Commander – is recommended, or navigate manually in terminal:
   `cd ..\..\tutorials\`
   and run all by one command:
   `CAEProcessor --project pipe --allrun`

9. All the process is done automatically: new case is written in project folder into default case name `simulationRun1`, mesh is created, case is set up, case is simulated, results are evaluated and report is written down. When the simulation is finished the final results report is written down immediately. It is located inside the test case:
   `./pipe/simulationRun1/TCFD/report-steadystate-TCFD-efficiency1`

10. Visual postprocessing can be done in ParaView using Turbo Blade Post plugins (included).

When using Command Prompt (cmd) terminal:

6. Run Command Prompt (cmd).

7. Navigate to the `tutorials` directory:
   `C:\TCAE\23.03\tutorials\`
   and run all by one command:
   `C:\TCAE\23.03\bin\CAEProcessor --project pipe --allrun`
   NOTE: The full path to the `CAEProcessor` executable has to be provided. User can enable a direct access to `CAEProcessor` by adding the path `C:\TCAE\23.03\bin` to "Environment Variables".

8. All the process is done automatically: new case is written in project folder into default case name `simulationRun1`, mesh is created, case is set up, case is simulated, results are evaluated and report is written down. When the simulation is finished the final results report is written down immediately. It is located inside the project folder:
   `./pipe/simulationRun1/TCFD/report-steadystate-TCFD-efficiency1`

9. Visual postprocessing can be done in ParaView using Turbo Blade Post plugins (included).

Multiple TCAE versions

Several TCAE versions can live together. All the TCAE files are always associated with the latest TCAE installation. To have an option of using older TCAE versions go to file `C:\ProgramFiles\CFDsupport` and delete paths after equality sign in the second and the third line. New `paraview-launcher.ini` file should look like this:

```
[associations]
foam=
tcae=
```

Installation folder in Windows

When TCAE is about to be run from an external application it may be useful to read automatically the TCAE installation directory from Windows registry with following command:

```
>> REG QUERY HKLM\SOFTWARE\WOW6432Node\Microsoft\Windows\CurrentVersion\Uninstall\TCAE-23.03 /
    v InstallationPath \&
```

2.2 Linux

2.2.1 Software Requirements

Operating system requirements

TCAE is compatible with the Linux distribution having GLIBC library version 2.13 and higher. Specially, the software was tested in:

- openSUSE Leap 15.0 (Linux 4.12.14, GLIBC 2.26)
- CentOS 7.5 (Linux 3.10.0, GLIBC 2.17)
- Ubuntu 20.04.4 LTS (Linux 5.4.0, GLIBC 2.27)
- Ubuntu 18.04.4 LTS (Linux 5.4.0, GLIBC 2.27)
- Ubuntu 16.04.4 LTS (Linux 4.4.0, GLIBC 2.23)
- Ubuntu 14.04.5 LTS (Linux 3.13.0, GLIBC 2.19)

Operating system language

TCAE requires "." (dot) as the decimal separator. If you are using a specific non-English version of your Linux distribution, please check the "LC_NUMERIC" setup using the shell command "locale".

Additionally, TCAE supports a standard ASCII encoding in path names and in all other strings which are used by TCAE, i.e., avoid special characters such as á, ß, ö, è, î, ç, š, ...

2.2.2 Installation

1. Request the trial version (this step is only needed for TCAE trial version - in case you already purchased the TCAE perpetual version - please skip this step):
   `http://www.cfdsupport.com/tcae-demo.html`

2. Download TCAE *TCAE-23.03*.sh*

3. In terminal install it into your favorite directory (e.g. `/home/user/`):
   `bash TCAE-23.03*.sh -install`

4. Copy the license file (*License.dat*) into the TCAE installation directory, typically: `/home/user/TCAE-`

5. Ready to run now!

6. In a new terminal source the system variables (with every new terminal, or add this to your `$HOME/.bashrc` for permanent usage):
   `source /home/user/TCAE-23.03*/OpenFOAM-dev/etc/`
   `bashrc-release`

7. A ready-to-run tutorial can be found in:
   `/home/user/TCAE-23.03*/TCAE/tutorials/`
   `TCAETutorials/TCFD+TFEA/pipe`
   Other tutorials can be downloaded from:
   `https://www.cfdsupport.com/download-cases.html`

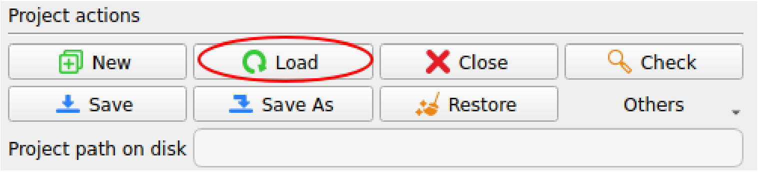
When using GUI:

8. Launch ParaView by typing the command `TCAE` in the terminal.

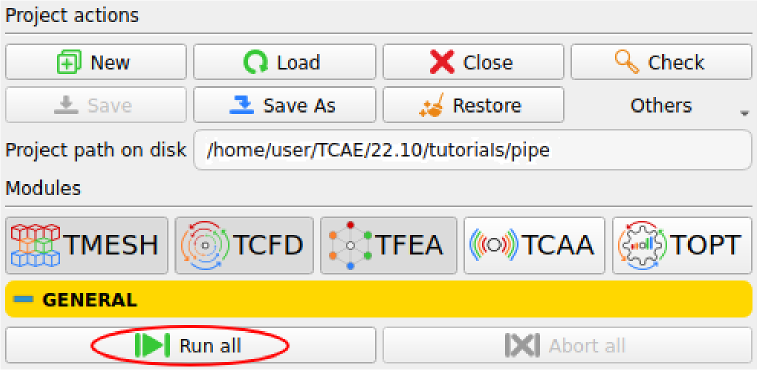
Important Technical Notice

In case you see some OpenGL errors after launching ParaView use the command `TCAE-lp`. This usually happens on remote-controled computers and clusters for which OpenGL drivers are missing.

9. Open the project folder e.g. *pipe* in via the *Load* button in the *General* section of the *Properties* panel located in the left area of the ParaView window.



10. Click *Run All.*



Project path on disk /home/user/TCAE/22.10/tutorials/pipe

11. And all the process is done automatically: new case is written in project folder into default case name `simulationRun1`, mesh is created, case is set up, case is simulated, results are evaluated and report is written down.

12. When the simulation is finished the final results report is written down immediately. It is located inside the project folder:
`./pipe/simulationRun1/TCFD/report-steadystate-TCFD-efficiency1`

13. Inside the *Line Chart View*, the instant residuals can be watched. For more information about the computation, the report can be updated anytime during the simulation.

14. Visual postprocessing can be done in ParaView using Turbo Blade Post plugins (included).

When using terminal:

8. In terminal navigate to the `tutorials` directory (`mc` – Midnight Commander – is recommended, or navigate manually in terminal) and run all by one command:
`CAEProcessor --project pipe --allrun &`

9. And all the process is done automatically: new case is written in project folder into default case name `simulationRun1`, mesh is created, case is set up, case is simulated, results are evaluated and report is written down. When the simulation is finished the final results report is written down immediately. It is located inside the project folder:
`./pipe/simulationRun1/TCFD/report-steadystate-TCFD-efficiency1`

10. Visual postprocessing can be done in ParaView using Turbo Blade Post plugins (included).

Chapter 3

TCAE – Architecture

3.1 About TCAE

TCAE has a modular architecture, and consists of several modules, which are connected through the "control centre"

TCAEManager

Each module is focused on the different field of the CAE workflow. Currently, there are five modules available:

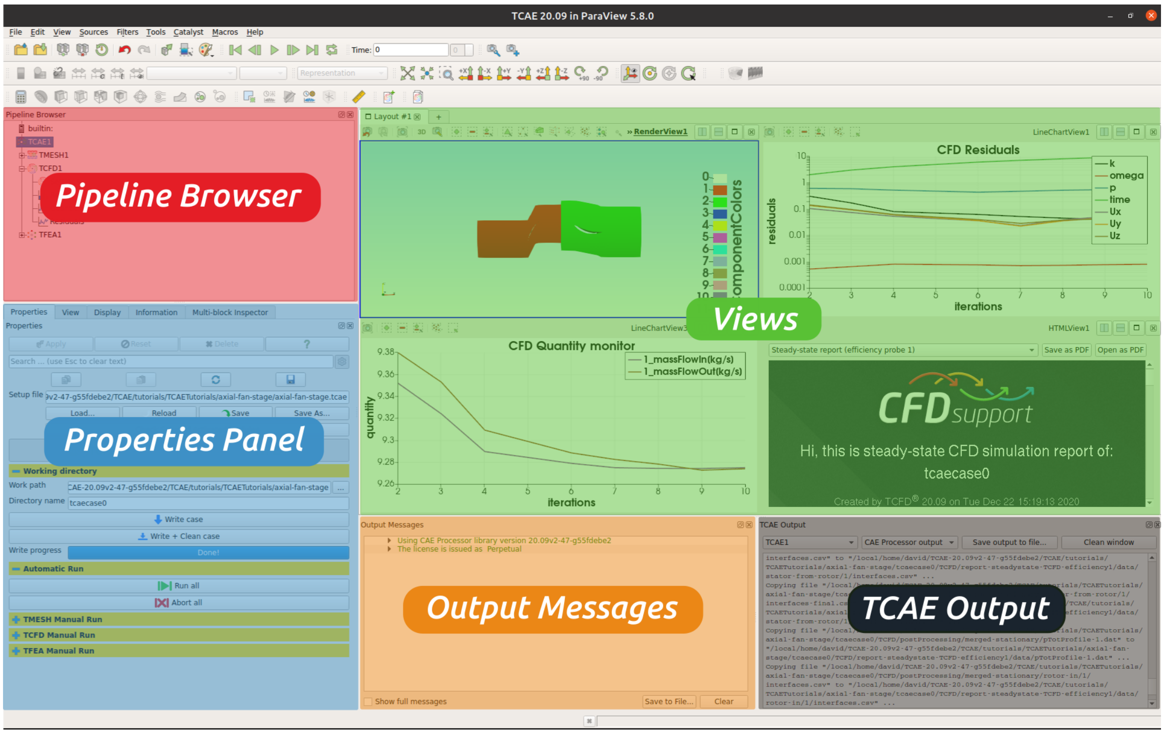
- TMESH module for creation of CFD and FEA meshes
- TCFD module for setting up the CFD calculation
- TFEA module for setting up the FEA calculation
- TCAA module for setting up the CAA calculation
- TOPT module for setting up the optimization

The responsibilities are *separated*: each module serves only as an *interface* for setting the calculation parameters, but actual *running* of the simulation is *managed* in the TCAEManager.

TCAE also has its own command-line backend, which is called CAEProcessor, and it can be controlled directly using the terminal, without GUI (see section 5).

3.2 TCAE – GUI Layout

TCAE's graphical user interface is created in the ParaView software (3.8). The original ParaView GUI is modified for our specific needs, but all of its functionality has been preserved. The GUI layout of the TCAE is shown in Figure 3.1.



Hi, this is steady-state CFD simulation report of: tcaecase0

Created by TCFD® 20.09 on Tue Dec 22 15:19:13 2020

```
▸ Using CAE Processor library version 20.09v2-47-g55fdebe2
▸ The license is issued as  Perpetual
```

```
interfaces.csv" to "/local/home/david/TCAE-20.09v2-47-g55fdebe2/TCAE/tutorials/
TCAETutorials/axial-fan-stage/tcaecase0/TCFD/report-steadystate-TCFD-efficiency1/data/
stator-from-rotor/1/interfaces.csv" ...
Copying file "/local/home/david/TCAE-20.09v2-47-g55fdebe2/TCAE/tutorials/TCAETutorials/
axial-fan-stage/tca                                          r-from-rotor/1/
interfaces-final.cs                                          /TCAE/tutorials/
TCAETutorials/axial                                          efficiency1/data/
stator-from-rotor/1
Copying file "/local/                                        ials/TCAETutorials/
axial-fan-stage/tcaecase0/TCFD/postProcessing/merged-stationary/pTotProfile-1.dat" to
"/local/home/david/TCAE-20.09v2-47-g55fdebe2/TCAE/tutorials/TCAETutorials/axial-fan-
stage/tcaecase0/TCFD/report-steadystate-TCFD-efficiency1/data/pTotProfile-1.dat" ...
Copying file "/local/home/david/TCAE-20.09v2-47-g55fdebe2/TCAE/tutorials/TCAETutorials/
axial-fan-stage/tcaecase0/TCFD/postProcessing/merged-stationary/rotor-in/1/
interfaces.csv" to "/local/home/david/TCAE-20.09v2-47-g55fdebe2/TCAE/tutorials/
TCAETutorials/axial-fan-stage/tcaecase0/TCFD/report-steadystate-TCFD-efficiency1/data/
rotor-in/1/interfaces.csv" ...
```

Figure 3.1: TCAE – GUI layout.

The screen is divided into several sections. Apart from the top menu and toolbar, which aren't mostly needed for the running of the simulation, following sections are present:

- Pipeline Browser serves as the main navigation tool. It contains the *Pipeline* – tree-like menu, which is used for switching between TCAE modules, and between their outputs. There is top-level item TCAEManager. This item is renamed after the project folder name. Another items connected to the TCAEManager are the modules TMESH, TCFD (if enabled), TFEA (if enabled), TOPT (if enabled) and TCAA (if enabled). Each module has then its *Output Ports*, that are used to display the output of the modules in the *Views*, as will be described below. The *Pipeline Browser* is depicted in Figure 3.2.

- Properties Panel shows the setting of the selected module, so when user selects some module in the *Pipeline Browser*, its input settings can be adjusted.

- Output Messages window shows mostly messages containing errors in settings or warnings (see section 4.1) and errors that occur during the simulation. Besides that it serves as the main interface for displaying other notifications produced by ParaView.

- TCAE Output window displays the running text, which source is selectable between these options:
  - *CAE Processor output* is produced by the TCAE, and contains the information of such a type as what application is currently running, and what steps of the simulation are underway.

  - *Solver output* shows the raw text output of the application, which is currently running (mesher, CFD solver, FEA solver...).

- Views visualize inputs (like geometry) and the outputs (mesh, residuals etc.) of the simulation. Working with views is straightforward, user can open and close views, divide some view into two smaller, resize them, add new view layout etc. There are different views dependent on the type of the data being displayed. Displaying the data works closely with the *Pipepline Browser*. Each *Output Port* (e. g. *CFD Geometry*) has its own button 👁, which controls visibility of the underlying data in the active *View* (Figure 3.1). So it is possible to have more *Views*, even of the same type, and to show the data just in some of them, as desired. These views are utilized in TCAE:
  - *RenderView* shows the geometry and meshes - the outputs of the *Output Ports* of the TMESH: CFD Geometry, CFD Mesh, FEA Geometry and FEA Mesh
  - *LineChartView* shows the charts with representation of live monitored quantities and residuals - outputs of the ports of the TCFD: Quantities and Residuals
  - *HTMLView* is needed for the representation of HTML report of the calculation, that is the output of the Report ports of the TCFD, TFEA and TCAA modules

For better understanding of the ParaView software, please have a look at the ParaView manual.

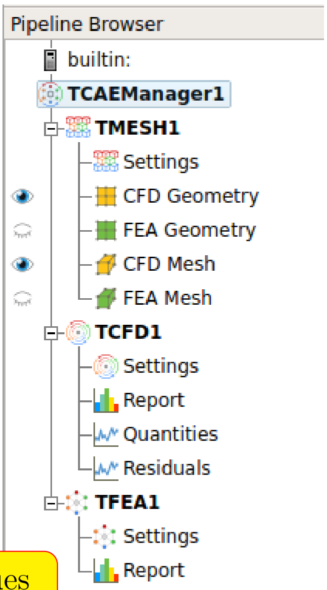
3.3 TCAE – Configuration File

TCAE project saves and loads its settings on/from disk in the form of setup file, which is regular, human readable, text file. It contains all setup needed for running the simulation.

For common usage, one doesn't need to interact with the file, just load/save project folder in TCAE graphical interface. For the advanced users, one might want to edit this file manually and run TCAE from the command line (see section 5). This practice is highly unrecommended for regular users.

Watch out!

It is unrecommended to manually modify keywords and their values in the setup.tcae file, unless you are a highly experienced user.



ure 3.2: TCAE – GUI layout.

The setup file allows writing comments, which have to be introduced by a hash sign ('#').

The settings are written in the form of keyword-value pairs. The keywords are strings uninterrupted by a whitespace character. They contain only alphanumerical characters, numbers, dashes and underscores. Every keyword can be generally set to an arbitrary number of values. A value is a string uninterrupted by a whitespace character. Some keywords have a special structure *N_something*, where *N* is a positive number. These are used to assign properties to individual components, speedlines, working points, post-processing modes etc. This nest to deeper levels, resulting in keywords like $X\_keywd1 - Y\_keywd2 - Z\_keywd3$.

Each module has its own distinct keywords, that correspond to its settings. For complete overview of keywords see corresponding sections for the modules (for TMESH section 8, for TCFD section 12, for TFEA section 19).

Only one keyword, that doesn't belong to any module, is keyword *version*. Corresponding value represents the version of the TCAE software, and is necessary for backward-compatibility provision, so one is able to load a `*.tcae` file (which were used in older version of TCAE) in the newer version of TCAE.

Both the UNIX-like (LF) and Windows (CR+LF) line endings are supported on both UNIX-like and Windows operating systems.

Some numerical values in the TCAE file can bear a user-selectable physical unit. This is always indicated in the keyword table as "Selectable unit". Table 3.1 summarizes the available units. The unit is appended to the numerical value using an underscore. So, for instance, one millimeter is denoted as `1_mm` or `0.1_cm` or `0.001_m`. While the temperature quantities allow the use of the degree symbol, e.g. `100_°C`, it is recognized in UTF-8 encoding only (degree symbol is not part of ASCII). This is natural in modern Unix-like operating systems, but we recommend sticking to `1_C` in Windows. When a number is used without unit suffix, it is assumed that it is in SI (i.e. the metric system).

Quantity	Symbol	Known units	Suffixes
time	t	seconds (SI)	`_s`, `_sec`, `_seconds`
		minutes	`_m`, `_min`, `_minutes`
		hours	`_h`, `_hrs`, `_hours`
		revolutions	`_revolutions`
		degrees	`_degrees`
length	l	metres (SI)	`_m`
		decimetres	`_dm`
		centimetres	`_cm`
		millimetres	`_mm`
		inches	`_in`
		feet	`_ft`
velocity	u	metre per second	`_m/s`
		metre per minute	`_m/min`
		metre per hour	`_m/h`
		kilometer per hour	`_km/h`
temperature	T	Kelvin (SI)	`_K`, `_Kelvin`
		degrees of Celsius	`_°C`, `_C`, `_Celsius`
		degrees of Fahrenheit	`_°F`, `_F`, `_Fahrenheit`
pressure / stress	p	Pascal (SI)	`_Pa`, `_Pascal`
		MegaPascal (SI)	`_MPa`, `_MegaPascal`
		GigaPascal (SI)	`_GPa`, `_GigaPascal`
		bar	`_bar`
		millibar	`_mbar`, `_millibar`

Quantity	Symbol	Known units	Suffixes
		atm	`_atm`
		Torr	`_torr`, `_Torr`, `_mmHg`
		pounds per square inch	`_psi`
density	ρ	kg per cubic metre (SI)	`_kg/m^3`
		g per cubic centimetre	`_g/cm^3`
heat capacity	c	J per kg per Kelvin (SI)	`_J/(kg.K)`
		Btu per pound per Fahrenheit	`_Btu/(lb.F)`
heat conductivity		Watts per metre per Kelvin (SI)	`_W/m.K`
		Watts per centimetre per Kelvin	`_W/cm.K`
		Btu per hour per feet per Fahrenheit	`_Btu/h.ft.F`
thermal expansion coefficient		per Kelvin	`_K^-1`
angular velocity	ω	radian per second (SI)	`_rad/s`, `_s^-1`
		revolutions per minute	`_RPM`
mass flow rate	$\dot{m}$,ϕ	kg per second (SI)	`_kg/s`
		kg per minute	`_kg/min`
		kg per hour	`_kg/h`
		g per second	`_g/s`
		g per minute	`_g/min`
		g per hour	`_g/h`
volumetric flow rate	ϕ	m^3 per second (SI)	`_m^3/s`
		m^3 per minute	`_m^3/min`
		m^3 per hour	`_m^3/h`
		litres per second	`_l/s`
		litres per minute	`_l/min`
		litres per hour	`_l/h`
		US gallon per second	`_USgal/s`
		US gallon per minute	`_USgal/min`
		US gallon per hour	`_USgal/h`
dynamic viscosity	μ	poiseouille (SI)	`_Pa.s`, `_Pl`
		poise	`_P`
		centipoise	`_cP`

Table 3.1: Physical units recognized by TCAE.

3.4 TCAE – Backward Compatibility

TCAE is being developed with backward compatibility with older versions of the software in mind. It is possible to load *.tcae file saved in an older version of TCAE and also old *.tcfd file from product TCFD 19.10 (predecessor of TCAE).

3.5 TCAE – Typical Workflow

The brief common usage of the TCAE software is given below and its graph is displayed in Figure 3.3. The detailed description of each of the steps is the content of the following chapters.

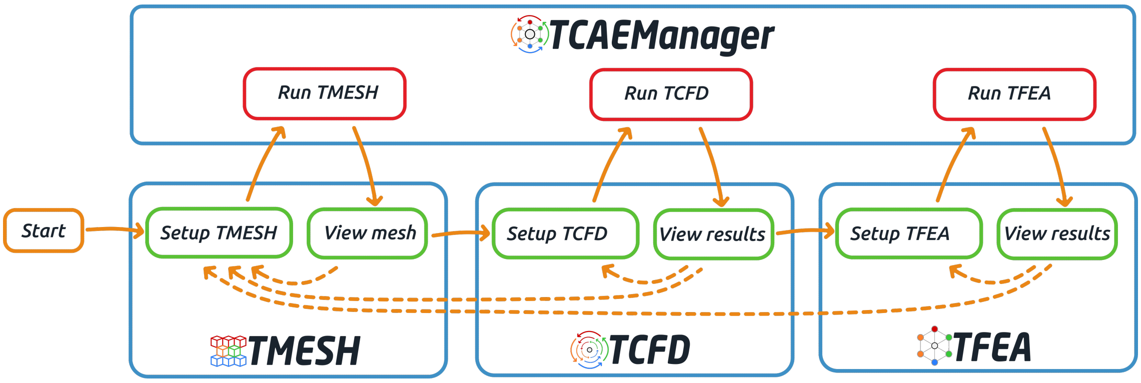


Figure 3.3: TCAE – user workflow

- Launch TCAE, inside the TCAEManager load project folder with some previously saved setup, or just create a new project by saving the default settings.
- Switch to the TMESH module, load geometry (or some existing external mesh), and adjust the parameters of the meshing phase.
- Switch back to the TCAEManager, click on Save, that saves the project settings on disk.
- Now run the meshing phase.
- When the mesh is created, you can see it and check whether the mesh is suitable for the simulation.
- Switch to the TCFD module. Set up the CFD simulation parameters, e.g. physics and boundary conditions.
- Go back to the TCAEManager, click on Save and run the CFD calculation.
- When finished, it is possible to generate report of the calculation, and show the complete results.

This is not the only way, how to proceed, one can for example set up all of the modules in advance, and then simply run the whole simulation in one step (mesh, CFD, FEA). This "linear" workflow is depicted in Figure 3.4.

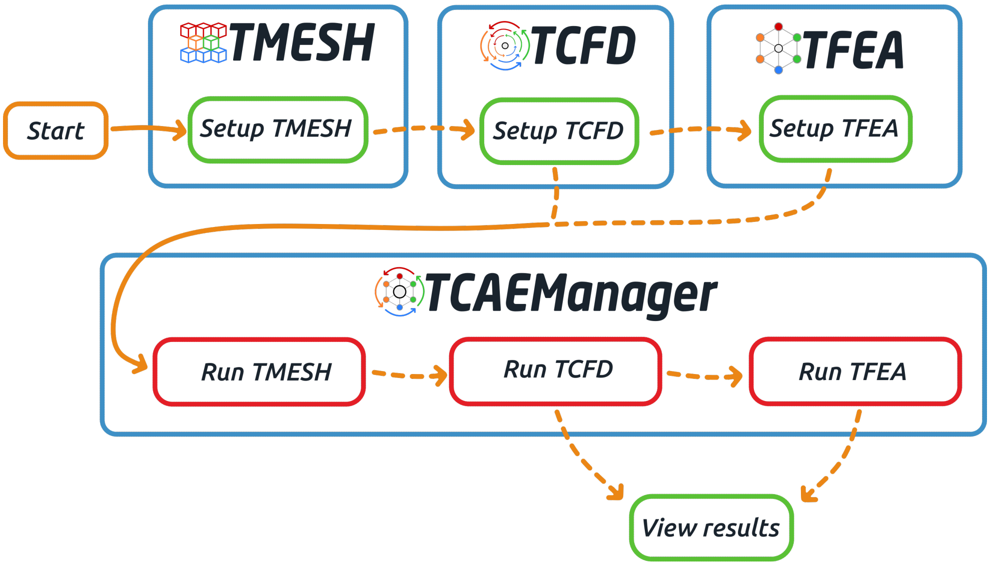


Figure 3.4: TCAE – "linear" user workflow

3.6 TCAE – Case Structure

TCAE manages the computational case inside the project folder. Its directory structure corresponds to the structure of the modules and is depicted in the table 3.2 in its basic form.

The whole directory with the case should be read-only (obvious exception is when user needs to delete the whole case), and its basic structure is shown here (as well as in the sections corresponding to the individual modules), because user might want to extract some useful data for postprocessing or to look at the logs.

3.7 Turbo Blade Post

Turbo Blade Post is a product of company CFD SUPPORT LTD. It was especially created to enable an effective graphical postprocessing of rotating machinery - both radial and axial machines such as pumps, hydro turbines, compressors, turbochargers, propellers and many more.
Turbo Blade Post is a set of following plugins (filters) for ParaView:

- Turbo Unwrap - for visualization of the blade-to-blade view or 2D plots around the blades.
- Meridional Average - for evaluation and visualization of the meridional averages in the blade passage.

More information about Turbo Blade Post can be found in Chapter 16.

3.8 ParaView

TCAE graphical user interface makes use of the ParaView software, which is an open source multiple-platform application for interactive, scientific visualization. It has a client–server ar-

🗀 project_folder	Project directory (name chosen by user)
└ 🗀 simulationRun	
└ 🗀 TMESH	
└ . . .	All files, that belong to TMESH module
└ 🗀 TCFD	
└ . . .	All files, that belong to TCFD module
└ 🗀 TFEA	
└ . . .	All files, that belong to TFEA module
└ 🗋 statuses	List of modules, that are used for this case
└ 🗀 logs	
└ . . .	Text files with output log produced by the TCAE for each individual computational case. These logs contain all of the text, which is displayed during the computation in the *TCAE Output* window (see 3.2)
└ 🗀 commonData	
└ 🗀 TMESH	
└ 🗀 imported	
└ . . .	
└ 🗋 setup	Setup file, that was used to initialize this case
└ 🗋 info	
└ 🗋 pointIDIndices	

Table 3.2: TCAE – case structure

chitecture to facilitate remote visualization of datasets, and generates level of detail (LOD) models to maintain interactive frame rates for large datasets. It is an application built on top of the Visualization Tool Kit (VTK) libraries. Where VTK is a set of libraries that provide visualization services for data, task, and pipeline parallelism, ParaView is an application designed for data parallelism on shared-memory or distributed-memory multicomputers and clusters. It can also be run as a single-computer application.

ParaView offers the possibility to extend its functionality in several directions. This encompasses modifications to the GUI and implementation of new plugins.

3.8.1 ParaView License

ParaView uses a permissive BSD license that enables the broadest possible audience, including commercial organizations, to use the software, royalty free, for most purposes. In addition, there are other licenses that are applicable because of other packages leveraged by ParaView or developed by collaborators. Lastly, there are specific packages for the ParaView binaries available on `https://www.paraview.org` that have applicable licenses.

Copyright (c) 2005-2017 National Technology & Engineering Solutions of Sandia, LLC (NTESS), Kitware Inc. Sandia National Laboratories, New Mexico PO Box 5800 Albuquerque, NM 87185 Kitware Inc., 28 Corporate Drive, Clifton Park, NY 12065, USA

Under the terms of Contract DE-NA0003525 with NTESS, the U.S. Government retains certain rights in this software: Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. Neither the name of Kitware nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

Chapter 4

TCAE – Running & PostProcessing

4.1 TCAEManager

The main structure for setting, running and postprocessing TCAE simulation includes the TCAEManager. It offers a simple user interface for all these tasks in the framework of TCAE. TCAEManager is selected after launching TCAE and user can switch to it by selecting TCAEManager1 in the *Pipeline Browser*.

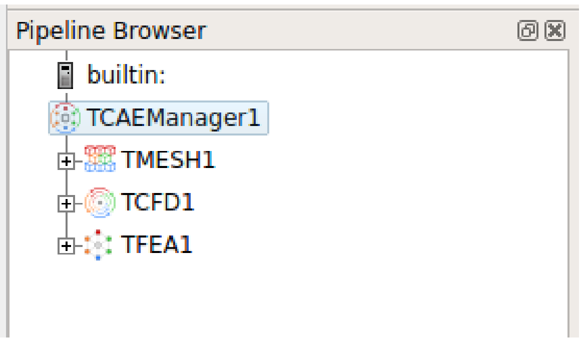
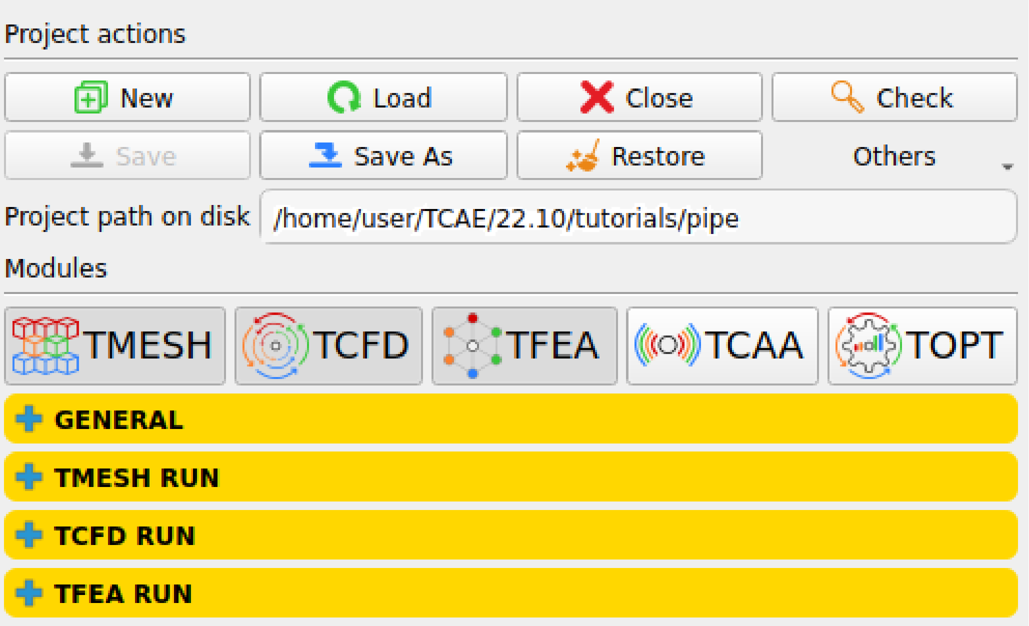

Pipeline Browser
builtin:
TCAEManager1
TMESH1
TCFD1
TFEA1

Figure 4.1: TCAEManager and its *Pipeline Browser*.

The settings of the TCAEManager in the *Properties Panel* are displayed in the figure 4.2 and are described in the following sections.

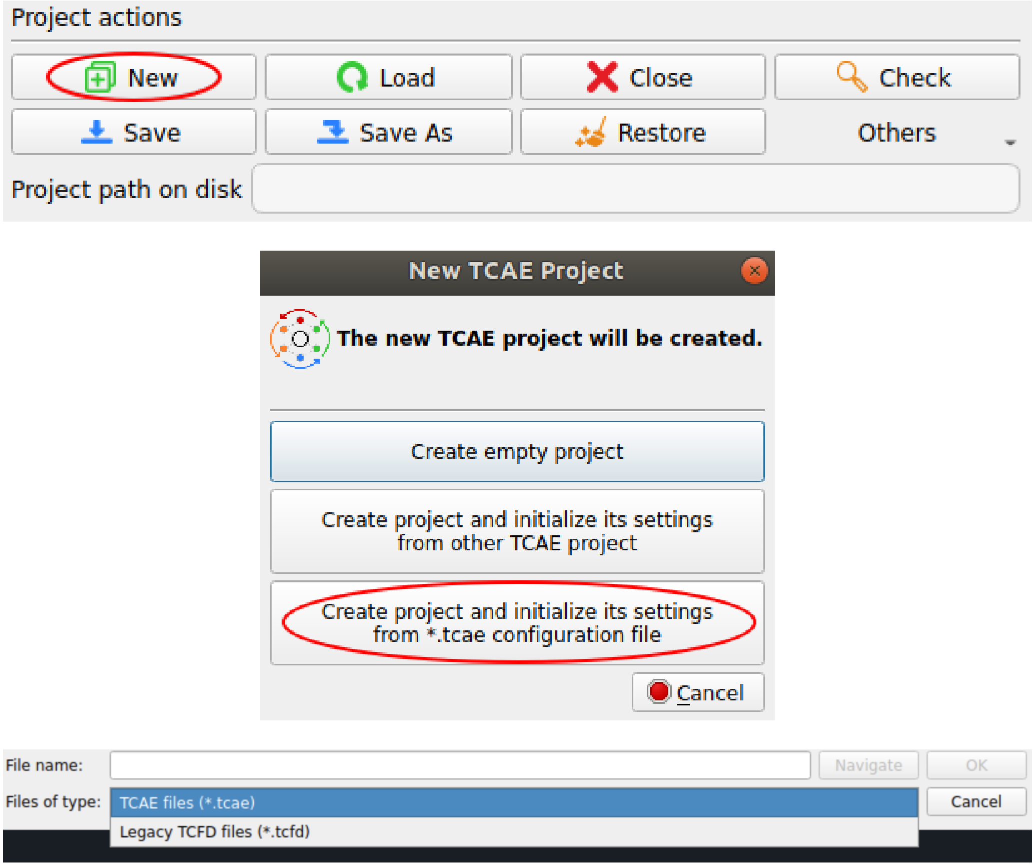
Setup project

Project folder can be loaded, saved or closed. As mentioned in the section 3.4, it is possible to load *.tcae file from the older versions of the TCAE and the *.tcfd file from TCFD version 19.10. To do this, user has to click on *New*, in the pop-up window check Load settings from the *.tcae configuration file and initialize the project using them option and then browse files and choose either TCAE files (*.tcae) or Legacy TCFD setup (*.tcfd) in the Open file menu. (see figure 4.3).



Project path on disk /home/user/TCAE/22.10/tutorials/pipe

Figure 4.2: TCAEManager – settings in the *Properties Panel*.


Project actions
New
Load
Close
Check
Save
Save As
Restore
Others
Project path on disk
New TCAE Project
Create empty project
Create project and initialize its settings from other TCAE project
Create project and initialize its settings from *.tcae configuration file
Cancel
File name:
Navigate
OK
Cancel
Legacy TCFD files (*.tcfd)

The new TCAE project will be created.

Files of type: TCAE files (*.tcae)

Figure 4.3: TCAEManager - switching between loading of the *.tcae files and legacy *tcfd files.

Check Setup

The button Check runs Check Setup, which means, that whole setup of all activated modules is checked for the inconsistencies or bad values, which may lead to unfeasible calculation. The problems can have two forms.

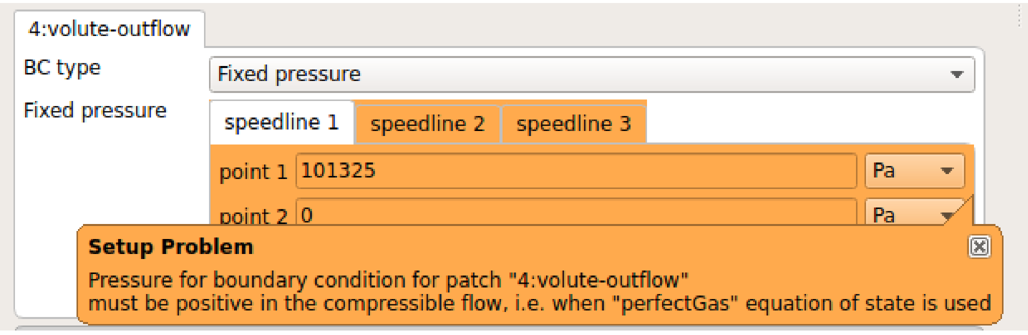
First is Error, which means, that some option has incorrect and the calculation cannot be performed, because there is be some discrepancy in the settings. Examples are:

- Zero pressure in the boundary condition for the compressible flow
- Non-connected components via interfaces
- Wrong combination of inlet and outlet boundary conditions

If there such a problem occurs, a relevant message with the explanation of the problem shows up in the window *Output Messages* and, simultaneously, the orange painted tooltip will point to the specific place in the setup menu (see Figure 4.4) what value is wrong, so user is able to remedy the issue.

Second possibility is Warning, which only shows up in the window *Output Messages*. This covers non-fatal problems, which don't cause the crash of the simulation, such as if the selected number of processor cores is higher than the actual number of physical cores of user's processor.

It is recommended to run Check each time before running any calculations (sec. 4.2), but if not, TCAE will run it anyway automatically after pressing the Save button. If some problem is found, the workflow is interrupted, and user must fix the setup.



101325

0

Setup Problem

Pressure for boundary condition for patch "4:volute-outflow"
must be positive in the compressible flow, i.e. when "perfectGas" equation of state is used

Figure 4.4: Check Setup - Error tooltip example

Modules buttons

These buttons (figure 4.2) controls the presented modules of TCAE. If one of them is pressed, the corresponding module shows in GUI, so user is able to adjust its settings, and then run it. Currently, as TCAE has modules TMESH, TCFD, TFEA, TCAA and TOPT there are five buttons present, each for seperate module. TMESH module has to be active always, therefore any combinations is possible between the remaining modules.

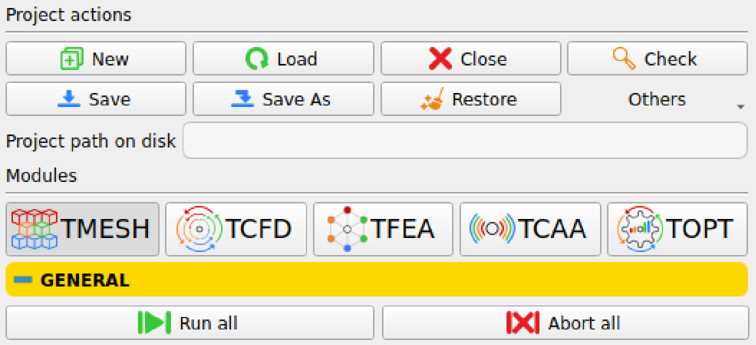


Figure 4.5: TCAEManager – GENERAL.

4.2 GENERAL

In this subsection the basic control of the TCAE workflow is available (Fig. 4.5).

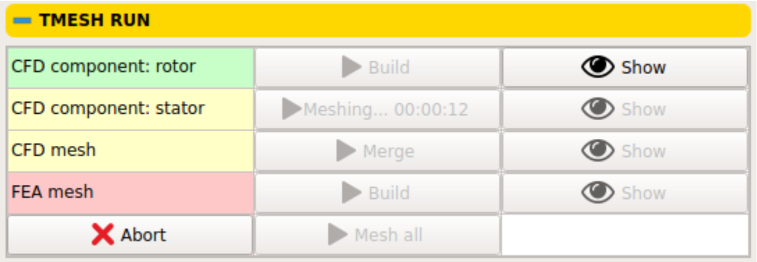
- "Run all" ▶ button will go through all modules and run all steps of the simulation. If more control is required, the user can use the buttons in the TMESH/TCFD/TFEA RUN sections.
- "Abort all" ☒ button will terminate the calculation phase in progress, no matter what it is (meshing, CFD processing / postprocessing, FEA processing / postprocessing). Note that there is some delay between pressing the button and the actual termination itself.
- "New" button will create empty TCAE project from scratch.
- "Load" button will load an existing TCAE project from disk.
- "Close" ✗ button will close the opened TCAE project.
- "Check" button will check whether the project setup is correct or if there are any problems.
- "Save" button will save the TCAE project to its path on the disk.
- "Save as" button will save the TCAE project to the new path on the disk.
- "Restore" button will restore the TCAE project on the disk, delete all results and only the project setup will remain.
- "Purge" button will clean the TCAE project on the disk and compress it to the smallest possible size to save space.
- "Delete" button will close and delete the opened TCAE project from the disk.

4.3 TMESH RUN

This section controls the execution of the TMESH module, i.e. managing the meshing.

It contains a table that gives the summary of the meshes, that will be created (see Fig. 4.6). There is one row for each component of the CFD mesh, one row for the whole CFD mesh and one row for the FEA mesh.

- The buttons "Build" ▶ is to be used to prepare the each component of the CFD mesh and/or the whole FEA mesh. In the case of external CFD mesh (those can be distinguished by the note "(ext)" displayed in the title of the component), the patches are renamed and mesh is analyzed and copied to the appropriate destinations. Otherwise the full meshing phase is performed. When the domain given by the boundary geometry is meshed. The progress of meshing is indicated by elapsed time displayed on the button.
- The button "Merge" ▶ is used to concatenate CFD meshes of individual components and create the final merged CFD mesh of the whole machine. The Merge button cannot be used until all components are successfully meshed.
- The effect of the button "Mesh all" ▶ is equivalent to using Build buttons for each of the CFD components, the Merge button and the Build button for FEA mesh.
- The success is indicated by a change of colour. Red colour means no mesh, yellow colour means invalid mesh (or meshing in progress), green colour means valid mesh.
- The progress of the meshing can be observed in the text format in *TCAE Output* window (see section 3.2).
- "Abort" ✖ button will stop the meshing immediately.
- Once a component's mesh (or merged mesh) is ready, it can be displayed. To display the mesh toggle the "Show" 👁 button. The mesh will be visible in the *RenderView* window (if it is present, otherwise it is needed to open it). The visibility of the meshes can be enabled/disabled by the *Output Ports* CFD Mesh and FEA Mesh of the TMESH module (see section 3.2 about the *Output Ports* and section 7).



TMESH RUN

CFD component: rotor	▶ Build	👁 Show
CFD component: stator	▶Meshing... 00:00:12	👁 Show
CFD mesh	▶ Merge	👁 Show
FEA mesh	▶ Build	👁 Show
✖ Abort	▶ Mesh all	

Figure 4.6: TCAEManager – TMESH RUN.

4.4 TCFD RUN

This section controls the execution of the TCFD module, i.e. managing the CFD calculation. It is divided into two collapsible subsections: "Processing" and "Postprocessing" (Fig 4.7).

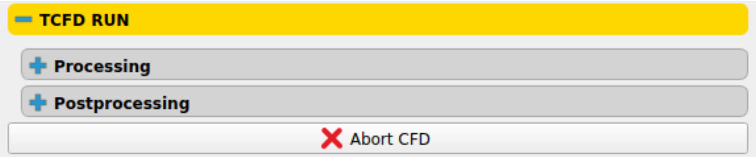


Figure 4.7: TCAEManager – TCFD RUN.

Under these subsections, the button "Abort CFD" ✕ is located. If pressed, it will gracefully terminate the calculation allowing for further use of TCAE. Note that there is some delay between pressing the button and the termination itself.

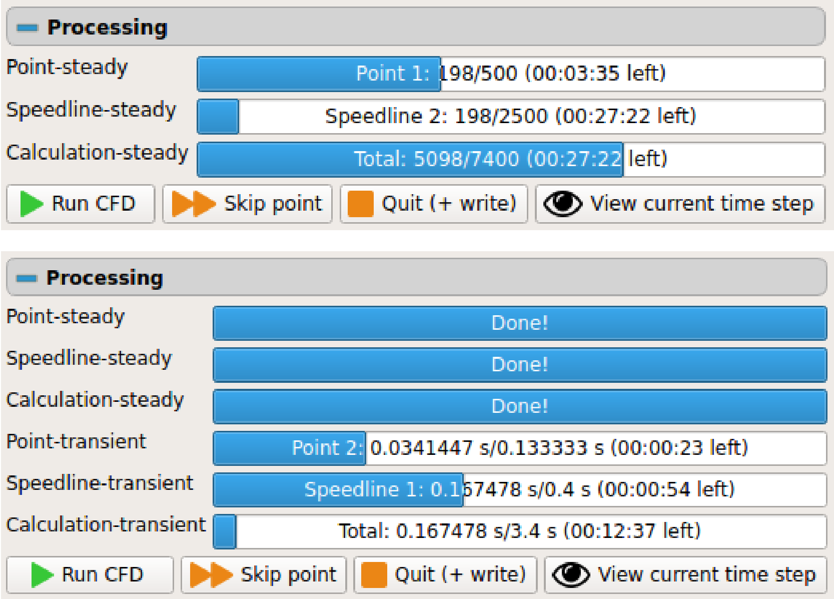
4.4.1 Processing

In this section the CFD calculation is manually controlled (Fig. 4.8).

- The button "Run CFD" ▶ will run the calculation of the physical fields for each simulation point. The progress of each point is shown in the point progress bar "Point-steady" and the progress of the whole calculation, which can be composed of a sequence of points, is shown in the calculation progress bar "Calculation-steady" (see figure 4.8). There are also the unsteady progress bars if the setup includes a transient run. The progress of the calculation can be observed in the text format in *TCAE Output* window (see section 3.2).
- The button "Skip point" ⏩ can be used during the calculation e.g. when the currently calculated point seems to be converged and the user wants to continue with the next point. Note that there is always a delay between pressing the button and the skipping action, which amounts to one or two iterations.
- The button "Quit (+ write)" ■ quits the calculation, but with all the postprocessing as if the calculation would finish successfully. Useful for terminating early if the results are satisfactory already.
- The button "View current time step" 👁 writes the results from the current iteration (in steady-state calculation) or from the current time step (in transient one) on disk, and the results are shown as a new item in the *Pipeline Browser* (Fig. 4.9), and it is possible to show them in the *RenderView* window (see 4.4.5), and manipulate them or extract some data from them using ParaView standard tools.

4.4.2 Run-time tuning

Simulation parameters can be also changed run-time in TCFD module. These parameters are still active, whereas the others are inactive, see Fig. 4.10. Particular changes affects the simulation after clicking on **Save** button in TCAEManager.



Processing

Point-steady: Point 1: 198/500 (00:03:35 left)

Speedline-steady: Speedline 2: 198/2500 (00:27:22 left)

Calculation-steady: Total: 5098/7400 (00:27:22 left)

Processing

Point-steady: Done!

Speedline-steady: Done!

Calculation-steady: Done!

Point-transient: Point 2: 0.0341447 s/0.133333 s (00:00:23 left)

Speedline-transient: Speedline 1: 0.167478 s/0.4 s (00:00:54 left)

Calculation-transient: Total: 0.167478 s/3.4 s (00:12:37 left)

Figure 4.8: TCAEManager – TCFD RUN: Processing. Apart from the graphical representation, it shows the index of the point currently being solved, number of iterations finished, number of iterations in total and estimated remaining time. In transient calculation (below) the iterations are replaced by the simulated time. The bottom three progress bars are only available for a transient calculation.

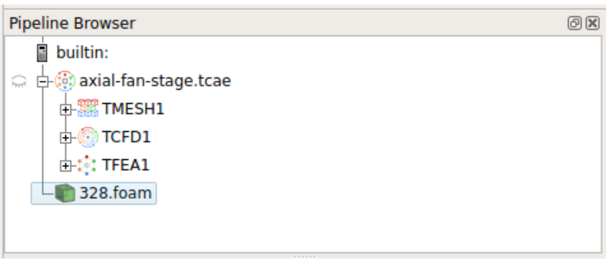
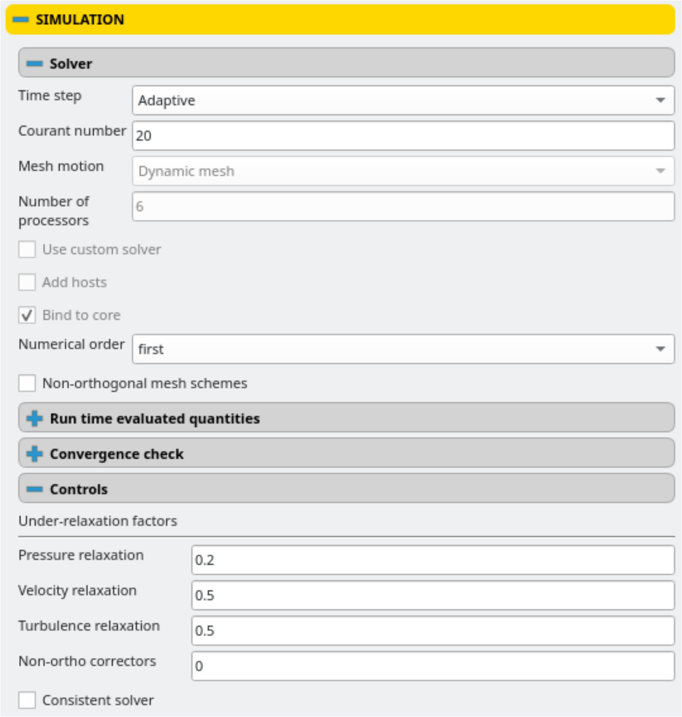


Figure 4.9: TCAEManager – CFD results of the current time step as a new item in *Pipeline Browser*.



SIMULATION

Solver

Time step: Adaptive

Courant number: 20

Mesh motion: Dynamic mesh

Number of processors: 6

☐ Use custom solver

☐ Add hosts

☑ Bind to core

Numerical order: first

☐ Non-orthogonal mesh schemes

Run time evaluated quantities

Convergence check

Controls

Under-relaxation factors

Pressure relaxation: 0.2

Velocity relaxation: 0.5

Turbulence relaxation: 0.5

Non-ortho correctors: 0

☐ Consistent solver

Figure 4.10: TCAEManager – TCFD RUN: TCFDSimulation menu during simulation.

The following parameters can be changed run-time:

- Numerical order
- Under-relaxation factors
- Solver parameters - bounding limits and tolerances
- Time step (in transient calculation only)
- PIMPLE algorithm correctors and tolerances (in transient calculation only)

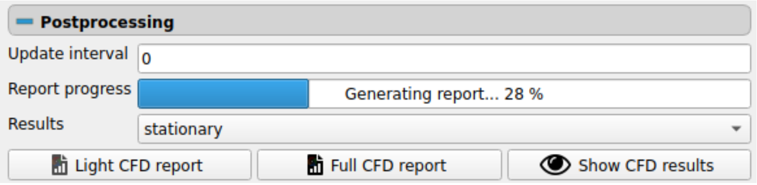
It is strongly recommended not to use this option during your final simulations and use it just for tuning and looking for the best parameter setup for the particular case. If you recompute the case again with the updated parameters from the very beginning you can get different results.

Any change of these run-time parameters does not trigger a request for recomputing the simulation. This rule is applied only for changing the parameters after the simulation.

4.4.3 Postprocessing

In this subsection user can manually trigger the composition of HTML report, and show calculation results (Fig 4.11).

- The button "Full report" will regenerate the report containing various extracted data, like the residua, efficiencies, pressure information etc. Some of the values and plots may


Postprocessing

Update interval: 0

Report progress: Generating report... 28 %

Results: stationary

Figure 4.11: TCAEManager – TCFD RUN: Postprocessing

not be available until the end of the run. The report is a HTML document, which can be displayed in the *HTML View* window. The visibility of the report can be enabled/disabled by the *Output Port* Report of the TCFD module (see section 3.2 about the *Output Ports* and section 11).

- "Light report" has a similar function as Full report, but produces only a subset of the full report, particularly the "Efficiency", "Head" and "Total pressure difference" sections. This is useful to monitor convergence of these results during the calculation.
- The option "Results" selects which results are to be shown by "Show CFD results" button – stationary or transient. In order to plot the results of a given computational point, select among various points on the speedline from the drop-down list next to the time drop-down list.
- "Update interval" – The report update is generally fast, but for long transient calculations or calculations with many points and speedlines it can take even a few minutes. The report is also automatically updated at the end of the calculation. Moreover, one can set the auto-refresh interval (in minutes) using this field (and confirm *Apply*). If a positive value is set, it will be used during the calculation.
- The bar "Report progress" indicates the progress of the report update.
- The button "Show CFD results" will show the results of the CFD calculation selected in the drop-down list Results in the GUI. There is one case for the whole stationary simulation and separate cases for individual points of the transient calculation.

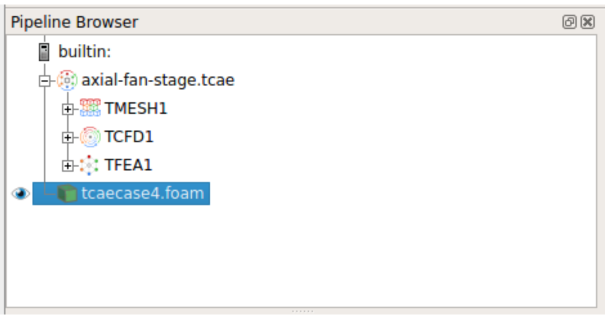
  Results appear as the new item in the *Pipeline Browser* (Fig. 4.12), and it is possible to show them in the *RenderView* window (see 4.4.5), and manipulate them or extract some data from them using ParaView standard tools.

4.4.4 Run-time monitoring

TCAEManager offers the run-time monitoring of the CFD calculation, so user can observe the convergence progress.

It is done through the two *Output Ports* of the TCFD module (see 11) – Quantities and Residuals, which produce the integral quantities each iteration.

To show these quantities, there has to be a *LineChartView* window in the *Views* part of GUI (see Sec. 3.2). By default there is one such a window present after launching TCAE. If not, simply open a new one.


Pipeline Browser

- builtin:
  - axial-fan-stage.tcae
    - TMESH1
    - TCFD1
    - TFEA1
- tcaecase4.foam

Figure 4.12: TCAEManager – CFD results as a new item in *Pipeline Browser*.

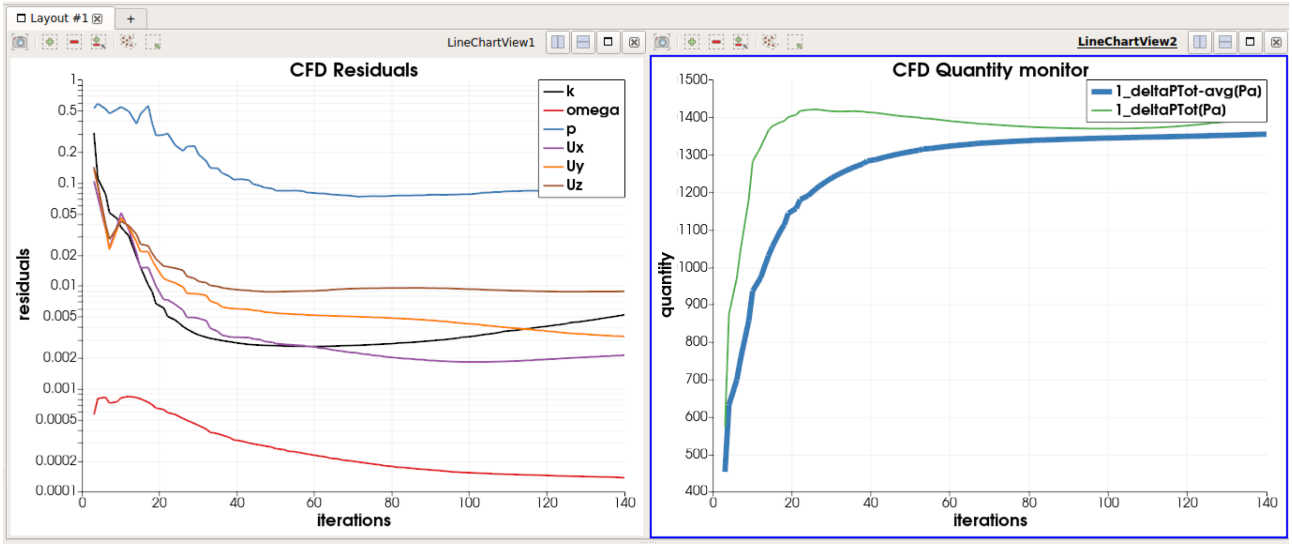


Figure 4.13: TCAEManager – Two *SpreadsheetView* windows displaying CFD Residuals and CFD Quantity monitor

Then, if the window is selected, select the desired *Output Port* (Quantities / Residuals) and make it visible (if not yet) in the active window through the eye icon 👁. The chart is then going to be drawn in the selected *LineChartView* window.

The Quantities *Output Port* produces live values of all integral quantities, that are computed each iteration. Available quantities depend on the setup in the Run time evaluated quantities section of the TCFD module (see Sec. 11.3.2). There, each Efficiency probe, Forces probe and Probe has the checkbox monitor, and if it is enabled (which is done by default), the quantities, that are computed by the particular probe, are pushed to the Quantities.

The Residuals *Output Port* shows the continuously computed residuals of each physical quantity.

For both the Quantities and Residuals one can adjust the chart properties for the given window, such as the axis ranges, scales etc in the tab *View* of the *Properties Panel*. Morover, in the tab *Display*, section *Series Parameters*, there is list of all of the present quantities, and by checking on/off they appear/disappear in the window *LineChartView*. There is also possible to change the actual appearance of the chart lines.

The Quantities and Residuals charts during the calculation are displayed in the Fig. 4.13.

4.4.5 Loading results

Results from the TCAE calculation can be loaded in the TCAEManager by clicking on the button Show results after, or View current time step during the calculation.

A new item appears in the *Pipeline Browser*, which has name TCFDResults.foam in the case of final results or *n*.foam for the current results, where *n* is the last finished iteration (time in the transient case), when the acquiring of the results was triggered.

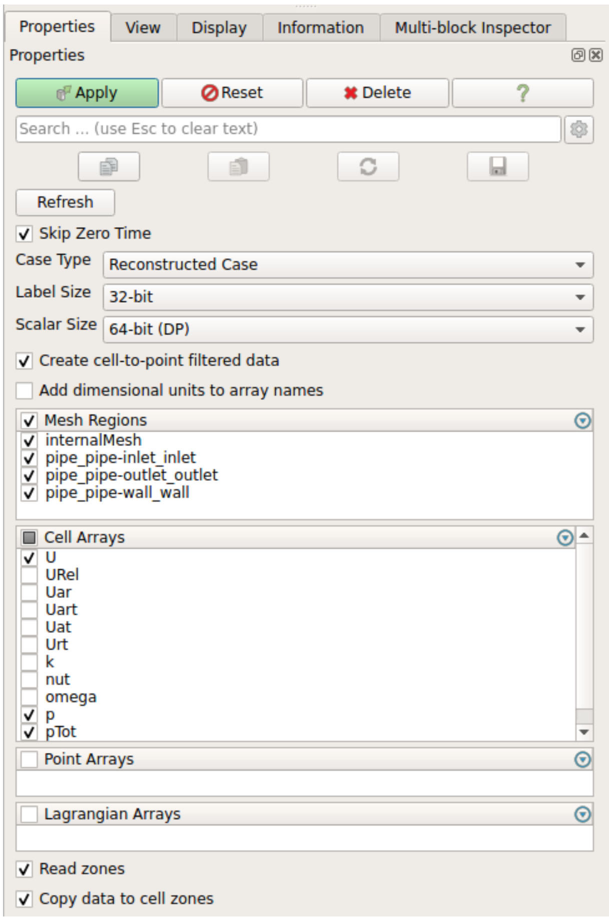
This new item, if selected, offers several options in the *Properties Panel* (Fig. 4.14). The most important are:

- *"Mesh Regions"* – In this window user can select which parts of the mesh will be loaded. The most needed option is *internalMesh*, i.e. the whole computational domain. Besides that, each patch is also available.
- *"Cell Arrays"* – In this window there is a list of available physical fields, and user can select which of these will be loaded.
- *"Read zones"* – If this checkbox is enabled, the individual mesh zones are loaded, which means, that it is then possible to show or hide single components.
- *"Copy data to cell zones* – this option has to be enabled, if the Read zones is switched on.

After adjusting these options, click *Apply* and the results will be loaded from the disk and made visible in the *RenderView*. Then, choose some non-zero simulation time in the toolbar (Fig. 4.15).

The visibility of whole mesh / components / patches is set in the tab *Multi-block inspector* of the *Properties Panel* (see Fig. 4.16). The actual displayed quantity is, along with its representation, selected in the top toolbar (Fig. 4.17).

The appearance of the *RenderView* and the results can be adjusted in tabs *View* and *Display* of the *Properties Panel*. With the results present in the *Pipeline Browser*, it is possible



Properties | View | Display | Information | Multi-block Inspector

Properties

Apply | Reset | Delete | ?

Search ... (use Esc to clear text)

Refresh

☑ Skip Zero Time

Case Type: Reconstructed Case

Label Size: 32-bit

Scalar Size: 64-bit (DP)

☑ Create cell-to-point filtered data

□ Add dimensional units to array names

☑ Mesh Regions
- ☑ internalMesh
- ☑ pipe_pipe-inlet_inlet
- ☑ pipe_pipe-outlet_outlet
- ☑ pipe_pipe-wall_wall

▨ Cell Arrays
- ☑ U
- □ URel
- □ Uar
- □ Uart
- □ Uat
- □ Urt
- □ k
- □ nut
- □ omega
- ☑ p
- ☑ pTot

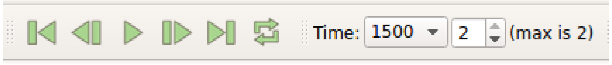
□ Point Arrays

□ Lagrangian Arrays

☑ Read zones

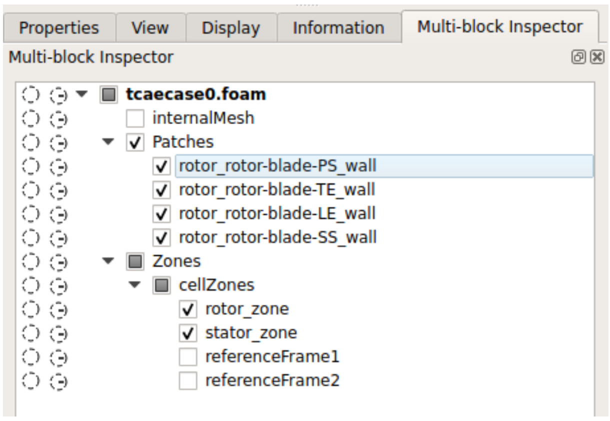
☑ Copy data to cell zones

Figure 4.14: TCAEManager – *Properties Panel* settings of the loaded results



Time: 1500 | 2 | (max is 2)

Figure 4.15: TCAEManager – Time settings in the top toolbar



Properties | View | Display | Information | Multi-block Inspector

Multi-block Inspector

- ▨ **tcaecase0.foam**
  - □ internalMesh
  - ☑ Patches
    - ☑ rotor_rotor-blade-PS_wall
    - ☑ rotor_rotor-blade-TE_wall
    - ☑ rotor_rotor-blade-LE_wall
    - ☑ rotor_rotor-blade-SS_wall
  - ▨ Zones
    - ▨ cellZones
      - ☑ rotor_zone
      - ☑ stator_zone
      - □ referenceFrame1
      - □ referenceFrame2

Figure 4.16: TCAEManager – *Multi-block inspector* - changing the visibility of the mesh parts

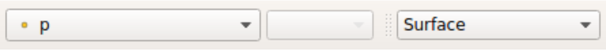


Figure 4.17: TCAEManager – Settings in the top toolbar - selected field *p* (static pressure) and *Surface* representation

to connect various ParaView postprocessing *filters* to manipulate them or extract some data from them.

4.5 TFEA RUN

This section controls the execution of the TFEA module, i.e. managing the FEA calculation. It is divided into two collapsible subsections: "Processing" and "Postprocessing" (Fig 4.18).

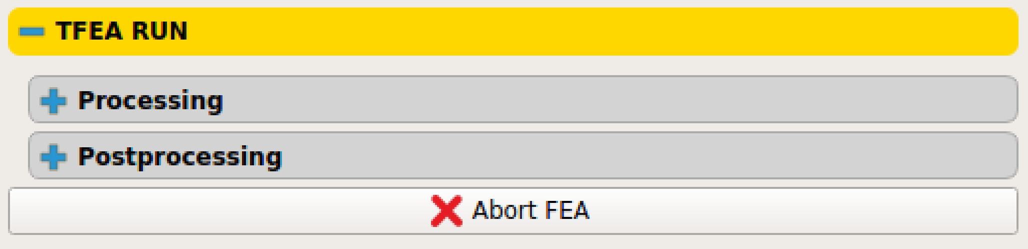


Figure 4.18: TCAEManager – TFEA RUN.

Under these subsections, the button "Abort FEA" ❌ is located. If pressed, it will terminate the calculation allowing for further use of TCAE. Note that there is some delay between pressing the button and the actual termination itself.

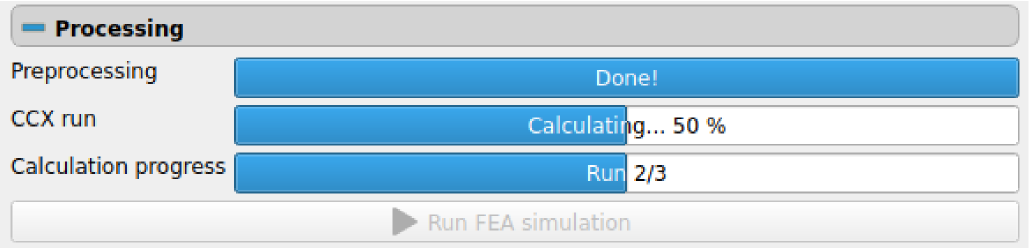
4.5.1 Processing

In this section the FEA calculation can be manually run (Fig. 4.19).

- The button "Run FEA simulation" ▶ will run the calculation. The first step is to find the mesh nodes in boundary regions corresponding to the boundary conditions set. Then the maping of the external data on the FEA mesh is performed, these are the boundary condition for the Fluid-Structure Interaction cases (if there is no FSI there is nothing to do in this step). Once the boundary conditions are set the actual calculations are run. If TFEA is dependent on the current TCFD run (which needs to be finished by now), static FEA calculation is computed for each CFD speedpoint. If, moreover, is the TCFD case transient, TFEA is performed for each speedline, the FSI data are taken averaged over the last transient window.

- The bar "CCX run" indicates the progress of the Calculix calculation.

- The bar "Calculation progress" marks completed calculation for each speedpoint.



Processing

Preprocessing: Done!

CCX run: Calculating... 50 %

Calculation progress: Run 2/3

Run FEA simulation

Figure 4.19: TCAEManager – TFEA RUN: Allows you to run TFEA separately and monitor the calculation progress.

4.5.2 Postprocessing

In this subsection user can create the FEA report and show the calculation results (Fig. 4.20).

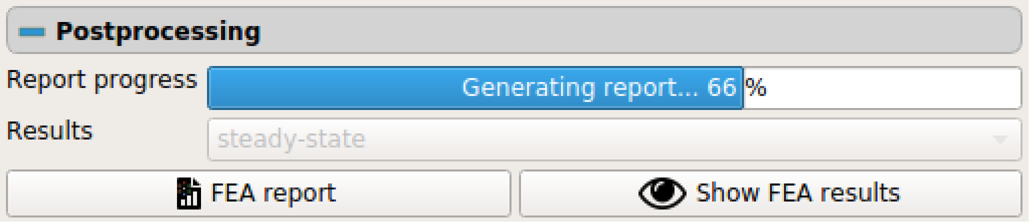
- The button "FEA report" will generate the report containing various data, like (if calculated) displacement and stress fields or eigenfrequencies data. The report is a HTML document, which can be displayed in the *HTML View* window. The visibility of the report can be enabled/disabled by the *Output Port* Report of the TFEA module (see section 3.2 about the *Output Ports*).

- The option "Results" selects which results are to be shown by "Show FEA results" button – steady-state or transient (only in case of Fluid-Structure Interaction with TCFD results).

- The bar "Report progress" indicates the progress of the report update.

- The button "Show FEA results" will show the results of the FEA calculation selected in the drop-down list Results in the GUI. There is a choice only in transient Fluid-Structure Interaction case, in this case, there is one option for the whole stationary simulation and separate options for individual points of the transient calculation.

  Results appear as the new item in the *Pipeline Browser* (Fig. 4.12), and it is possible to show them in the *RenderView* window, and manipulate them or extract some data from them using ParaView standard tools.



Postprocessing

Report progress: Generating report... 66 %

Results: steady-state

FEA report | Show FEA results

Figure 4.20: TCAEManager – TFEA RUN: Allows you to run TFEA separately and monitor the calculation progress.

4.6 TCAA RUN

In this section the execution of TCAA module is controled. There are two collapsible sections "Processing" and "Postprocessing" where the corresponding tasks of the simulation can be run and monitored. See Figure 4.21.

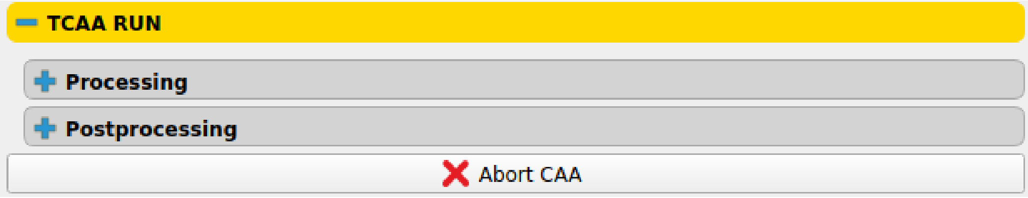


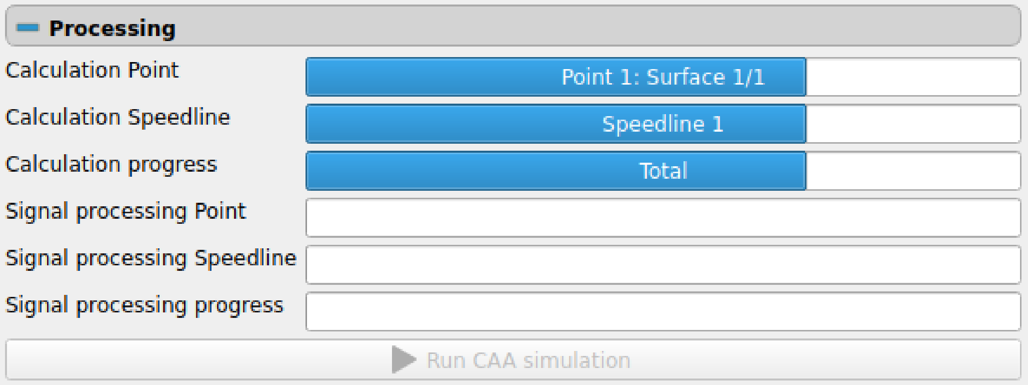
Figure 4.21: TCAEManager – TCAA RUN.

Under these subsections, the button "Abort CAA" ✕ is located. If pressed, it will terminate the calculation allowing for further use of TCAE. Note that there is some delay between pressing the button and the actual termination itself.

4.6.1 Processing

Here the run of CAA calculation can be executed, the layout is as shown in Figure 4.22.

- The button "Run CAA simulation" ▶ triggers execution of TCAA module. After the input data are loaded the acoustic analogy simulations starts. Its result is the time evolution of acoustic pressure values for each observer. Each point of each speedline runs separately. When multiple source surfaces are present there is one acoustic analogy run for each surface, the results of these separate runs are finally merged. When acoustic analogy calculation is successfully finished the signal processing calculation starts. The signals in each point of speedline are processed separately for each observer. The results of this step are acoustic quantities in frequency domain.
- The progress bar "Calculation point" indicates the progress in given point. It also says which source surface is being processed and what is their total number.
- The progress bar "Calculation speedline" indicates the progress in given speedline.
- Finally, the progress bar "Calculation progress" says total progress of the whole acoustic analogy simulation.
- The progress bar "Signal processing point" indicates the progress in given point. It also says which observer is being processed and what is their total number.
- The progress bar "Signal processing speedline" indicates the progress in given speedline.
- Finally, the progress bar "Signal processing progress" says total progress of the whole signal processing process. Note that the signal processing step is faster than acoustic analogy run.



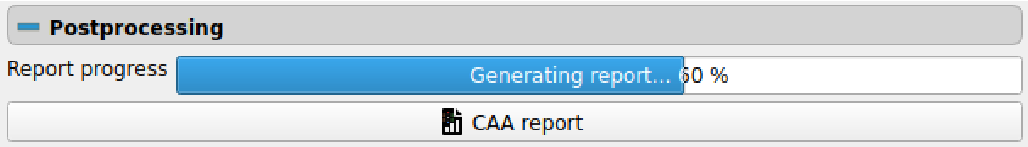
Calculation Point: Point 1: Surface 1/1
Calculation Speedline: Speedline 1
Calculation progress: Total
Signal processing Point
Signal processing Speedline
Signal processing progress

Figure 4.22: TCAEManager – TCAA RUN: Allows you to run TCAA separately and monitor the calculation progress.

4.6.2 Postprocessing

In this section the user can trigger and monitor the CAA report creation, see Figure 4.23.

- The button "CAA report" will generate the report containing graphs and tables of acoustic quantities for each observer and overall summary of the simulation. The report is a HTML document, which can be displayed in the *HTML View* window. The visibility of the report can be enabled/disabled by the *Output Port* Report of the TCAA module (see section 3.2 about the *Output Ports*).
- The bar "Report progress" indicates the progress of the report update.



Report progress: Generating report... 60 %

Figure 4.23: TCAEManager – TCAA RUN: Allows you to run TCAA separately and monitor the calculation progress.

Chapter 5

TCAE – Running in Terminal

Apart from the intuitive and interactive graphical user interface in ParaView, a preset TCAE workflow can be executed also in terminal using the project folder. This allows further batch processing and automatization of the calculations by embedding them in larger (e.g. blade design optimization) scripted workflows. The command-line utility used to execute TCAE setup is called CAEProcessor and can be used in the following way:

```
CAEProcessor --project pipe -- allrun
```

This command would prepare an calculation, in line with the settings present in the *setup* file of the project, write the calculation configuration into the directory *simulationRun1*, create the mesh, execute the solver and generate the same HTML report.

The same command can be used both in Linux and in Windows. In Windows, the command should be executed from the OpenFOAM-for-Windows (Cygwin) command line. In Linux, the OpenFOAM-in-Box environment[1] needs to be sourced. Remote execution of CAEProcessor over SSH is also supported. In that case, though, one should always forward X11 calls (`ssh -X`) to enable 3D processing of Turbo Blade Post renders, i.e. the blade-to-blade views and meridional averages. When those are not needed, simple non-forwarding session is sufficient.

NOTE: If there is a problem with terminal when it is not possible to write commands after starting midnight commander then it can be avoided by deleting last line in *bash.bashrc* file, which is in directory *etc* in TCAE installation directory, i.e., *C:\TCAE\XX.XX\cygwin64\etc* and manually sourcing TCAE after opening new terminal window by writing

```
source /etc/bash.bashrc.cfdsupport
```

5.1 CAEProcessor command line options

- `-h`, `--help`
  This option will produce only text output demonstrating usage of CAEProcessor similar to this chapter. The same effect happens also when no command line option is given at all.

[1] OpenFOAM-in-Box is distributed together with TCAE.

- `--inputfile <filepath>`
  Set the path for *.tcae or *.tcfd setup files when working with cases, which where saved in older version of TCAE, respectively TCFD.

- `--setup <filepath>`
  Same as "`--inputfile <filepath>`".

- `--import-geometries`
  If this option is specified after `--inputfile <filepath>` or `--setup <filepath>`, the external geometries/meshes for the TMESH will be imported into the project.

- `--project <projectpath>`
  Set the path to an existing project, which will be loaded.

- `--save | --write`
  Save the project (write on disk).

- `--save-as <path>`
  Save the project to a new path.

- `--apply-diff <filepath>`
  Change the project setting by specifying the path to the text file with TCAE configuration keywords, which are to be changed.

- `--mesh`
  Run the creation of mesh for CFD, FEA, or both (depends on the project settings). The project has to be saved.

- `--mesh-cfd`
  Run the meshing stage for CFD. The project must be saved.

- `--mesh-fea`
  Run the meshing stage for FEA. The project must be saved.

- `--calc`
  Run the CFD, FEA and CAA calculations. The project must be saved and CFD and FEA meshes must be valid and present.

- `--calc-cfd`
  Run CFD calculation, starting with the stationary one, optionally followed by transient. The project must be saved and a valid CFD mesh must be present.

- `--calc-fea`
  Run FEA calculation. The project must be saved and a valid FEA mesh must be present.

- `--calc-caa`
  Run CAA calculation. The project must be saved.

- `--exec`
  Combines options `--mesh` and `--calc`.

- `--optimize`
  Run optimization loop, if the project has TOPT module enabled.

- `--report`
  Generate HTML reports from the results of CFD, FEA and CAA simulations. It is possible to generate report any time after the case is written, even during simulation.

- `--report-cfd`
  Generate HTML report from the results of CFD simulation. It is possible to generate report any time after the case is written, even during simulation.

- `--report-cfd-light`
  This is a faster and less complete variant of the above; it contains fewer sections. A light report is useful to monitor convergence of efficiency and other basic parameters.

- `--report-fea`
  Generate HTML report from the results of FEA simulation. It is possible to generate report any time after the case is written, even during simulation.

- `--report-caa`
  Generate HTML report from the results of CAA simulation. It is possible to generate report any time after the case is written, even during simulation.

- `--report-opt`
  Generate HTML report from the results of optimization loop.

- `--allrun`
  Finally, this is a combination of the following switches: `--save`, `--mesh`, `--calc` and `--report`.

- `--restore`
  Clean the project and restore it to the original state. All results are deleted and progresses reseted.

- `--purge`
  Same as `--restore`, but the project will be compressed to a minimal size to occupy as little disk space as possible.

5.2 CAEProcessor command mode

The command-line utility CAEProcessor has a limited interactive or batch command mode. The command mode can be entered during the solver execution by issuing the SIGINT signal (corresponds to the keyboard shortcut *Ctrl + C* in common terminal emulators). If this is done, CAEProcessor will check the existence of a file named *caep-command.txt*; located at the same level as the project folder; if the file is found, commands are read from it (= “batch command mode”). If the file is not found, CAEProcessor will prompt the user for interactive entry of commands (= “interactive command mode”). The currently known commands are the following:

- `help`
  — Display list of available commands.

- `suspend`
  — Pause the calculation (in memory).

- `continue`
  — Resume the paused calculation.

- `abort`
  — CFD phase: Terminate calculation at the end of the next iteration.
  — Meshing phase: Terminate meshing immediately.

- `exit`
  — CFD phase: Terminate calculation immediately.
  — Meshing phase: Terminate meshing immediately.

- `quitWrite`
  — CFD phase: Quit calculation, but with all the postprocessing as if it would finish normally.

- `next`
  — CFD phase: Skip to the next calculation point.

- `kill`
  — Forcibly terminate CAEProcessor.

The command mode can be used to terminate a detached calculation with a checkpoint being written. Provided that "DIR" is the directory with the calculation and "PID" is the process ID of the running CAEProcessor program, the following two commands will do the trick:

```
# echo quitwrite > DIR/cfdp-command.txt
# kill -SIGINT PID
```

5.3 Environment variables

CAEProcessor reacts to several environment variables that can be used to control its operation. They are the following:

- `CAEP_VERBOSE` — If set, will increase verbosity of the CAEProcessor output, providing some additional (debugging) output.

- `CAEP_MKDIR_PERMISSIONS` — (Linux only.) If set to a valid octal number, will be used as the permission triplet when creating new directories. By default, CAEProcessor uses 770.

- `CAEP_MPIEXEC_ARGS` — (Linux only.) If set, will be used as a replacement for most arguments passed to *mpiexec*. This variable is useful for detailed mapping / ranking / binding of processes by job schedulers.

5.4 Running under task scheduler (Linux only)

CAEProcessor is compatible with the PBS scheduler (PBS Pro, Open PBS) and it will automatically detect that it is running within a submitted job and pass that information to the bundled MPI library. For this to work, the MPI library needs to be able to find the required PBS library on all compute nodes the calculation will be running on. This may require adding the path to the PBS library to the `LD_LIBRARY_PATH` environment variable in cases where the library is not in the system default location. You can verify that the PBS library is correctly being found by MPI by checking its dynamic dependencies after loading the TCAE environment. The following output indicates that the PBS library is not found and that CAEProcessor will not cooperate with PBS:

```
$ ldd $MPI_ARCH_PATH/lib/openmpi/mca_plm_tm.so | grep pbs
        libpbs.so.0 => not found
```

On the contrary, outputs like the one below confirm that MPI is able to find the PBS library and that it will be able to use the PBS resource and task manager.

```
$ ldd $MPI_ARCH_PATH/lib/openmpi/mca_plm_tm.so | grep pbs
        libpbs.so.0 => /opt/pbs/lib/libpbs.so.0 (0x00007f499b63d000)
```

A typical PBS submission script (submitted by `qsub run.sh`) looks like this:

```
#!/bin/bash
#PBS -N tcae-test
#PBS -l select=4:ncpus=16:mem=16gb
#PBS -l walltime=04:00:00

# load TCAE/OpenFOAM environment
. /opt/TCAE-21.09/OpenFOAM-dev/etc/bashrc-release

# change to the directory this script was submitted from
cd $PBS_O_WORKDIR

# run the simulation from the same directory
CAEProcessor -setup pipe.tcae -allrun
```

The number of parallel processes specified in the setup file needs to be compatible with the scheduler resource allocation. In the above example case, it would need to be equal to 64 (= 4 × 16).

Part II

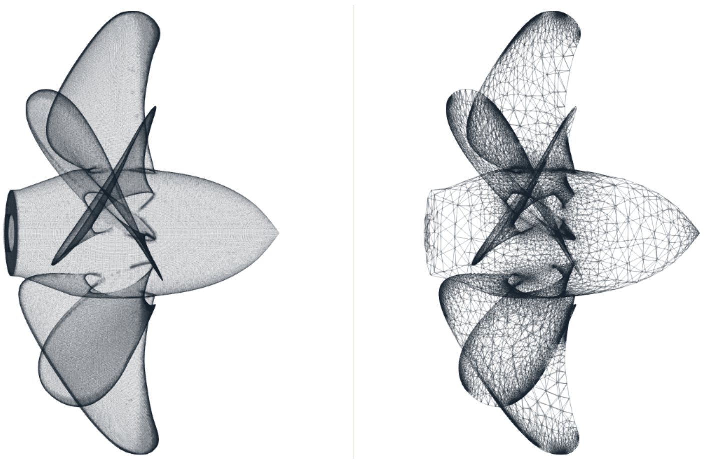
TMESH

Chapter 6

TMESH – Introduction

6.1 What is TMESH?

TMESH is one of the modules of the TCAE software, and is responsible for loading, creating, converting and analyzing of the computational meshes, which are then used in the TCFD and TFEA modules. The data workflow of the TMESH is diaplayed in the Fig. 6.1.



6.2 Meshes for CFD

Meshes for CFD simulation can be either created automatically from the user-supplied geometry, or the external meshes can be employed (6.1).

6.2.1 Geometry for CFD

TMESH is able to load STL geometries, and then automatically mesh them using snappyHexMesh application. STL files can be both ASCII and binary, and multisolid STL files are also supported.

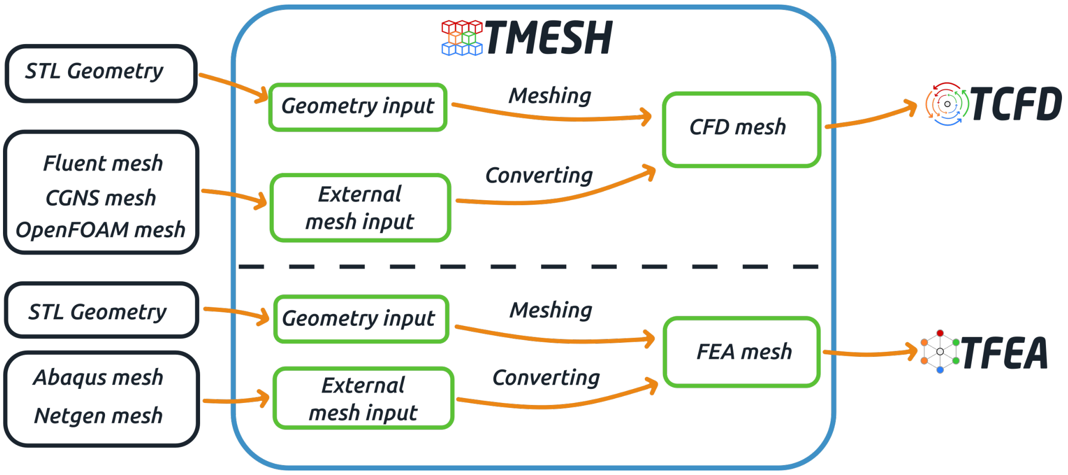


Figure 6.1: TMESH – data workflow

6.2.2 External meshes for CFD

TMESH also offers ability to load Your meshes created in some other software. These are loaded, and internally transformed to the TMESH specification. The supported formats are:

- Fluent meshes (MSH format)
- CGNS meshes
- OpenFOAM meshes

CGNS

TMESH can import CGNS meshes in both ADF and HDF5 format. Due to the great generality of the CGNS data format, TMESH currently supports just a subset of all possible CGNS layouts and features. TMESH first combines all bases in the CGNS file, and then interprets zones within those bases as available components. Then:

- if a zone is structured (hexahedral), TMESH will use its 6 boundaries as patches and rename them using the BC names as given in the appropriate BC node, provided that the BCs apply to the whole geometrical patches and not only to their subsets,
- if a zone is unstructured, TMESH will assume that the CGNS zone contains just one volume section to be used as the volume mesh, and right enough compatible surface sections to cover the volume mesh to be used as patches (without further reference to BCs, even if present).

If an unstructured mesh is in "mixed" format, TMESH expects that the element connectivity is stored in the interleaved format. If an unstructured mesh is polyhedral, TMESH requires the volume mesh element connectivity data to be stored in the interleaved format, and corresponding polygonal patches too, all within the same CGNS zone. In the present version of the CGNS reader, faces of the polyhedral volume elements must be sorted in the OpenFOAM convention, i.e., boundary faces must be at the end of the face list and always grouped by patch. Polygonal patch element connectivity data then must consist of (boundary) faces in the same order and with exactly the same point definitions as used by the volume section.

TMESH does not support mixed/polyhedral meshes written by CGNS 3.4+ (post-CPEX0041), which use split element connectivity into point references and data offsets.

6.3 Meshes for FEA

Our Finite Element solver works exlcusively on tetrahedrons, as a concequence we spupport only tetrahedral meshes. You cannot create, load or process any other then tetrahedral mesh (i.e., hexahedral meshes are not supported in the context of TFEA simulations).

TMESH can create new Finite Element meshes from geometry files or load and process external tetrahedral meshes. As a geometry file STL and STEP files are accepted, the meshing itself is then provided by NetGen [24] or GMSH[35]. Whose native formats, VOL and MSH, respectively together with INP format are possible inputs to TMESH as external FEA meshes.

The first format of external meshes that is currently supported is VOL. VOL format is native to NetGen, it contains all the meshing information and is therefore easy to load back to NetGen for subsequent mesh processing. However, we are not aware of an easy way of converting mesh from another format to VOL. Because of this, the way to obtain an VOL file is to mesh a geometry in NetGen and then save it as VOL. TMESH creates VOL mesh everytime it meshes an STL geometry or process an external mesh, the resulting file is usually stored on the path `case_directory/TMESH/FEA/final/TFEAFinalMesh.vol`. This mesh can be used as an external mesh for an other TFEA simulation (with the same geometry, of course) which will save some time of the meshing procedure.

MSH format is native for GMSH software, this format is a little bit more common in the FEA community and some mesh converters support it. Note that the MSH extension is also used for FLUENT meshes but those are different meshes so don't be confused.

The INP format of mesh is the Abaqus mesh format. CalculiX, the Finite Element tool we use for FEA, reads meshes in this format. INP is a simple file containig a list of points and a list of elements. CalculiX reads the order of finite elements from the mesh, for quadratic elements it requires also points in the middle of edges. However, TMESH can convert between linear and qudratic meshes so the user don't have to care about the order of external INP mesh.

6.4 TMESH directory structure

TMESH directory is subdirectory of TCAE case (see 3.2).

The whole directory should be read-only, and its basic structure is shown in the table 6.1. The TMESH directories are rarely needed to be examined, and if so, mostly to look at some meshing logs.

🗀 project_folder	Project directory
└ 🗀 simulationRun	
└ 🗀 TMESH	all files, that belong to TMESH module
└ 🗀 CFD	folder with CFD meshes, later used in TCFD module
└ 🗀 component1	OpenFOAM directory with mesh for CFD component 1
└ 🗀 component2	OpenFOAM directory with mesh for CFD component 2
└ 🗀 . . .	
└ 🗀 final	OpenFOAM directory with final CFD mesh, merged from individual component meshes
└ 🗀 FEA	folder with FEA meshes, later used in TFEA module
└ 🗀 component1	directory with mesh for FEA component 1
└ 🗀 component2	directory with mesh for FEA component 2
└ 🗀 . . .	
└ 🗀 final	directory with final FEA mesh, merged from individual component meshes
└ 🗀 logRun	output text logs of all of the applications, that are run by the TMESH

Table 6.1: TMESH – Directory structure

Chapter 7

TMESH – GUI Setup & Options

Module TMESH in the GUI is selected by the clicking on the *TMESH* item in the *Pipeline Browser*, or on one of its *Output Ports*, which are:

- Settings
  Using this port the keywords and their values of the TCFD module are displayed as a table in *SpreadSheet View*

- CFD Geometry
  If the input geometry for CFD meshing is loaded, it is shown in *RenderView* through this port.

- FEA Geometry
  If the input geometry for FEA meshing is loaded, it is shown in *RenderView* through this port.

- CFD Mesh
  If the CFD mesh is finished, it might be shown in *RenderView* and its visibility is controlled through this port.

- FEA Mesh
  If the FEA mesh is finished, it might be shown in *RenderView* and its visibility is controlled through this port.

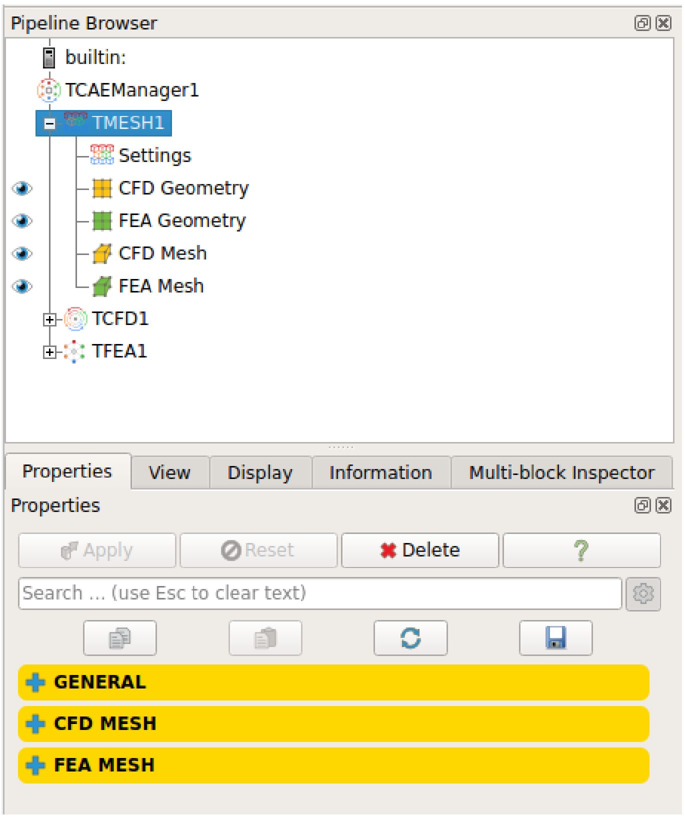
After selection the TMESH, user can see what is depicted in the Fig. 7.1.

In the *Properties Panel*, there are three sections. "GENERAL", "CFD MESH" and "FEA MESH".

7.1 GENERAL

This section contains an important option Mesh output. It determines, which meshes are going to be created. The options are:

- CFD Mesh
  Only mesh for CFD simulation is set up, section CFD MESH is visible.



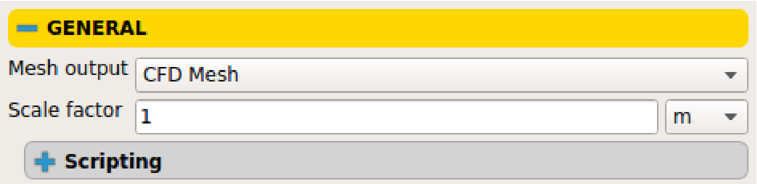
Pipeline Browser

- builtin:
- TCAEManager1
- TMESH1
  - Settings
  - CFD Geometry
  - FEA Geometry
  - CFD Mesh
  - FEA Mesh
- TCFD1
- TFEA1

Figure 7.1: TMESH – *Pipeline Browser* and *Properties Panel*

- FEA Mesh
  Only mesh for FEA simulation is set up, section FEA MESH is visible.

- CFD & FEA Mesh
  Meshes for both CFD and FEA simulations are set up, both sections CFD MESH and FEA MESH are visible.



GENERAL

Mesh output: CFD Mesh

Scale factor: 1 m

Scripting

Figure 7.2: TMESH – GENERAL section.

The second entry is the "Scale factor", which sets the scale of the length units to be used when processing the input geometry (for both CFD and FEA) and some other form entries. The options, which are affected by the Scale factor are described in the following sections of this manual as "Affected by the Scale factor". If "1" is given, then TMESH assumes that all lengths and point coordinates are in meters, whereas if (e.g.) "0.001" is given, it is assumed that they are in millimeters. This setting is global: It is not possible to set different length scales for the geometry and for the form entries, or to have individual geometry files in different scales.

Watch out!

It is extremely important to set the parameter Scale factor correctly as it defines the scale of the STL model. Wrong Scale factor will easily make the whole geometry ten- (hundred-, thousand-) times bigger or smaller than it is in reality, resulting in a completely meaningless calculation!

Besides that, just the submenu "Scripting" is present in this section.

7.1.1 Scripting

For experienced users it is possible to extend the TMESH workflow by custom scripts `(advanced)` (Fig. 7.3). These scripts are expected to be written in basic Python 3.7 and they are executed at specific moments during the workflow. These "Execution points" are:

- "afterWrite"
- "beforeCFDMeshing"
- "afterCFDMeshing"
- "beforeFEAMeshing"
- "afterFEAMeshing"

Their meaning is obvious. It is allowed to assign multiple execution points to a single script.

The script can use the predefined variable CaseDirectory, which contain the full absolute path to the TCAE case directory (in the figure 6.1 it is the top level directory tcaecase). Besides the standard Python functions one can also use the TCAE-specific functions `SetEntry` and `WriteFile`. For example the following script

```
    SetEntry("TMESH/meshFactory/component2/system/snappyHexMeshDict",
"castellatedMeshControls/refinementRegions", "surface.stl")
    SetEntry("TMESH/meshFactory/component2/system/snappyHexMeshDict",
"castellatedMeshControls/refinementRegions/surface.stl/mode", "distance")
    SetEntry("TMESH/meshFactory/component2/system/snappyHexMeshDict",
"castellatedMeshControls/refinementRegions/surface.stl/levels", "((50 3)(150 2)(300 1))")
    WriteFile("TMESH/meshFactory/component2/system/snappyHexMeshDict")
```

adjusts the meshing of the CFD mesh - it will change the refinement levels based on the distance from the geometry (level 3 up to the distance 50, level 2 up to 150 and level 1 up to 300), followed by writing the modified file. The function `SetEntry` does all modifications in memory and the result is written to disk only when the function `WriteFile` is used, or during writing requested by the workflow. Apart from the two them, there are also two another access functions

```
    RenameEntry(<file>, <entry>, <newname>)
    DeleteEntry(<file>, <entry>)
```

with obvious purpose: The former changes the name of an entry (i.e. the keyword that introduces the entry), whereas the latter erases the whole entry from the file.

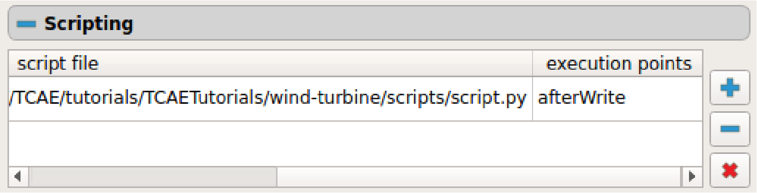

Scripting
script file
execution points
/TCAE/tutorials/TCAETutorials/wind-turbine/scripts/script.py
afterWrite

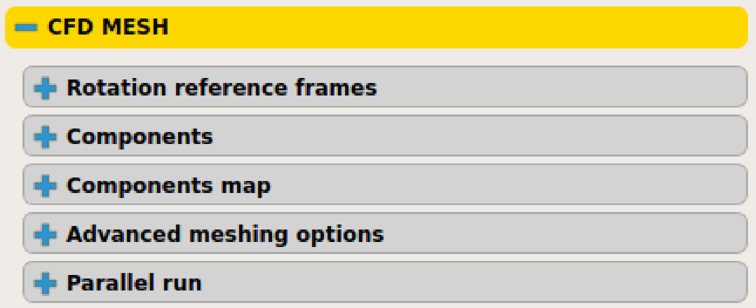
Figure 7.3: TMESH – Scripting.

7.2 CFD MESH

This section includes all adjustments for the CFD mesh, so it is available only if the value of the Mesh output is either CFD Mesh or CFD & FEA Mesh. Its subsections that are shown in the figure 7.4. The meshing itself is run in the TCAEManager and it is then passed on to the TCFD module.

7.2.1 Reference frames

The section "Rotation reference frames" (Figure 7.5) describes a general setup of reference frames. Multiple reference frames are needed in case of simulating the machines, where some parts are rotating. Here the reference frames with arbitrary axis are created, and they are used afterwards in "Components" section, where each component and patch has assigned its reference frame.



CFD MESH

Rotation reference frames

Components

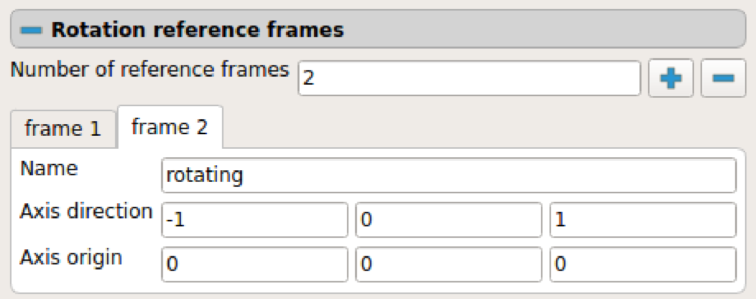
Components map

Advanced meshing options

Parallel run

Figure 7.4: TMESH – CFD MESH section with its subsections.

Then, in TCFD, each reference frame might be set as rotating with desired angular velocity, and by means of this approach, it is possible to execute simulations of rotating machines, where arbitrary number of parts rotate (each of them with its independent angular velocity), and some parts are static.



Rotation reference frames

Number of reference frames: 2

frame 1 | frame 2

Name: rotating

Axis direction: -1 | 0 | 1

Axis origin: 0 | 0 | 0

Figure 7.5: TMESH – Reference frames setup.

The number of reference frames is controlled by field "Number of reference frames". When the number of frames changes, the number of panels in this section is adjusted accordingly. Every frame holds following parameters:

- "Name" defines user-defined name of the reference frame.
- "Axis" defines the components of the axis direction vector.
- "Origin" defined the coordinates of the axis origin.

To make the adjustments easier, the axis of the reference frame, which is currently selected, is visualized in *RenderView* as thin black line.

Important note:

For stationary frames having rotational symmetry, it is important to define axis of rotation correctly. TMESH uses this information for applying appropriate boundary condition for outlet/inlet interfaces or rotationalAMI boundary patches.

7.2.2 Components

The computational domain can be split into any number of sub-domains – the components, which are meshed individually and are individually postprocessed. Each component can have any number of inlets and outlets and can be connected to any number of other components through interface patches. A sample setup of the "Components" section is depicted in Figure 7.6.

At any time, the current topology is graphically represented in the graph in the Components map (see section 7.2.3).

Several parameters are common to all components:

- Checkbox "Bounding box" enables bounding box. Using this feature, TCAE can be used for simulations of external flows (preferably in conjunction with the virtualTunnel Type in TCFD, see 11.1). If "Bounding box" is checked, axis oriented box is added, which works as a virtual wind tunnel, and another section with following entries will show up (Figure 7.7):
  - "Flow direction" entry determines the direction of the flow through the bounding box with respect co coordinate frame axis. The actual velocity vector might slightly differ from this direction (depends on the type of the inlet boundary condition), as purpose of "Flow direction" is to decide, which patches of the box will be inlet and outlet. There is also an option None, which means, that there is neither inlet nor outlet and all bounding box patches are walls.
  - "Bounding box point 1" sets the coordinates of the one vertex of the box.
  - "Bounding box point 2" sets the coordinates of the opposite vertex.

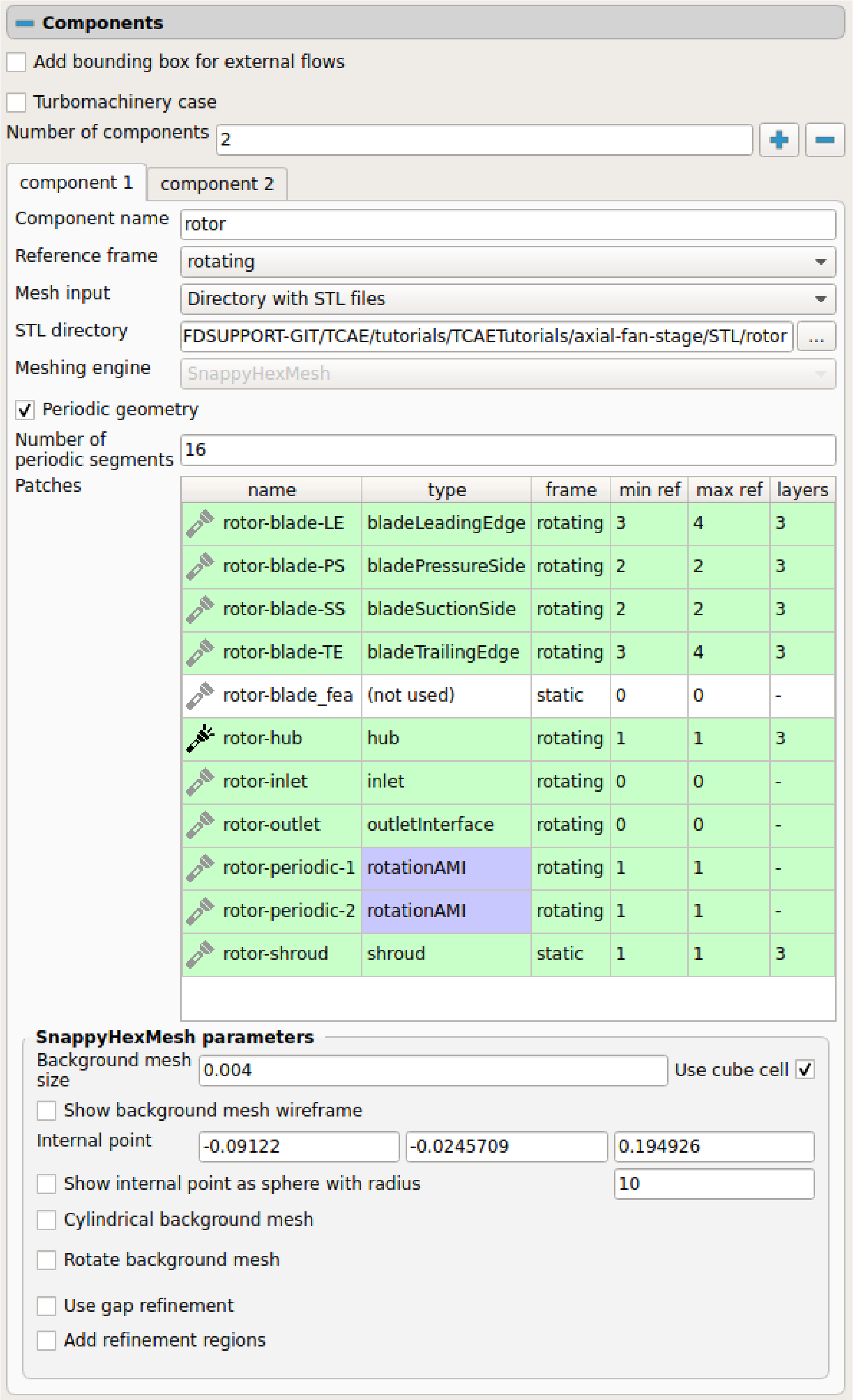
  Bounding box is displayed as a wireframe in the *RenderView*, and flow direction as two arrows in the inlet and outlet patches (Figure 7.8). If the bounding box is added, its patches will appear in the Patches table. These patches are named bb_inlet, bb_outlet, x_max, x_min, y_max, y_min, z_max, z_min. Only six of them are used at any time (inlet, outlet and four sides), and it depends on the value of the Flow direction. For instance, if Flow direction is Plus X axis, following patches are added: bb_inlet, bb_outlet, y_max, y_min, z_max, z_min.

- "Turbomachinery case" switch - if enabled, some additional patch types and meshing parameters, that are useful when meshing the turbomachinery, are shown.

The number of components is controlled by the field "Number of components". When the number of components changes, the number of panels in this section and also in the advanced mesh properties panels is adjusted accordingly. Every component needs a water-tight boundary geometry divided into non-overlapping segments or a ready-to-use mesh. Each component has these parameters:

- "Component name", which sets the name of the component used in patch names and report.

- "Reference frame" sets a frame from Reference frames section. If the selected reference frame is then set as rotating in TCFD, the whole component will rotate. Otherwise, the component will be stationary.



Components

□ Add bounding box for external flows

□ Turbomachinery case

Number of components: 2

Component name: rotor

Reference frame: rotating

Mesh input: Directory with STL files

STL directory: FDSUPPORT-GIT/TCAE/tutorials/TCAETutorials/axial-fan-stage/STL/rotor

Meshing engine: SnappyHexMesh

☑ Periodic geometry

Number of periodic segments: 16

Patches

name	type	frame	min ref	max ref	layers
rotor-blade-LE	bladeLeadingEdge	rotating	3	4	3
rotor-blade-PS	bladePressureSide	rotating	2	2	3
rotor-blade-SS	bladeSuctionSide	rotating	2	2	3
rotor-blade-TE	bladeTrailingEdge	rotating	3	4	3
rotor-blade_fea	(not used)	static	0	0	-
rotor-hub	hub	rotating	1	1	3
rotor-inlet	inlet	rotating	0	0	-
rotor-outlet	outletInterface	rotating	0	0	-
rotor-periodic-1	rotationAMI	rotating	1	1	-
rotor-periodic-2	rotationAMI	rotating	1	1	-
rotor-shroud	shroud	static	1	1	3

SnappyHexMesh parameters

Background mesh size: 0.004 Use cube cell ☑

□ Show background mesh wireframe

Internal point: -0.09122 -0.0245709 0.194926

□ Show internal point as sphere with radius 10

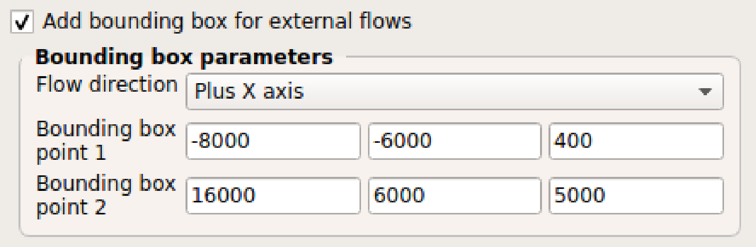
□ Cylindrical background mesh

□ Rotate background mesh

□ Use gap refinement

□ Add refinement regions

Figure 7.6: TMESH – Components section.



☑ Add bounding box for external flows

Bounding box parameters

Flow direction: Plus X axis

Bounding box point 1: -8000, -6000, 400

Bounding box point 2: 16000, 6000, 5000

Figure 7.7: TMESH – bounding box setup

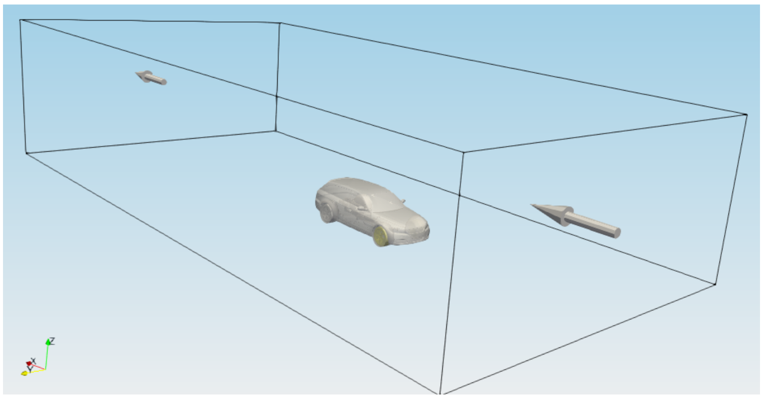


Figure 7.8: TMESH – bounding box representation in *RenderView*.

- "Mesh input" sets the source of the geometry or external mesh. Possible options are Directory with STL files, One multi-solid STL file, External Fluent Mesh, External OpenFOAM Mesh, External CGNS Mesh.

- "Meshing engine" selects the meshing system to use for creating the mesh, if Directory with STL files or One multi-solid STL file is selected as Mesh input. Currently only available option is SnappyHexMesh (part of an OpenFOAM SW).

- "Periodic geometry" makes the simulation periodic, so the input geometry represents only a one periodic segment of the full wheel.

- "Number of periodic segments" appears, if Periodic geometry switch is enabled, and gives the number of the segments that put together whole wheel.

- Directory with STL files – The directory must contain STL files (both ASCII and binary) which, when merged, give rise to a water-tight boundary surface. Note that the STL file names should consist only of alpha-numerical characters, dashes and underscores.

- Multi-solid STL file – The input is a single STL file containing multiple solids. Every solid must have a unique name within the STL file. That name will be used for naming the patch. Please make sure that the solid names consist only of alpha-numerical characters, dashes and underscores.

- External OpenFOAM mesh – The patches (and mesh) are taken from an already existing OpenFOAM mesh. The mesh creation step for this component will be skipped.

- External Fluent mesh – The patches (and mesh) are taken from an already existing mesh in MSH format. The mesh creation step for this component will be skipped. The MSH mesh reader is compatible both with the ASCII and binary formats.

- External CGNS mesh – The patches (and mesh) are taken from an already existing mesh in CGNS format. The mesh creation step for this component will be skipped.

If the new geometry/external mesh is loaded, it is displayed in *RenderView* and user can switch its visibility by enabling/disabling the output port *CFD Geometry* in *Pipeline Browser* (see figure 7.1). If the geometry can't be seen, it might be caused by the wrong position of the camera in *RenderView*. To focus the camera on the geometry try clicking on the *Reset* or *Zoom To Data* buttons in the top toolbar (figure 7.9).

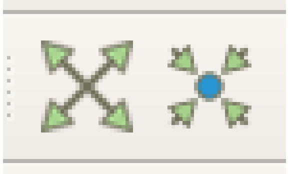


Figure 7.9: *Reset* and *Zoom To Data* buttons in toolbar.

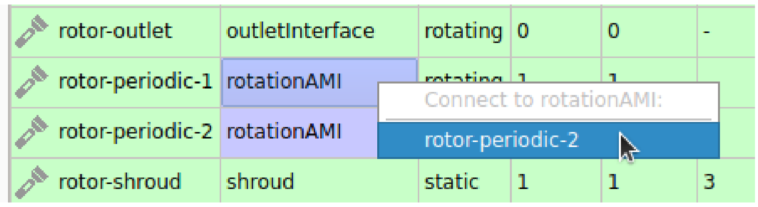
Patches table

Table "Patches" gives summary of all patches for the selected component, where each row of this table represents one patch and its properties. First column "name" shows the name of the patch, which is non-editable. Second column, "type", sets the patch type, which is one of the following:

- inlet – Entry to the simulated machine, mostly to the first component.

- outlet – Exit from the simulated machine, mostly the last component.

- internalAMI – Internal non-conformal mesh interface. It is necessary to connect this patch to another internalAMI patch within the same component by right-clicking on the row and selecting the patch, see Figure 7.10. This change will be indicated by a colour change of the rows of the connected patches.



rotor-outlet	outletInterface	rotating	0	0	-
rotor-periodic-1	rotationAMI	rotating	1	1	
rotor-periodic-2	rotationAMI				
rotor-shroud	shroud	static	1	1	3

Figure 7.10: TMESH – connecting the internalAMI / translationAMI / rotationAMI patches.

- translationAMI – Periodic interface in simulation of a translationally symmetric domain. It is necessary to connect this patch to another translationAMI patch within the same component by right-clicking on the row and selecting the patch (figure 7.10). This change will be indicated by a colour change of the rows of the connected patches.

- rotationAMI – Patches on the boundaries of a segment, which are (periodically) mapped to each other. It is necessary to connect this patch to another rotationAMI patch within the same component by right-clicking on the row and selecting the patch (see figure 7.10). This change will be indicated by a colour change of the rows of the connected patches.

- empty – Boundary in non-physical, artificial, dimension; used in two-dimensional simulations.
- symmetry – Special symmetry boundary condition for calculation of just a half of a perfectly symmetrical system. For compatibility reasons, using this boundary condition causes using segregated linear solver (instead of default coupled) for velocity. This has, in vast majority of cases, no influence on results.
- wall – General wall.
- hub – A specific type of wall.*
- shroud – A specific type of wall.*
- blade – A specific type of wall.*
- bladePressureSide – A specific type of wall.*
- bladeSuctionSide – A specific type of wall.*
- bladeLeadingEdge – A specific type of wall.*
- bladeTrailingEdge – A specific type of wall.*
- bladeHubFillets – A specific type of wall.*
- bladeShroudFillets – A specific type of wall.*
- bladeCap – A specific type of wall.*
- cutWater – A specific type of wall.*
- inletInterface – Entry to other components, connected to some preceding component. It is necessary to connect this interface to a specific outletInterface patch in another component by right-clicking on the row and selecting the component and patch (see figure 7.11). This change will be reflected in the component graph.**

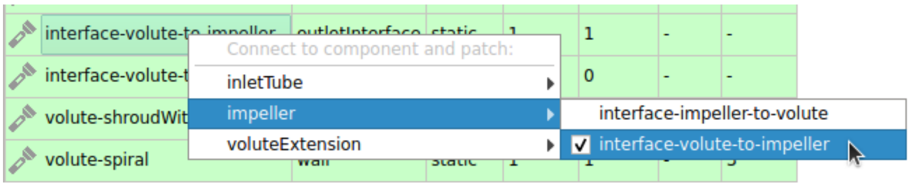

interface-volute-to-impeller
interface-volute-t
volute-shroudWit
volute-spiral
Connect to component and patch:
inletTube
impeller
voluteExtension
interface-impeller-to-volute
☑ interface-volute-to-impeller

Figure 7.11: TMESH – connecting the inletInterface / outletInterface / freestreamInterface patches.

- outletInterface – Exit from other components, connected to some following component. It is necessary to connect this interface to a specific inletInterface in another component by right-clicking on the row and selecting the component and patch (figure 7.11). This change will be reflected in the component graph.**

- freestreamInterface – For interfaces where both flow direction may occur, e.g., open propellers. It is necessary to connect this interface to a specific freestreamInterface patch in another component by right-clicking on the row and selecting the component and patch (figure 7.11). This change will be reflected in the component graph. This type always imposes cyclicAMI boundary condition.

** Available only if Turbomachinery case is enabled.*
*** As the names "inlet" and "outlet" suggest, the choice between these two patch types should be mede with consideration of the anticipated direction of flow.*

Besides the specific type, every patch has its own reference frame assigned, which is controlled by the column labeled "frame". A common approach is to assign the same frame (assigned to the whole component) to all patches in the component. But it is also possible to set the different frame, than the one the component has. This approach allows to have some rotating patches in non-rotating component.

If mesh is to be created (not just converted from external) and Meshing engine is SnappyHexMesh, further three columns ("min ref", "max ref", "layers") are available. These columns contain the minimal and maximal cell refinement and number of layers around the patch. The SnappyHexMesh mesher will subdivide the cells of the background mesh a few times. The count of these subdivisions is always in the given interval ⟨min ref, max ref⟩ and depends on the vicinity of other patches and the local curvature.

If user double-clicks on the patch name in the first column, the patch is highlighted in the *RenderView*, so user clearly see, which patch he/she is currently manipulating with (figure 7.12). The highlighted patch is also marked in the table with the flashlight "turned on", other patches have flashlight "turned off" (as seen in Fig. 7.6). If one double-clicks second time on that previously selected row, the higilighting will be disabled.

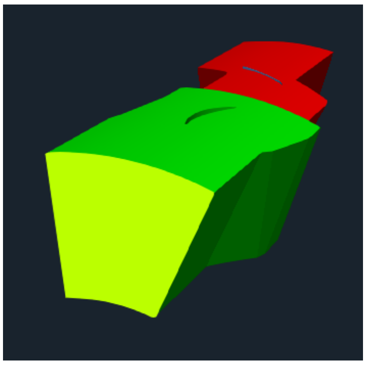


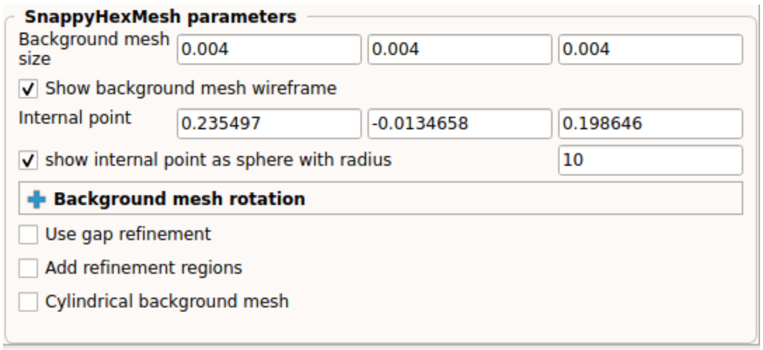
Figure 7.12: TMESH – highlighting of the outlet patch in the *RenderView*.

SnappyHexMesh parameters

This section is present, only if external mesh is not employed and Meshing engine is SnappyHexMesh. It contains basic parameters, that are neccesary for meshing using SnappyHexMesh (figure 7.13). The parameters are:

- "Background mesh cell size" is the most important parameter, and sets the basic cell size in the three directions for the rectangular background mesh (all cells will be such or smaller). Background mesh can be displayed in the *RenderView* by the checkbox "Show background mesh wireframe" (see figure 7.14). If the background mesh is cylindrical, see below, the meaning of the Background mesh cell size entries is different. The first number controls the edge length of cells comprising the inner rectangular prism, except for edges parallel with the axis, whose size is controlled by the third number. The second number controls the size of radially aligned edges in the rest of the cylinder (see 9.6).

- "Use cube cell" switches the background cells to cubes, so user needs to enter just one size



SnappyHexMesh parameters

Background mesh size: 0.004 | 0.004 | 0.004

☑ Show background mesh wireframe

Internal point: 0.235497 | -0.0134658 | 0.198646

☑ show internal point as sphere with radius: 10

□ Use gap refinement

□ Add refinement regions

□ Cylindrical background mesh

Figure 7.13: TMESH – SnappyHexMesh parameters.

(just for rectangular, not for cylindrical mesh). Instead of three input boxes only one is therefore shown.

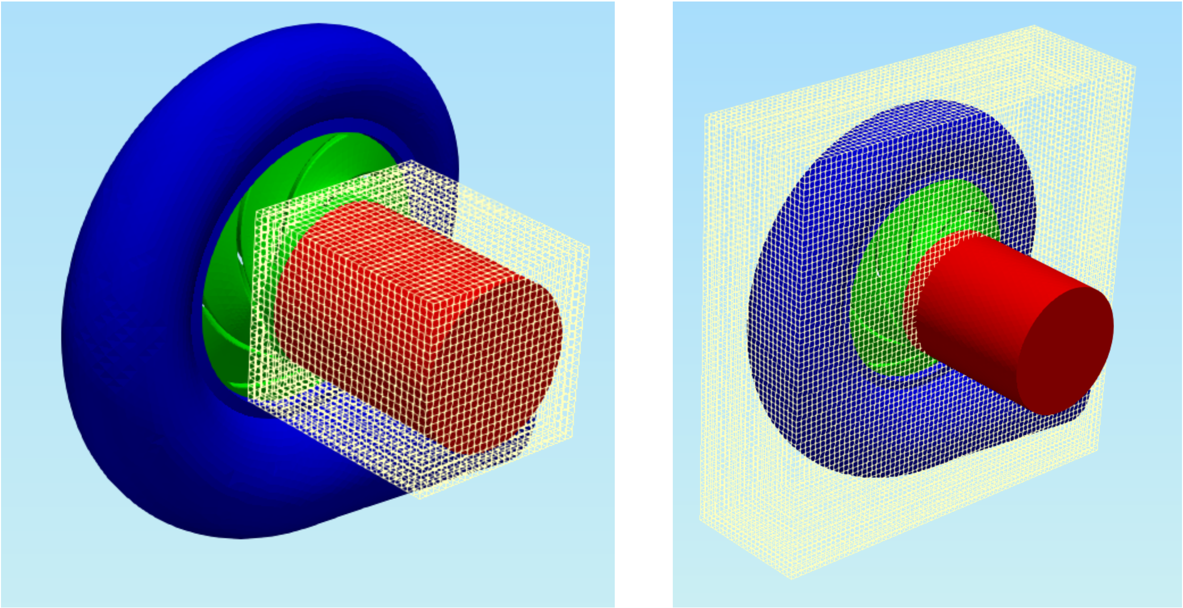
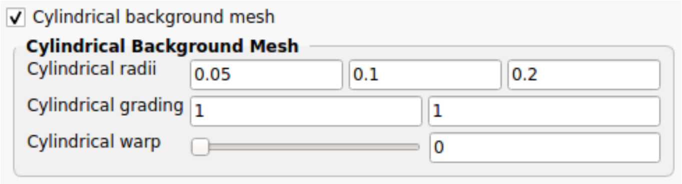


Figure 7.14: TMESH – background mesh wireframe in *RenderView*, shown for two components.

- "Internal point", which specifies arbitrary internal point. This is needed by the mesher. Internal point can be displayed in the *RenderView* by the checkbox "Show internal point as sphere with radius" (you can specify the point size for better visibility). When shown, it can be directly manipulated by mouse.
- "Cylindrical background mesh" requests the use of a cylindrical background mesh instead of the rectangular, see Figure 7.15, and enables the following parameters.
  - "Cylindrical radii", which sets the parameters r_0, r_1 and r_2 of the cylindrical mesh,
  - "Cylindrical grading", which sets the parameters g_1, and g_2 of the cylindrical mesh and

- "Cylindrical warp", which sets the parameter w of the cylindrical mesh.

The meaning of these parameters is explained in the section 9.6.



☑ Cylindrical background mesh

Cylindrical Background Mesh

Cylindrical radii: 0.05, 0.1, 0.2

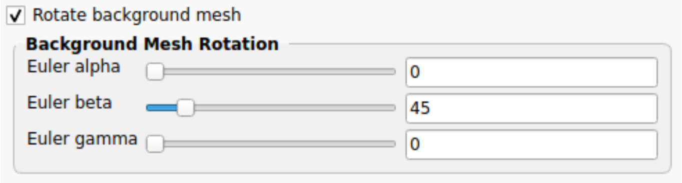
Cylindrical grading: 1, 1

Cylindrical warp: 0

Figure 7.15: TMESH – cylindrical background mesh.

If the cylindrical background mesh is selected, then the interpretation of "Background mesh cell size" changes. Instead of cell sizes in the x, y and z axes it sets the approximate cell sizes in radial, circumferential and axial directions (with respect to the chosen axis).

- "Rotate background mesh" enables the rotation of the background mesh.
- The submenu "Background mesh rotation" appears, if the switch Rotate background mesh is enabled (Fig. 7.16. Then three items "Euler alpha/beta/gamma" are available. These set the rotation of the background mesh, which might be useful to achieve better alignment with the input geometry. Eulers angles intrinsic (z-y'-z") rotation convention is used.



☑ Rotate background mesh

Background Mesh Rotation

Euler alpha: 0

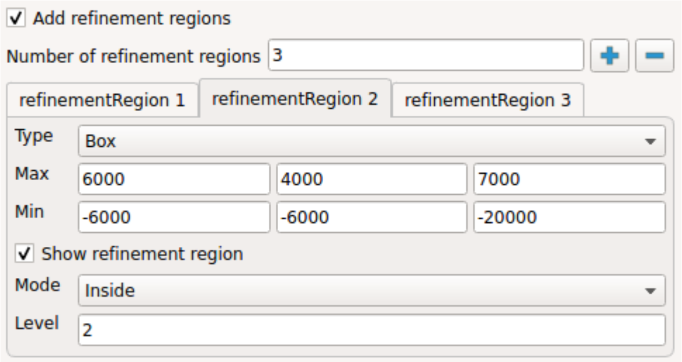
Euler beta: 45

Euler gamma: 0

Figure 7.16: TMESH – background mesh rotation.

- "Use gap refinement" checkbox enables setting higher level of refinement then max ref value for meshing small gaps between different parts of the input geometry. If enabled, a new column "gap ref" in the table Patches is shown, which controls the level of refinement. The minimum value is equal to max ref value. Note, that is the gap is situated between two individual patches, gap ref has to be active at both of them.
- "Add refinement regions" enables defining additional refinement regions (Fig. 7.17):
  - "Type" – shape of the refinement regions ("box", "cylinder" or "sphere").
  - "Mode" – if the mesh refinement is applied inside or outside of the refinement region.
  - "Level" – level of refinement to be applied

  For better imagination, the currently active refinement region can be shown in the *RenderView* as transparent object by clicking the "Show refinement region" button, so it is possible to position the region in the correct place (see Fig. 7.18).



☑ Add refinement regions

Number of refinement regions 3

Type Box

Max 6000 4000 7000

Min -6000 -6000 -20000

☑ Show refinement region

Mode Inside

Level 2

Figure 7.17: TMESH – refinement regions.

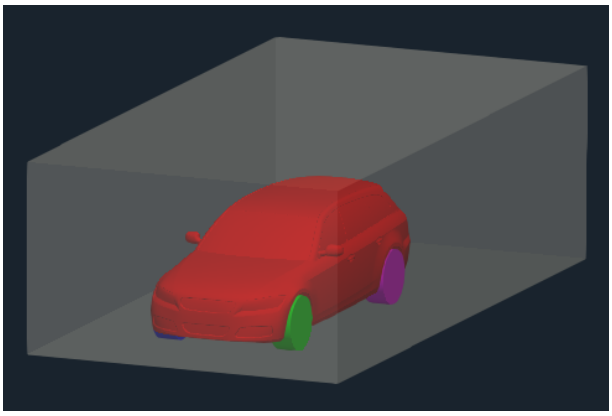
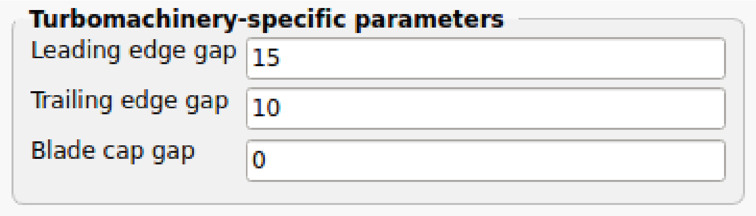


Figure 7.18: TMESH – refinement region representation.

- If the aforementioned switch Turbomachinery case is enabled, subsection "Turbomachinery-specific parameters" is displayed (figure 7.19). It contains options that are useful for meshing of the turbomachinery geometries:
  - "Leading edge gap" is an optional parameter specifying distance between the leading edge of the blade and the adjacent interface (if any). When given, then the refinement level for leading edge STL surfaces will be calculated such that the gap contains at least 8 cells across.
  - "Trailing edge gap" is an optional parameter specifying distance between the trailing edge of the blade and the adjacent interface (if any). When given, then the refinement level for trailing edge STL surfaces will be calculated such that the gap contains at least 8 cells across.
  - "Blade cap gap", commonly called "clearance" is the distance between the blade cap and the shroud. When given, then the refinement level for blade cap STL surfaces will be calculated such that the gap contains at least 8 cells across.



Turbomachinery-specific parameters

Leading edge gap 15

Trailing edge gap 10

Blade cap gap 0

Figure 7.19: TMESH – Turbomachinery-specific parameters.

7.2.3 Components map

In the "Components map" section (shown in Figure 7.20) all components and their interfaces are displayed. Thick arrows point always from inlet to outlet interface. If some interface is not available (or connected), the arrows become dashed and point elsewhere. This then indicates an invalid topology. All components must be connected into a single domain using the inlet- + outlet- / freestream- + freestream- -Interface pairs. The graph can be detached from the *Properties Panel* by a double-click; this transfers it to a new window. The separated window stays by default on top of all other windows (this can be manually unselected in the window manager menu). Further double-click merges the window back into the panel, as does also closing the window in any other way. The colours in the graph correspond to colours of the individual components in *RenderView*, assuming the colouring by *Component Colors* is chosen. The graph also displays all inlets and outlets, with inlet and outlet components assigned to them.

7.2.4 Advanced meshing options

This panel contains following sections, that include many parameters for fine tuning of the mesher SnappyHexMesh, so it is only visible if the STLs are going to be meshed. These options are set separately for each component.

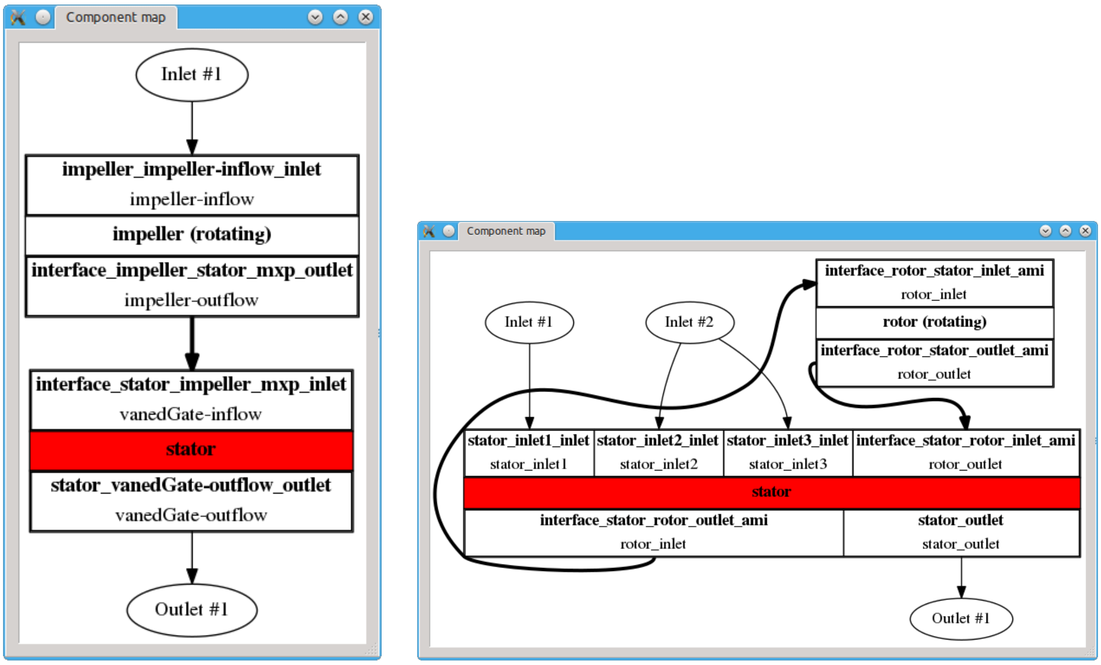


Figure 7.20: TMESH – two examples of the (detached) Components map. *Left:* Machine has linear topology with single inlet and single outlet. The fluid enters impeller component, then goes to stator component, where it leaves. *Right:* Machine has cyclic topology and multiple inlets. Here, the fluid enters stator component via one of the three inlets. The stator component can be left through a single outlet, or the fluid can flow through the rotor part first. The rotor can be skipped e.g. due to a leakage – fluid that goes through the leakage goes directly from inlet to outlet, without ever visiting rotor.

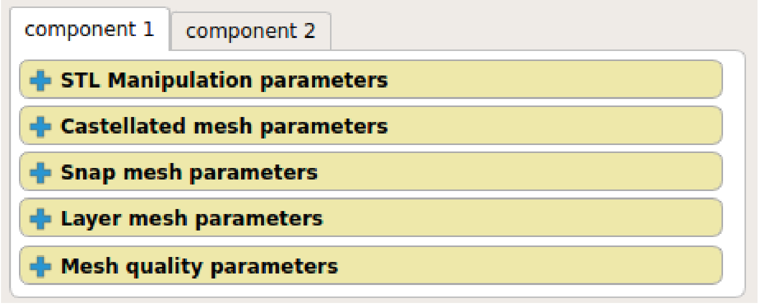


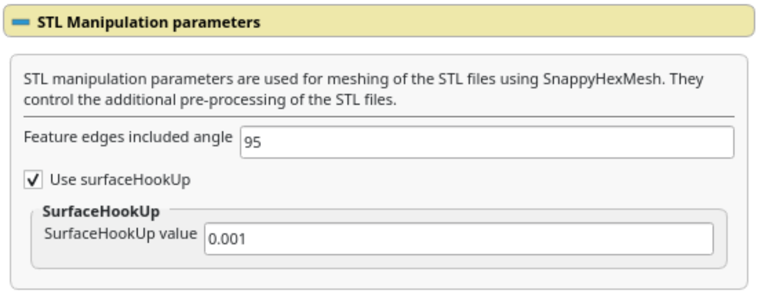
Figure 7.21: TMESH – Advanced meshing options section.

STL Manipulation parameters

This subsection contains two parameters:

- The entry "Feature edges included angle" is a tuning parameter that specifies maximal angle (in degrees) that is considered "sharp" by the mesher. When two faces of a boundary geometry make an angle smaller or equal to this number, then their common edge will be preserved in mesh (the cells' edges will be aligned with this line), otherwise it may be smoothed away. If zero is given, only open edges of the boundary geometry will be preserved. This parameter is only used by components that are meshed from STL files.
- "Use surfaceHookUp", which enables healing non-water-proof STL boundaries.
  - "SurfaceHookUp value" sets a maximum size of "holes" to be healed.

This subsection is displayed in Figure 7.22.


STL Manipulation parameters

STL manipulation parameters are used for meshing of the STL files using SnappyHexMesh. They control the additional pre-processing of the STL files.

Feature edges included angle 95

☑ Use surfaceHookUp

SurfaceHookUp

SurfaceHookUp value 0.001

Figure 7.22: TMESH – STL Manipulation parameters.

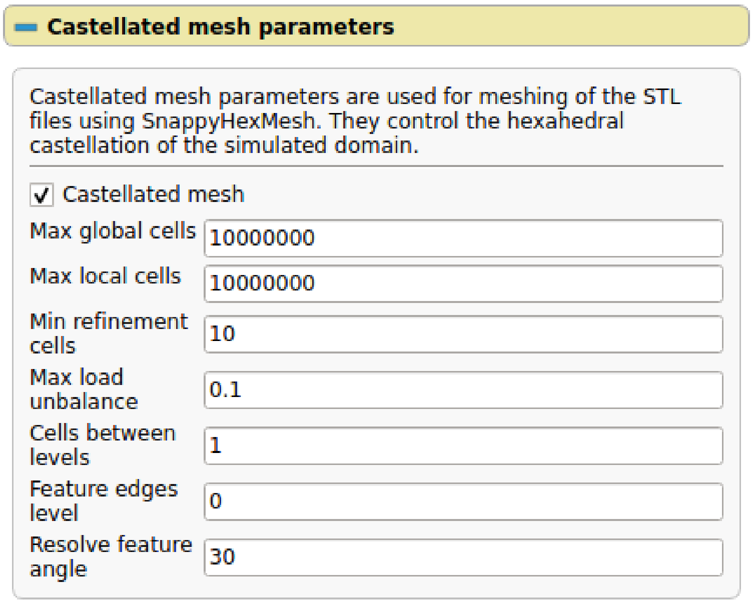
Castellated mesh parameters

Following parameters are available:

- The switch "Castellated mesh" enables and disables the *castellated mesh* phase. For normal operation it is always enabled.
- The value "Max global cells" sets a hard limit on the total number of cells to prevent memory overflow.
- The value "Max local cells" sets a soft limit on the number of cells per meshing process. When reached, the redistribution of the cells between the processes is done in a more careful (and slower) way.
- The value "Min refinement" sets a limit on the lowest number of cells refined in previous iteration of the mesher, which still triggers another iteration. If zero is given, the mesh will be refined ideally according to the algorithm of the mesher. Slightly higher values speed up the *castellated mesh* phase of the meshing without great impact on the quality of the mesh.

- The value "Max load unbalance" is the largest relative difference in number of cells across the mesher's processes, which is considered low and does not trigger (slow) redistribution.
- The value "Cells between levels" sets the minimal number of consecutive cells of a single refinement level in area where the refinement level dramatically changes.
- The number "Resolve feature angle" is an angle in degrees.

This subsection is displayed in Figure 7.23.



Castellated mesh parameters are used for meshing of the STL files using SnappyHexMesh. They control the hexahedral castellation of the simulated domain.

☑ Castellated mesh

Max global cells	10000000
Max local cells	10000000
Min refinement cells	10
Max load unbalance	0.1
Cells between levels	1
Feature edges level	0
Resolve feature angle	30

Figure 7.23: TMESH – Castellated mesh parameters.

Snap mesh parameters

This subsection is displayed in Figure 7.24.

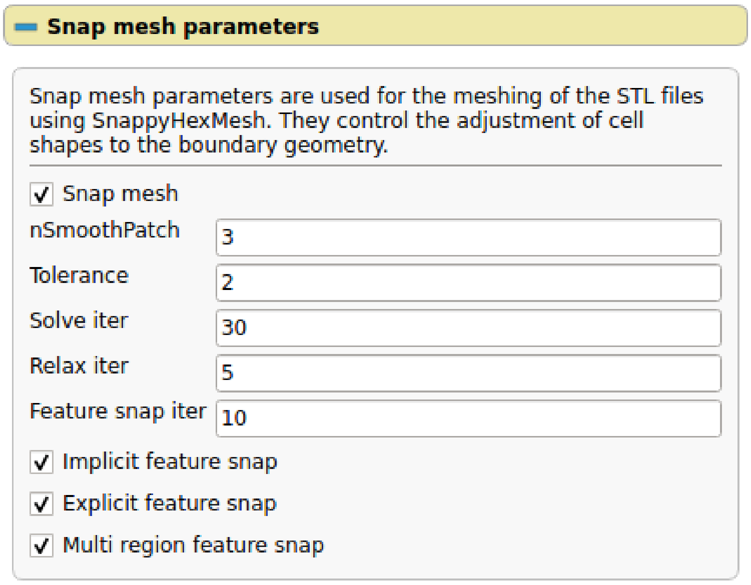
Layer mesh parameters

This subsection is displayed in Figure 7.25.

The switch "Add layers" enables and disables the *layer mesh* phase, where a boundary layer is added to the walls. The other parameters can be used to tune the mesher operation during this phase.

Mesh quality parameters

This subsection is displayed in Figure 7.26.



Snap mesh parameters

Snap mesh parameters are used for the meshing of the STL files using SnappyHexMesh. They control the adjustment of cell shapes to the boundary geometry.

☑ Snap mesh

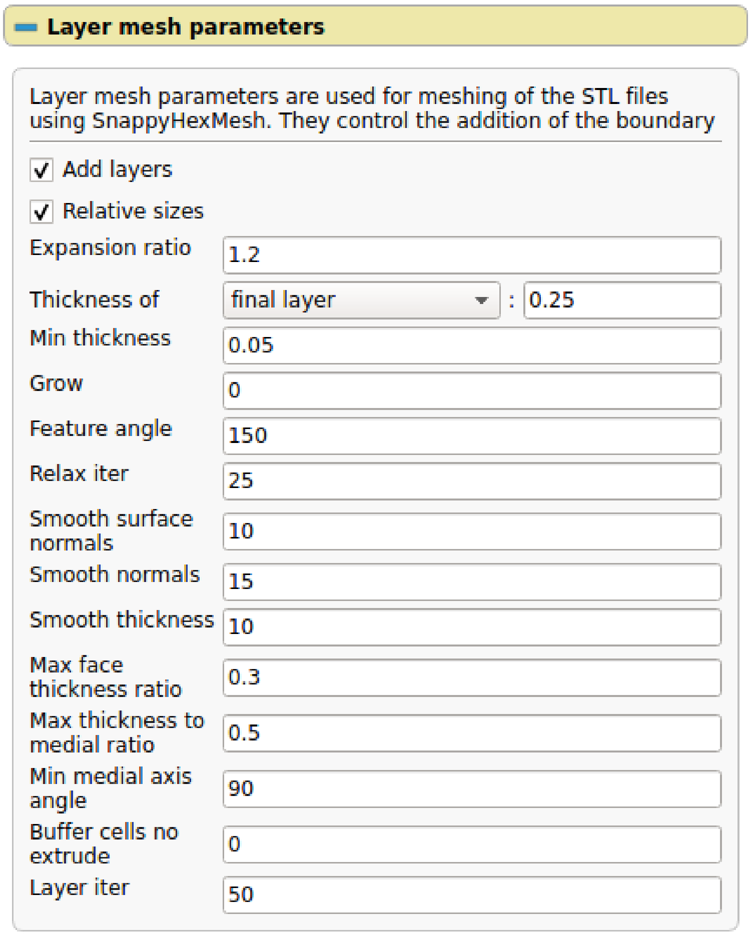
nSmoothPatch	3
Tolerance	2
Solve iter	30
Relax iter	5
Feature snap iter	10

☑ Implicit feature snap

☑ Explicit feature snap

☑ Multi region feature snap

Figure 7.24: TMESH – Snap mesh parameters.



Layer mesh parameters

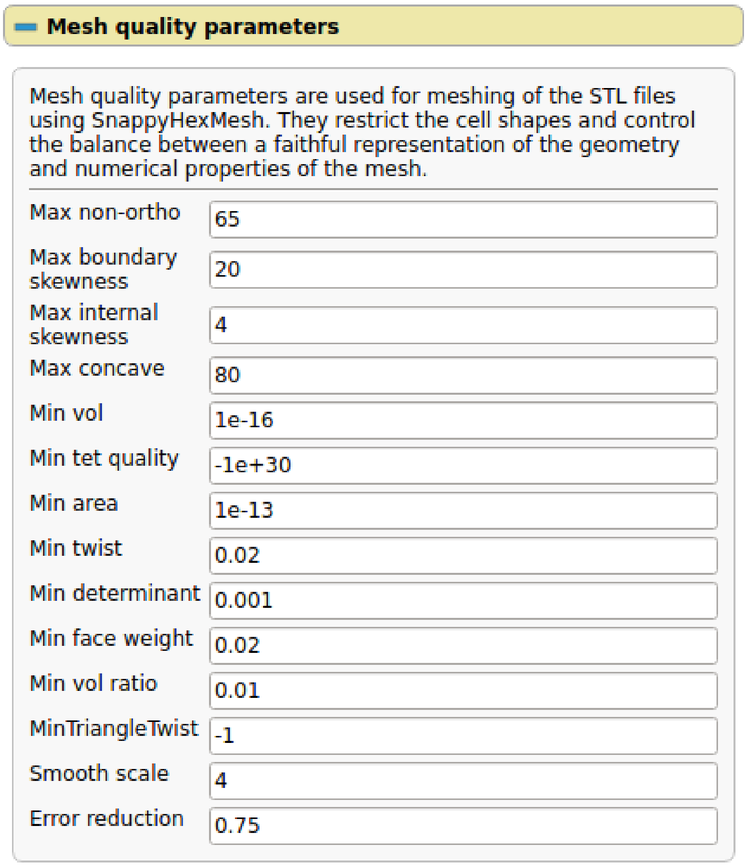
Layer mesh parameters are used for meshing of the STL files using SnappyHexMesh. They control the addition of the boundary

☑ Add layers

☑ Relative sizes

Expansion ratio	1.2
Thickness of	final layer : 0.25
Min thickness	0.05
Grow	0
Feature angle	150
Relax iter	25
Smooth surface normals	10
Smooth normals	15
Smooth thickness	10
Max face thickness ratio	0.3
Max thickness to medial ratio	0.5
Min medial axis angle	90
Buffer cells no extrude	0
Layer iter	50

Figure 7.25: TMESH – Layer mesh parameters.



Mesh quality parameters

Mesh quality parameters are used for meshing of the STL files using SnappyHexMesh. They restrict the cell shapes and control the balance between a faithful representation of the geometry and numerical properties of the mesh.

Max non-ortho	65
Max boundary skewness	20
Max internal skewness	4
Max concave	80
Min vol	1e-16
Min tet quality	-1e+30
Min area	1e-13
Min twist	0.02
Min determinant	0.001
Min face weight	0.02
Min vol ratio	0.01
MinTriangleTwist	-1
Smooth scale	4
Error reduction	0.75

Figure 7.26: TMESH – Mesh quality parameters.

7.2.5 Parallel run

This section dedicated to the meshing in parallel, which is currently supported only by the SnappyHexMesh mesher for CFD meshes, so this section is not visible, if the Mesh Output is FEA Mesh (Fig. 7.27).

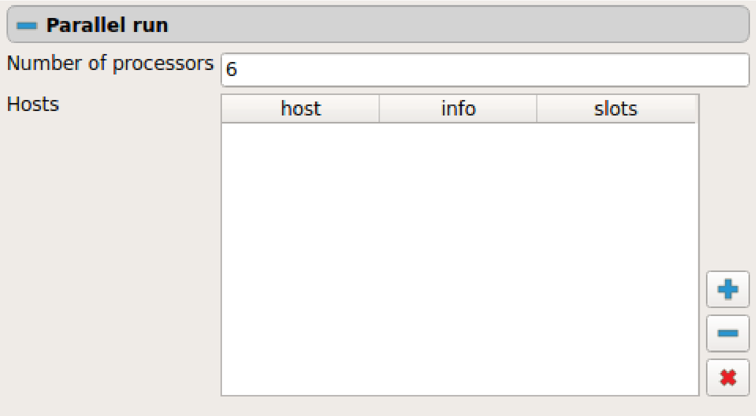


Figure 7.27: TMESH – Parallel run.

- The entry "Number of Processors" specifies number of processes used to run the SnappyHexMesh mesher. TCAE (more precisely OpenFOAM running in background) uses MPI (Message Passing Interface) for communication of the processes.
- The table "Hosts" (advanced) can be used in conjunction with the Number of Processors entry to schedule the running of the parallel jobs. This table contains nodes, on which the parallel processes will be launched. The total number of processes given by Number of Processors is evenly divided between the nodes. A new node is added by the plus button. This is by default the "localhost" node, but the word "localhost" in the first column can be edited (double-click to enable editing) to any other host name or IP address. The simple local network information provided by the system utility "getent" is then shown in the second column. If the node is unknown, the table row will turn red. Such nodes must be deleted, or the execution will fail. There are several restrictions on the usage of remote nodes:
  - All nodes must be accessible from the workstation where TCAE runs without password for the current user (i.e. using the public key authentication), and the same must be true for access between the nodes.
  - The MPI and TCAE installations on the workstation and all the nodes must be identical. Ideally, there is just one installation on a shared network file system.
  - The TCAE case directory is written on a shared network file system, so that it is accessible both to the front-end workstation used for solution management via TCAE and to the remote calculation nodes.

  Hosts scheduling is currently possible only in Linux systems.

7.3 FEA MESH

This section includes all adjustments for the FEA mesh, so it is available only if the value of the Mesh output is either FEA Mesh or CFD & FEA Mesh. The meshing itself is run in the TCAEManager and it is then passed on to the TFEA module. The settings of this section are depicted in Fig. 7.28.

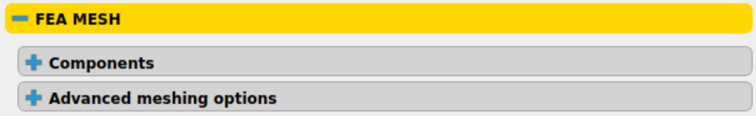


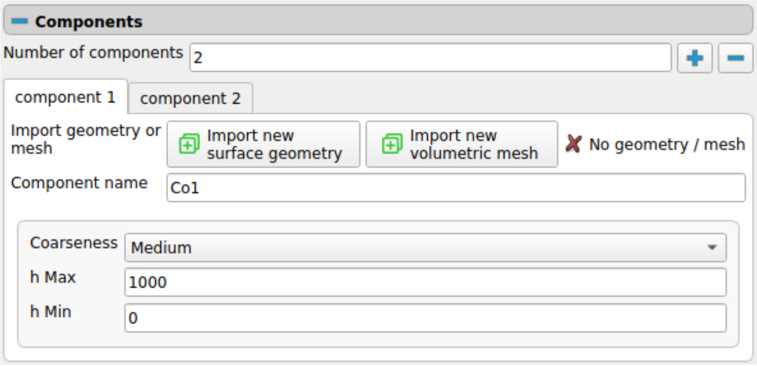
Figure 7.28: TMESH – FEA MESH section.

7.3.1 Components

The basic meshing options for FEA meshes of all the mesh components is set in this section. First, the number of components is chosen, then for each component the input geometry is selected and basic mesh coarseness options are set. Each component is meshed separately and independently. Therefore if two meshes have a common face the face is meshed twice, once per each component, and the component meshes does not coincide there.

After all components are meshed they are merged to a final mesh that is the input to the TFEA module. Note, even though everything is in a single mesh the original component can stay separated and independent. This depends on the setting of the TFEA module.

The layout of this section is shown in Fig. 7.29.



Number of components 2

Import geometry or mesh

Component name Co1

Coarseness Medium

h Max 1000

h Min 0

Figure 7.29: TMESH – components for FEA mesh.

Following parameters are present:

- "Number of components"
  The number of components for FEA mesh is chosen.

- "Component name"
  Each component needs to have its unique name, the default one is Co<number-of-component>.

- "Import geometry or mesh"
  The button "Import new surface geometry" opens file dialog where you navigate to the geometry file you want to use, the file is then imported to the project. Supported geometry formats are STL and STEP. The button "Import new volumetric mesh" opens file dialog where you navigate to the mesh file. Supported formats are .vol (NetGen), .mesh (gmsh) and .inp (Abaqus-like). Finaly, at the end of the row is marker indicating the state of imported geometry or mesh. This is visible if the checkbox "Use external geometries/meshes (files will not be imported into the project)" is not selected.

- "Mesh input"
  If the external geometries and meshes are used within the project, i.e., the checkbox "Use external geometries/meshes (files will not be imported into the project)" is selected, the user selects the mesh input here, this can be a geometry file "STL Geometry", "STEP Geometry" or an external mesh. The external mesh can be in INP (Abaqus) format, select "Abaqus Mesh", VOL (NetGen) format, in this case select "Netgen Mesh", or MSH (Gmsh) format, this is the "GMSH mesh" option.

- "STL file"
  Sets the path to the STL geometry file. Selectable only if "STL Geometry" is selected as a "Mesh input".

- "STEP file"
  Sets the path to the STEP geometry file. Selectable only if "STEP Geometry" is selected as a "Mesh input".

- "Netgen mesh file"
  Sets the path to the VOL mesh file. Selectable only if "Netgen Mesh" is selected as a "Mesh input".

- "Abaqus mesh file"
  Sets the path to the INP mesh file. Selectable only if "Abaqus Mesh" is selected as a "Mesh input".

- "Meshing engine"
  The engine behind the meshing process. Visible only if a geometry is used as input for this component. The choice is betvee "Netgen" and "gmsh".

Meshing parameters

If a new mesh is to be created, the meshing parameters are specified in this box.

- "Coarseness"
  There are five levels of the mesh coarseness the user can choose from. When an option is selected, the values in "Mesh size" box are set. These values can be further adjusted by the user. This option is only for "Netgen" meshing engine.

- "h Max"
  Sets the maximal size of an element in the units of the original (unscaled) geometry file.

- "h Min"
  Sets the minimal size of an element in the units of the original (unscaled) geometry file. Zero is also a valid input, in this case the element size is just not bounded from below.

7.3.2 Advanced meshing options

In this section the advanced user can fine-tune the mehing options for the mesh generator. The structure is like in Fig. 7.30 and the content of the subsections differ depending on the chosen meshing engine.

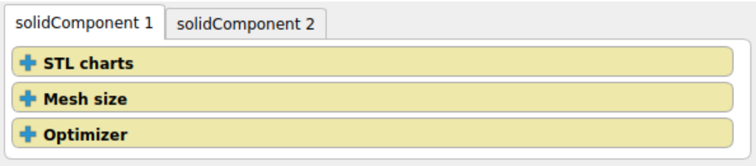


Figure 7.30: TMESH – advanced meshing options for FEA meshes.

Mesh size

First we discuss the NetGen parameter set for mesh size as depiceted in the Fig. 7.31. The values here are choosen by the "Coarseness" entry above, however, advanced user can edit these values by hand.

- "Grading"
  This line controls how rapidly the size of elements can change. The value needs to be between 0 and 1, the lower the value is the more the elements would like to have a similar size. Note, that one decimal place is enough.

- "Chart distance"
  Limit element size by distance to the neighboring chart. The higher the factor is, the more is the element size restricted.

- "Line length"
  Limit size of elements located near to chart boundary curves ends. The higher the factor is, the more is the element size restricted.

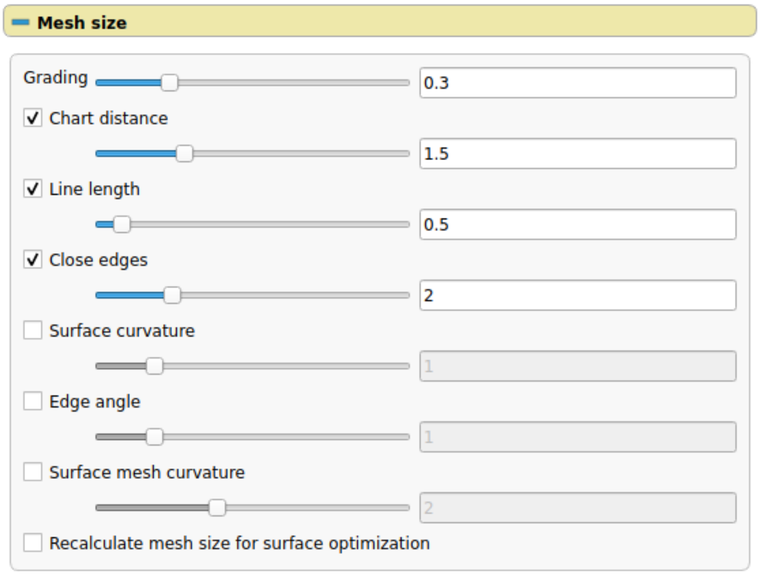
- "Close edges"
  Limit size of elements located near to chart boundary curves by distance to other doundary curve. The higher the factor is, the more is the element size restricted.

- "Surface curvature"
  Limit element size by surface curvature. The higher the factor is, the more is the element size restricted.

- "Edge angle"
  Limit element size by by chart boundary curvature. The higher the factor is, the more is the element size restricted.

- "Surface mesh curvature"
  Set number of elements per curvature radius.

- "Recalculate mesh size for surface optimization"
  Controls if the mesh size is calculated before surface optimization steps are executed.



Mesh size

Grading 0.3

☑ Chart distance 1.5

☑ Line length 0.5

☑ Close edges 2

□ Surface curvature 1

□ Edge angle 1

□ Surface mesh curvature 2

□ Recalculate mesh size for surface optimization

Figure 7.31: TMESH – NetGen mesh size options for FEA meshes.

The Gmsh set of parameters is depicted in the Fig. 7.32.

- "Surface meshing algorithm"
  Lets the user to choose surface meshing algorithm, the choices are "MeshAdapt", "Automatic", "InitialMesh", "Delaunay2D", "FrontalDelaunay", "BAMG", "FrontalDelaunayForQuads" and "PackingOfParallelograms". For explanation please consult the Gmsh manual [36] and references therein.

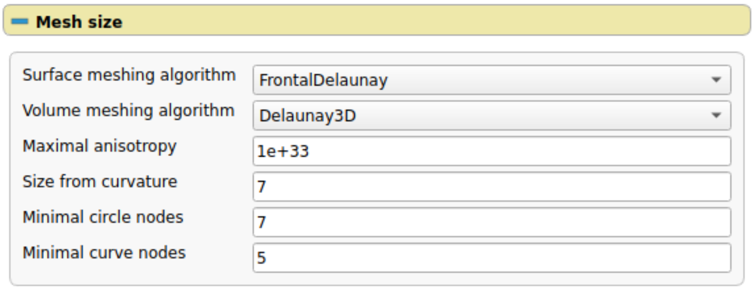
- "Volume meshing algorithm"
  Algorithm that will be used to advance on the surface mesh and create the volume mesh is to be selected here. The optins are "Delaunay3D", "InitialMeshOnly", "Frontal", "MMG3D", "RTree" and "HXT". For explanation please consult the Gmsh manual [36] and references therein.

- "Maximal anisotropy"
  Maximal anisotropy of the mesh.

- "Size from curvature"
  Number of elements per $2\pi r$, where r is the local curvature radius.

- "Minimal circle nodes"
  Minimal number of nodes used to mesh circle.

- "Minimal curve nodes"
  Minimal number of nodes used to mesh curves (other then lines, circles or elipses).



Mesh size

Surface meshing algorithm	FrontalDelaunay
Volume meshing algorithm	Delaunay3D
Maximal anisotropy	1e+33
Size from curvature	7
Minimal circle nodes	7
Minimal curve nodes	5

Figure 7.32: TMESH – Gmsh mesh size options for FEA meshes.

STL charts

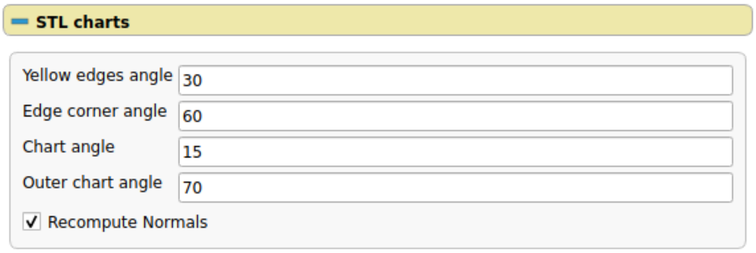
When creating mesh from a geometry described by an STL file the meshing engines need to create it's own description of the surface. This procedure can be controled by parameters from this section. Again, the sets of parameters a different from NetGen and Gmsh.

Let's start with the NetGen set as shown in Fig. 7.33.

- "Yellow edge angle"
  Minimum angle between normals of adjacent triangles at which the common edge is considered as a feature edge.
- "Edge corner angle"
  Minimum angle between adjacent edges of chart boundary, the chart is split for higher values.
- "Chart angle"
  Angle between normals of adjacent triangles under which the sahred edge is not considered as a chart boundary.
- "Outer chart angle"
  Angle to identify overlapping parts of chart.
- "Recompute Normals"
  This checkbox decides whether the normals of the STL triangles are recomputed before the meshing. If yes, the normals are obtained as cross products between two edges. This can help to mesh some geometries.

The parameter set for Gmsh engine can be seen in Fig. 7.34.

- "Use STL as surface mesh"
  If you like your STL that much you want to you it directly as the surface mesh you can select this option. In usual case you want to remesh the surface so check this only if you know what you are doing.



STL charts

Yellow edges angle 30

Edge corner angle 60

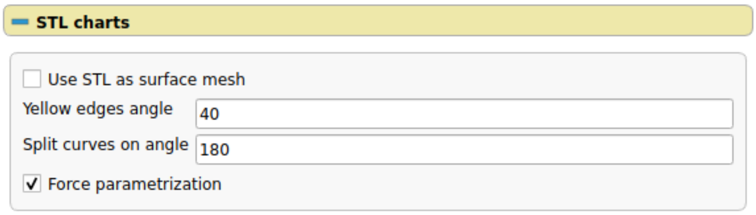
Chart angle 15

Outer chart angle 70

☑ Recompute Normals

Figure 7.33: TMESH – Netgen STL charts options for FEA meshes.

- "Yellow edge angle"
  Same as in the NetGen section.
- "Split curves on angle"
  The curves representing the surface will be split when they reach this angle.
- "Force parametrization"
  Gmsh will try to create parametric surface patches when this option is selected.



STL charts

☐ Use STL as surface mesh

Yellow edges angle 40

Split curves on angle 180

☑ Force parametrization

Figure 7.34: TMESH – Gmsh STL charts options for FEA meshes.

Optimizer

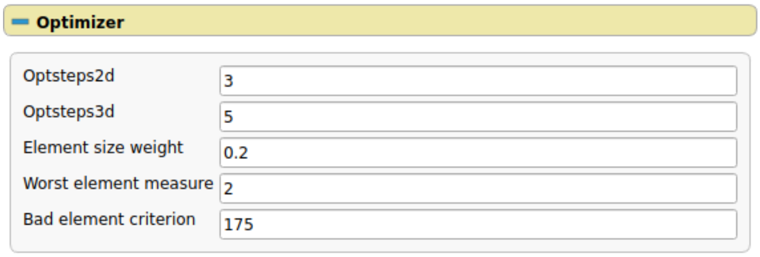
Once the surface or volume is meshed its quality can be further enhanced by performing smoothing optimization steps. They are controled by the these entries which again differ for NetGen and Gmsh.

For NetGen they look similar as on Figure 7.35.

- "Optsteps2d"
  Number of optimization steps for the 2D (surface) mesh. These steps are performed before the volume mesh is generated. (The surface mesh is generated first and, based on it, the volume meshing is done.)
- "Optsteps3d"
  Number of optimization steps for the 3D (volume) mesh. These steps are performed for the final volume mesh after the meshing. The purpose is to make the mesh cleaner and smoother (hence, having better numerical properties).
- "Element size weight"
  Weight of triangle size badness.

- "Worst element measure"
  Power of error used to approximate max error optimization.

- "Bad element criterion"
  Elements with faces with angle higher then this value are considered as bad.



Optsteps2d 3
Optsteps3d 5
Element size weight 0.2
Worst element measure 2
Bad element criterion 175

Figure 7.35: TMESH – NetGen optimizer options for FEA meshes.

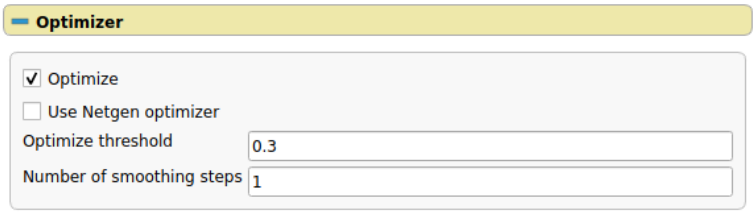
For Gmsh they are listed in following list and depicted in Figure 7.36.

- "Optimize"
  This checkbox controls whether to optimize the mesh or not.

- "Use Netgen optimizer"
  If this control is on Gmsh will use the NetGen optimizer for the mesh. Otherwise it will use its own default one.

- "Optimize threshold"
  Sets the lowest quality the elements can have. All lower quality elments are subjected to optimization.

- "Number of smoothing steps"
  Number of smoothing steps applied on the final mesh.



☑ Optimize
□ Use Netgen optimizer
Optimize threshold 0.3
Number of smoothing steps 1

Figure 7.36: TMESH – Gmsh optimizer options for FEA meshes.

Chapter 8

TMESH – Configuration File Options

This section contains the list of all keywords of the TMESH module, that might appear in the *.tcae file. General format of the *.tcae file is described in section 3.3.

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
meshOutput	CFD	–	yes
▶ *Which meshes to create. Possible values are CFD, FEA, CFDWithFEA*			
TMESH-numberOfProcessors	4	–	no
▶ *Number of CPU cores used for simulation. Default = 1. If greater than 1, then a domain-decomposition solution with MPI communication is used.*			
TMESH-hosts	node1 node2 node3	–	no
▶ *List of remote machines for scheduling parallel processes. Paswordless login must be available. Currently active only in Linux.*			
userDefinedFunctions-TMESH	scripts/run.py afterWrite	–	no
▶ *Custom user script setup.*			
scaleFactor	1	m	yes
▶ *Scale factor for STL files and various other metric properties. Selectable unit.*			
numberOfReferenceFrames	2	–	no
▶ *How many reference frames there is in total.*			
N_referenceFrame-name	static	–	no
▶ *Definition of Nth reference frame, name.*			
N_referenceFrame-axis	0 0 1	–	no
▶ *Definition of Nth reference frame, directional vector of the axis*			
N_referenceFrame-origin	0 0 0	–	no
▶ *Definition of Nth reference frame, coordinate of origin.*			
numberOfComponents	3	–	yes
▶ *Number of components.*			
boundingBox	yes	–	no
▶ *Will switch on the bounding box feature.*			
boundingBox-point1	-10 0 0	–	no
▶ *Bounds of the bounding box in x,y and z direction respectively.*			
boundingBox-point2	10 10 10	–	no
▶ *The other bound of the bounding box in x,y and z direction respectively.*			
boundingBox-flowDirection	plusX	–	no

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
▶ *The direction of flow in bounding box*			
N_meshInput	STLGeometry	–	yes
▶ *Format of the geometry or other mesh, from which the TMESH creates the mesh for Nth component. Possible options are: STLGeometry, MultiSolidSTLGeometry, FluentMesh, OpenFOAMMesh, CGNSMesh*			
N_geometryPath	./STL	–	yes*
▶ *Directory with STL files or path to a multi-solid STL of the Nth component. Can be both relative (w.r.t. TCAE file) and absolute.*			
N_meshingEngine	snappyHexMesh	–	no
▶ *Which meshing system to use for creating the mesh for the Nth component. Possible options are: snappyHexMesh*			
N_externalMeshRegion	mesh-draft	–	no
▶ *For use with external mesh, which region to use.*			
N_componentName	rotor	–	no
▶ *Custom name for the Nth component (used in patch names)*			
N_component-referenceFrame	1	–	no
▶ *To what reference frame the Nth component belongs.*			
N_numberOfPeriodicSegments	6	–	no
▶ *Periodic multiplier, number of segments of Nth component.*			
N_internalPoint	0 0 -200	scaleFactor	yes*
▶ *Point inside the Nth component (affected by scale factor). Defines the interior, where the fluid will simulated. Used only when meshing.*			
N_backgroundMeshSize	4.0 4.0 4.0	scaleFactor	yes*
▶ *Mesh size in each direction (affected by scale factor) for the Nth component. Used only when meshing.*			
N_meshRotation	0 45 0	–	no
▶ *Rotation of background mesh of Nth component around x, y and z axes respectively.*			
N_useCylindricalBackgroundMesh	true | false	–	no
▶ *Background mesh can be cylindrical or Cartesian, default=false. (Nth component)*			
N_cylindricalBackgroundMeshGrading	1 1	–	no
▶ *Cylindrical mesh gradients g1 g2. (Nth component)*			
N_cylindricalBackgroundMeshRadii	0.05 0.1 0.2	scaleFactor	no

Keyword ▶ ***Description***	Allowed / sample values	Units	Mandatory
▶ *Cylindrical mesh radiuses r0 r1 r2 default: r2/4 r2/2 maxR*1.01. (Nth component)*			
N_cylindricalBackgroundMeshWarp	0	–	no
▶ *Cylindrical mesh warp, default: 0. (Nth component)*			
N_useRefinementRegions	yes	–	no
▶ *Switch on use of refinement regions in the Nth component*			
N_numberOfRefinementRegions	1	–	no
▶ *Number of the refinement regions in the Nth component.*			
N_refinementRegion-M_type	box | sphere | cylinder	–	no
▶ *Type of the Mth refinement region in the Nth component.*			
N_refinementRegion-M_box-min	0.5 1.2 -0.8	scaleFactor	no
▶ *Nth component, Mth refinement region, type box, coordinates of one vertex of the box.*			
N_refinementRegion-M_box-max	1.8 2.5 -0.2	scaleFactor	no
▶ *Nth component, Mth refinement region, type box, coordinates of the opposite vertex of the box.*			
N_refinementRegion-M_sphere-centre	0 0 7.5	scaleFactor	no
▶ *Nth component, Mth refinement region, type sphere, coordinates of the sphere centre.*			
N_refinementRegion-M_sphere-radius	1.5	scaleFactor	no
▶ *Nth component, Mth refinement region, type sphere, radius of the sphere.*			
N_refinementRegion-M_cylinder-point1	-1.5 1 1	scaleFactor	no
▶ *Nth component, Mth refinement region, type cylinder, centre coordinates of the one cylinder base.*			
N_refinementRegion-M_cylinder-point2	3 1 1	scaleFactor	no
▶ *Nth component, Mth refinement region, type cylinder, centre coordinates of the other cylinder base.*			
N_refinementRegion-M_cylinder-radius	1.5	scaleFactor	no
▶ *Nth component, Mth refinement region, type cylinder, radius of the cylinder.*			
N_refinementRegion-M_mode	inside | outside	–	no
▶ *Refinement mode in the Mth refinement region in the Nth component.*			
N_refinementRegion-M_level	3	–	no
▶ *Refinement level in the Mth refinement region in the Nth component.*			
N_featureEdgesIncludedAngle	0	deg	no

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
▶ *Angle used to extract STL edges of the Nth component, 0 = only boundaries.*			
N_surfaceHookUp	true | false	–	no
▶ *Hook-up the STL files of the Nth component, default = true.*			
N_bladeCap-clearance	0.5	scaleFactor	no
▶ *Distance between bladeCap and shroud for unshrouded impelers for the Nth component.*			
N_bladeTrailingEdge-toInterfaceDistance	1	scaleFactor	no
▶ *Distance between trailingEdge and outlet interface for the Nth component.*			
N_bladeLeadingEdge-toInterfaceDistance	1.5	scaleFactor	no
▶ *Distance between leadingEdge and inlet interface for the Nth component.*			
N_castellatedMesh	true	–	no
▶ *Whether to generate castellated mesh during meshing of the Nth component.*			
N_castellate-maxGlobalCells	10000000	–	no
▶ *Maximal global number of cells during meshing (Nth component).*			
N_castellate-maxLocalCells	10000000	–	no
▶ *Maximal number of cells per MPI process during meshing (Nth component).*			
N_castellate-minRefinementCells	10	–	no
▶ *Minimal number of refined cells after a refinement iteration for termination of refinement iterations (Nth component).*			
N_castellate-maxLoadUnbalance	0.1	–	no
▶ *Maximal relative difference between individual processes' cell count not triggering redistribution of the mesh (Nth component).*			
N_castellate-nCellsBetweenLevels	1	–	no
▶ *Number of cell transition layers between cells of different refinement level (1 = no transition) (Nth component).*			
N_castellate-resolveFeatureAngle	30	deg	no
▶ *Maximal feature angle that has influence on refinement (Nth component).*			
N_castellate-featureEdgesLevel	0	–	no
▶ *Level of refinement of the castellated mesh along the feature edges (Nth component).*			
N_snap	true	–	no
▶ *Whether to snap the castellated mesh during meshing of the Nth component.*			

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
N_snap-nSmoothPatch	3	–	no
▶ *Number of patch smoothing iterations before finding correspondence to surface (Nth component).*			
N_snap-tolerance	2	–	no
▶ *Maximum relative distance for points to be attracted by surface (Nth component).*			
N_snap-nSolveIter	30	–	no
▶ *Number of mesh displacement relaxation iterations (Nth component).*			
N_snap-nRelaxIter	5	–	no
▶ *Maximum number of snapping relaxation iterations (Nth component).*			
N_snap-nFeatureSnapIter	10	–	no
▶ *Number of feature edge snapping iterations (Nth component).*			
N_snap-implicitFeatureSnap	true	–	no
▶ *Detect (geometric only) features by sampling the surface (Nth component).*			
N_snap-explicitFeatureSnap	true	–	no
▶ *Take into consideration manually generated feature edges (Nth component).*			
N_snap-multiRegionFeatureSnap	true	–	no
▶ *Detect features between multiple surfaces (Nth component).*			
N_addLayers	false	–	no
▶ *Whether to add layers to the snapped mesh during meshing of the Nth component.*			
N_layers-relativeSizes	true	–	no
▶ *Relative or absolute layer thickness (Nth component).*			
N_layers-expansionRatio	1.2	–	no
▶ *Expansion factor for layer mesh (Nth component).*			
N_layers-finalLayerThickness	0.25	– (m)	no
▶ *Wanted thickness of the layer furthest away from the wall (Nth component).*			
N_layers-minThickness	0.05	– (m)	no
▶ *Minimum overall thickness of total layers (Nth component).*			
N_layers-nGrow	0	–	no
▶ *If points get not extruded do nGrow layers of connected faces that also not grown (Nth component).*			

Keyword	Allowed / sample values	Units	Mandatory
▶ ***Description***			
N_featureAngle	90	deg	no
▶ *When not to extrude surface (Nth component).*			
N_layers-nRelaxIter	25	–	no
▶ *Max number of iterations after which relaxed meshQuality controls get used (Nth component).*			
N_layers-nSmoothSurfaceNormals	10	–	no
▶ *Number of smoothing iterations of surface normals (Nth component).*			
N_layers-nSmoothNormals	15	–	no
▶ *Number of smoothing iterations of interior mesh movement direction (Nth component).*			
N_layers-nSmoothThickness	10	–	no
▶ *Smooth layer thickness over surface patches (Nth component).*			
N_layers-maxFaceThicknessRatio	0.3	–	no
▶ *Stop layer growth on highly warped cells (Nth component).*			
N_layers-maxThicknessToMedialRatio	0.5	–	no
▶ *Reduce layer growth where ratio thickness to medial distance is large (Nth component).*			
N_layers-minMedialAxisAngle	90	–	no
▶ *Angle used to pick up medial axis points (Nth component).*			
N_layers-nBufferCellsNoExtrude	0	–	no
▶ *Create buffer region for new layer terminations (Nth component).*			
N_layers-nLayerIter	50	–	no
▶ *Overall max number of layer addition iterations (Nth component).*			
N_quality-maxNonOrtho	65	deg	no
▶ *Maximum non-orthogonality allowed. Set to 180 to disable (Nth component).*			
N_quality-maxBoundarySkewness	20	–	no
▶ *Max boundary skewness allowed (Nth component).*			
N_quality-maxInternalSkewness	4	–	no
▶ *Max internal skewness allowed (Nth component).*			
N_quality-maxConcave	80	–	no
▶ *Max concaveness allowed (Nth component).*			

Keyword ▶ ***Description***	Allowed / sample values	Units	Mandatory
N_quality-minVol	1E-16	m^3	no
▶ *Minimum pyramid volume. Is absolute volume of cell pyramid. Set to a sensible fraction of the smallest cell volume expected (Nth component).*			
N_quality-minTetQuality	-1E+30	–	no
▶ *Minimum quality of the tet formed by the face-centre and variable base point minimum decomposition triangles and the cell centre (Nth component).*			
N_quality-minArea	1E-13	m^2	no
▶ *Minimum face area (Nth component).*			
N_quality-minTwist	0.02	–	no
▶ *Minimum face twist (Nth component).*			
N_quality-minDeterminant	0.001	–	no
▶ *Minimum normalised cell determinant. This is the determinant of all the areas of internal faces. It is a measure of how much of the outside area of the cell is to other cells (Nth component).*			
N_quality-minFaceWeight	0.02	–	no
▶ *Relative position of face in relation to cell centres (from 0 to 0.5). Orthogonal mesh corresponds to 0.05 (Nth component).*			
N_quality-minVolRatio	0.01	–	no
▶ *Volume ratio of neighbouring cells (from 0 to 1) (Nth component).*			
N_quality-minTriangleTwist	-1	–	no
▶ *Per triangle normal compared to average normal (Nth component).*			
N_quality-nSmoothScale	4	–	no
▶ *Number of error distribution iterations (Nth component).*			
N_quality-errorReduction	0.75	–	no
▶ *Amount to scale back displacement at error points (Nth component).*			
N_patchName-numberOfMixingPlanes	0	–	no
▶ *Number of averaging planes, default=1 – if 0 then cyclicAMI is used.*			
N_defaultRefinement	1 4	–	no
▶ *Default minimal and maximal refinement for all surfaces (Nth component).*			
N_defaultWallLayers	3	–	no
▶ *Default number of suface layers for a wall (Nth component).*			

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
N_patchName-refinementSurfaces	1 4	—	no
▶ *Minimal and maximal refinement on patchName in the Nth component.*			
N_patchName-layers	3	—	no
▶ *Number of boundary layers on patchName in the Nth component. Only effective when addLayers is on.*			
N_patchName-gapRefinementLevel	3	—	no
▶ *Increased maximal refinement on patchName in the Nth component for meshing small gaps between parts of the input geometry.*			
numberOfSolidComponentS	1	—	yes
▶ *Number of components of solid mesh.*			
TMESH-N_solid-geometryType	surfaceGeometry	—	yes
▶ *Type of the geometry for N-th solid component - surface geometry or volumetric mesh.*			
TMESH-N_solid-importedSurfaceGeometryFormat	STLGeometry	—	yes
▶ *Format of surface file - STEP geometry or STL geometry.*			
N_solid-meshInput	STLGeometry	—	no
▶ *Type of external mesh input.*			
N_solid-meshingEngine	netgen	—	no
▶ *Tool for mesh generation.*			
N_solid-externalMeshPath	solid-mesh.inp	—	no
▶ *Path to the external mesh.*			
N_solid-geometryPath	solid.stl	—	no
▶ *Path to the external geometry file.*			
N_solid-name	Component1	—	no
▶ *Name of the N-th solid component.*			
N_solid-netgen-coarseness	fine	—	no
▶ *Level of coarseness of the generated mesh.*			
N_solid-netgen-hMax	10	—	no
▶ *Maximal size of element.*			
N_solid-netgen-hMin	0.1	—	no

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
▶ *Minimal size of element.*			
N_solid-netgen-grading	0.3	–	no
▶ *Willingness to change element size between neighboring elements.*			
N_solid-netgen-chartDistanceEnable	yes	–	no
▶ *Enable mesh size control with chart distance.*			
N_solid-netgen-chartDistancevalue	1.5	–	no
▶ *Limit element size by distance to the neighboring chart.*			
N_solid-netgen-lineLengthEnable	yes	–	no
▶ *Enable mesh size control with line length.*			
N_solid-netgen-lineLengthValue	0.5	–	no
▶ *Limit size of elements located near to chart boundary curves ends.*			
N_solid-netgen-closeEdgeEnable	yes	–	no
▶ *Enable mesh size control with close edges.*			
N_solid-netgen-closeEdgeValue	2.0	–	no
▶ *Limit size of elements located near to chart boundary curves by distance to other doundary curve.*			
N_solid-netgen-surfaceCurvatureEnable	yes	–	no
▶ *Enable mesh size control with surface curvature.*			
N_solid-netgen-surfaceCurvatureValue	1.0	–	no
▶ *Limit element size by surface curvature.*			
N_solid-netgen-edgeAngleEnable	yes	–	no
▶ *Enable meh size control with adge angle.*			
N_solid-netgen-edgeAngleValue	1.0	–	no
▶ *Limit element size by by chart boundary curvature.*			
N_solid-netgen-surfaceMeshCurvatureEnable	yes	–	no
▶ *Enable meh size control with adge angle.*			
N_solid-netgen-surfaceMeshCurvatureValue	3.0	–	no
▶ *Set number of elements per curvature radius.*			
N_solid-netgen-recalcMeshSizeForSurfOpt	yes	–	no

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
▶ *Recalculate mesh size for surface optimization.*			
N_solid-netgen-yellowEdgeAngle	30	–	no
▶ *Minimum angle between normals of adjacent triangles at which the common edge is considered as a feature edge.*			
N_solid-netgen-edgeCornerAngle	60	–	no
▶ *Minimum angle between adjacent edges of chart boundary, the chart is split for higher values*			
N_solid-netgen-chartAngle	15	–	no
▶ *Angle between normals of adjacent triangles under which the sahred edge is not considered as a chart boundary.*			
N_solid-netgen-outerChartAngle	70	–	no
▶ *Angle to identify overlapping parts of chart.*			
N_solid-netgen-recomputeNormals	yes	–	no
▶ *Normals are computed from the face coordinates instead of read from STL file.*			
N_solid-netgen-numberOf2DOptimizationSteps	3	–	no
▶ *Number of optimization steps for surface mesh.*			
N_solid-netgen-numberOf3DOptimizationSteps	5	–	no
▶ *Number of optimization steps for voume mesh.*			
N_solid-netgen-elementSizeWeight	0.2	–	no
▶ *Weight of triangle size badness.*			
N_solid-netgen-worstElementMeasure	2	–	no
▶ *Power of error used to approximate max error optimization.*			
N_solid-netgen-badElementCriterion	175	–	no
▶ *Elements with faces with angle higher then this value are considered as bad.*			
N_solid-gmsh-hMax	10	–	no
▶ *Maximal size of element.*			
N_solid-gmsh-hMin	0.1	–	no
▶ *Minimal size of element.*			
N_solid-gmsh-meshAlgorithm2D	frontal-delaunay	–	no
▶ *Gmsh algorithm to surface meshing.*			
N_solid-gmsh-meshAlgorithm3D	delaunay	–	no

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
▶ *GMsh algorithm to volume meshing.*			
N_solid-gmsh-maxAniso	1e33	–	no
▶ *Maximal anisotropy of Gmsh mesh.*			
N_solid-gmsh-sizeFromCurvature	7	–	no
▶ *Number of elements per $2\pi r$, where r is the local curvature radius.*			
N_solid-gmsh-minCircleNodes	5	–	no
▶ *Minimal number of nodes per circle.*			
N_solid-gmsh-minCurveNodes	5	–	no
▶ *Minimal number per curve (other then line, circle or elipse).*			
N_solid-gmsh-numberOfMeshAdaptSteps	6	–	no
▶ *Number of steps performed in Mesh adapt algorithm.*			
N_solid-gmsh-useSTLAsSurfaceMesh	no	–	no
▶ *Whether to you STL surface as a surface mesh.*			
N_solid-gmsh-yellowEdgesAngle	40	–	no
▶ *Minimum angle between normals of adjacent triangles at which the common edge is considered as a feature edge.*			
N_solid-gmsh-forceParametrization	yes	–	no
▶ *Parametrize surface patches.*			
N_solid-gmsh-splitCurvesOnAngle	160	–	no
▶ *Angle where the feature edge is split into two.*			
N_solid-gmsh-optimize	yes	–	no
▶ *Whether to perform mesh optimization.*			
N_solid-gmsh-optimizeThreshold	0.3	–	no
▶ *Elements of quality lower then this value are subject to optimization.*			
N_solid-gmsh-useNetgenOptimizer	yes	–	no
▶ *If set Netgen optimizer is used otherwise the native Gmsh optimizer is used.*			
N_solid-gmsh-numberOfSmoothingSteps	3	–	no
▶ *Number of smoothing steps performed on the mesh.*			

Chapter 9

TMESH – Notes & Recommendations

This chapter mainly deals with requirements for input data and its fundamental properties which have to be satisfied to get outstanding results with TFEA and TCFD workflow. Moreover, basic terminology for input mesh data and turbomachinery simulations are introduced.

9.1 Input geometry for TMESH

The input geometry preparation is a separate process from TCAE. It has significant influence on results quality; wrong input generates wrong result. Make always sure that your input geometry, either STL files or external meshes, has the best possible quality. The main quality parameters and common errors will be presented later on in this Section.

Typically, the geometry is created or modified in some CAD software. Afterwards, TCAE users have two basic options:

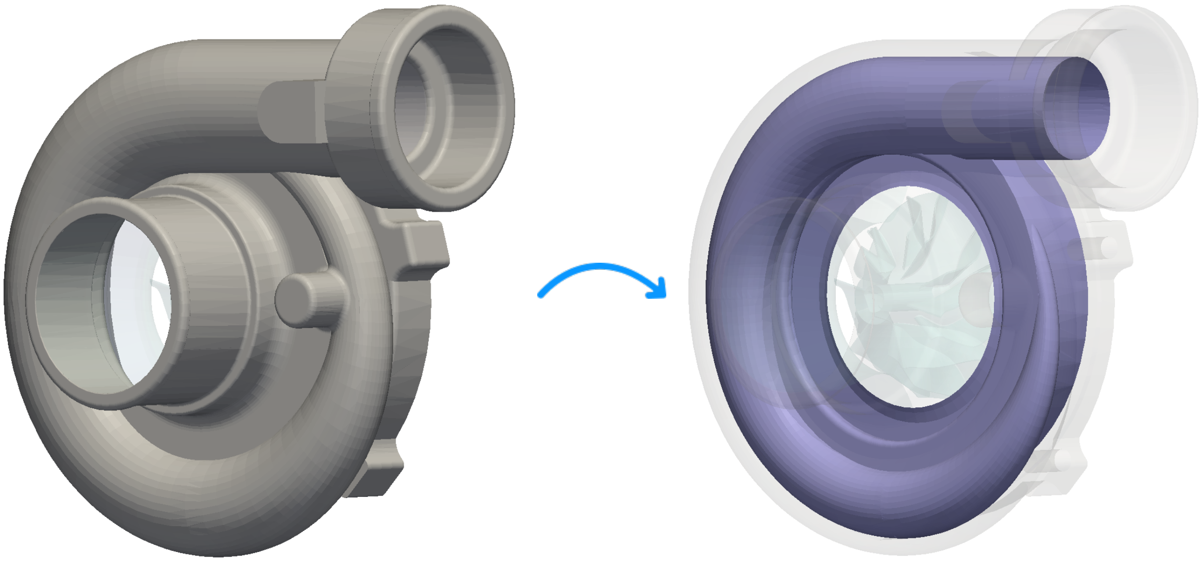
1. Generate and export STL geometry from CAD software and import the STL geometry into TCAE.
2. Import the CAD geometry into some meshing program and generate the computation mesh. The resulting mesh (in Fluent, CGNS or OpenFOAM format for TCFD simulation or in NetGen (*.vol) or Abaqus (*.inp) format for TFEA simulation) can be directly imported into TCAE.

Users can decide which approach to follow. Those who are familiar and experienced with some meshing software can follow the option **1** and those who have no preferences can choose the option **2**.

Both ways usually starts from the CAD geometry. The next sections bring general requirements for CAD geometries, we discuss separately the cases for TCFD and TFEA, as both computation engines have different requirements.

9.2 CFD - Flow path geometry

Any design of a real geometry contains complex details such as solids, screws, mounting joints, sealings, nuts, etc. To simulate a flow, a clear geometry describing a flow path (or a wet surface) is needed (see figure 9.1).



(a) Solid geometry

(b) Flow path geometry

Figure 9.1: Transition from solid geometry to flow path geometry

- **INPUT RULE 1:** Having a clear geometry describing the flow path.

9.3 CFD - Geometry details

Let us start with a very famous quota which can be applied to general tasks. Nevertheless, in the field of geometry preparation for CFD purposes it has significant importance:

"Everything should be made as simple as possible, but no simpler."

Albert Einstein

During a flow path geometry definition, one should keep in mind a simplicity of the geometry. For a simulation of a complex turbomachinery geometry, tiny details which does not significantly influence the flow should be removed from the flow path. Such details can not be captured by a reasonable amount of cells or can not be captured at all. The picture 9.2 depicts an (artificially created) example of such a detail.

- **INPUT RULE 2:** Create the flow path geometry as simple as possible.

9.4 CFD - Thinking in components

TCAE fully uses a beauty of component approach. Any geometry can be split into meaningful parts (**components**). Moreover, for rotating geometries, components can be grouped into rotating (rotors) and static (stators) components. For instance, for turbomachinery designs following components can be defined:

- stators
    - pipes

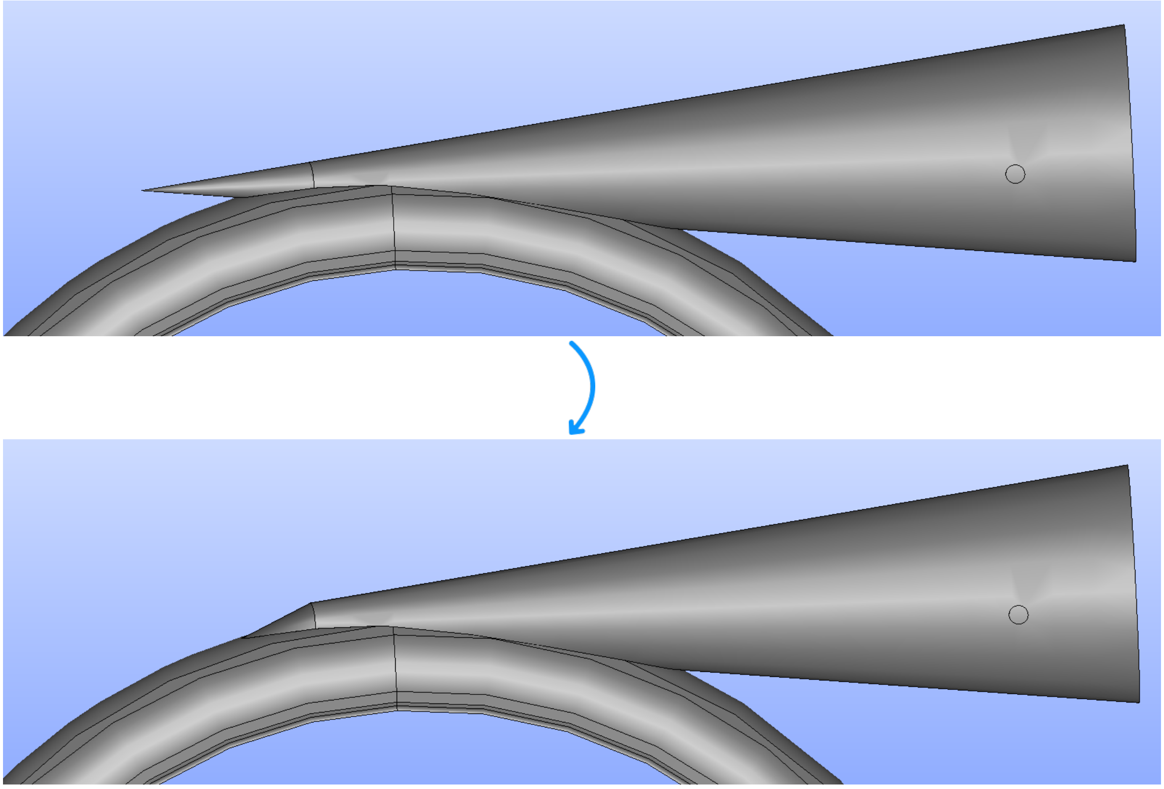


Figure 9.2: Geometry details

  - volutes
  - leakages
  - diffusers
  - general fluid domain
- rotors
  - impellers
  - propellers

Each component can be designed separately and afterwards connected together to build the whole machine geometry. Suppose the water pump geometry depicted in Fig. 9.3.

For CFD purposes, this pump must be disassembled into at least two parts: impeller and volute. Looking at the machine in more detail, we can disassembled it into four parts: inlet pipe, impeller, volute and outlet pipe.

The rotor component must have a rotational symmetry. Therefore, it would be possible to define the inlet pipe and the impeller as one component. But a more sensible option is to split it into two parts. The same treatment can be applied to the volute and the outlet pipe. It can be defined as one component or split into two components.

TCFD automatically evaluates intergral quantities at each interface between two components. Consequently, spliting the geometry into more components, as shown in Fig. 9.4 could be convenient.

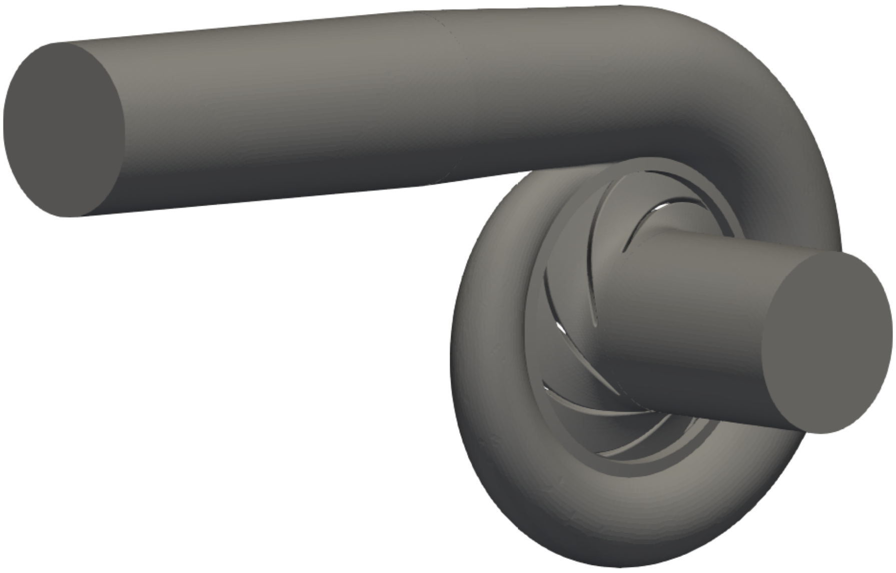
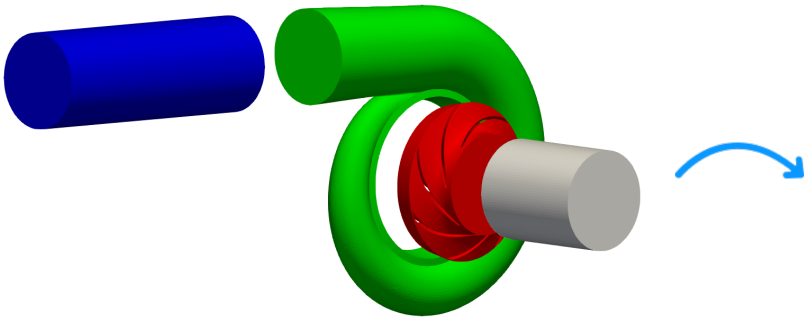
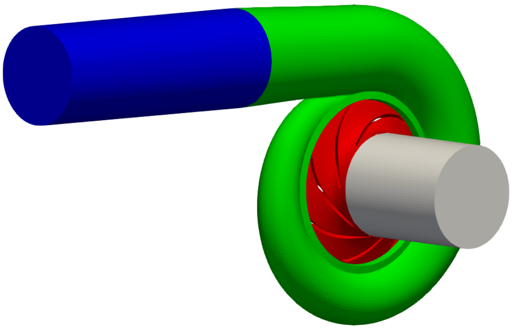


Figure 9.3: Water pump geometry



(a) Components creation



(c) Whole component geometry

Figure 9.4: Splitting geometry

- **INPUT RULE 3:** Split your geometry into reasonable amount of components having at least one rotor part with rotational symmetry.

Component properties

Each component has to satisfy following properties:

1. Partitioning into physical boundaries (inlet, outlet, blade, hub, ...)
2. Component interface alignment
3. Watertight property of input geometry

9.4.1 Physical boundaries

Each component has to be split into separate parts describing physical boundaries like inlet, outlet, walls, rotating walls, etc.

As a part of CFD process, each part of the geometry (**patch**) has to be described with a special type of boundary conditions. Therefore, it is necessary to have an access to:

- inlet and outlet boundaries for setting inlet and outlet boundary conditions,
- physical walls to set wall boundary condition.

Physical walls should be split as well to distinguish which part is rotating and which part is fixed. Specifically, for a rotor component, one should split the boundary into hub, shroud and blade part which is useful for a specific turbomachinery evaluation. Additionally, blade can be split into leading edge, trailing edge, suction side and pressure side to define different refinement at these walls or for detailed pressure distribution evaluation and other postprocessing purposes.

An example of a rotor component patch splitting is visualized in Figure 9.5.

- **INPUT RULE 4:** Each component has to be split into physical boundaries.

9.4.2 Component interface alignment

Each component has to be connected to at least one neighboring component. The corresponding interfaces, i.e., the inlet or outlet patch, have to be perfectly aligned to each other. For the radial pump example (see the figure 9.6):

- the outlet boundary of the inlet pipe component has to be perfectly aligned with the inlet boundary of the rotor component.
- the outlet boundary of the rotor component has to be perfectly aligned with the inlet boundary of the volute component
- the outlet boundary of the volute component has to be perfectly aligned with the inlet boundary of the outlet pipe component

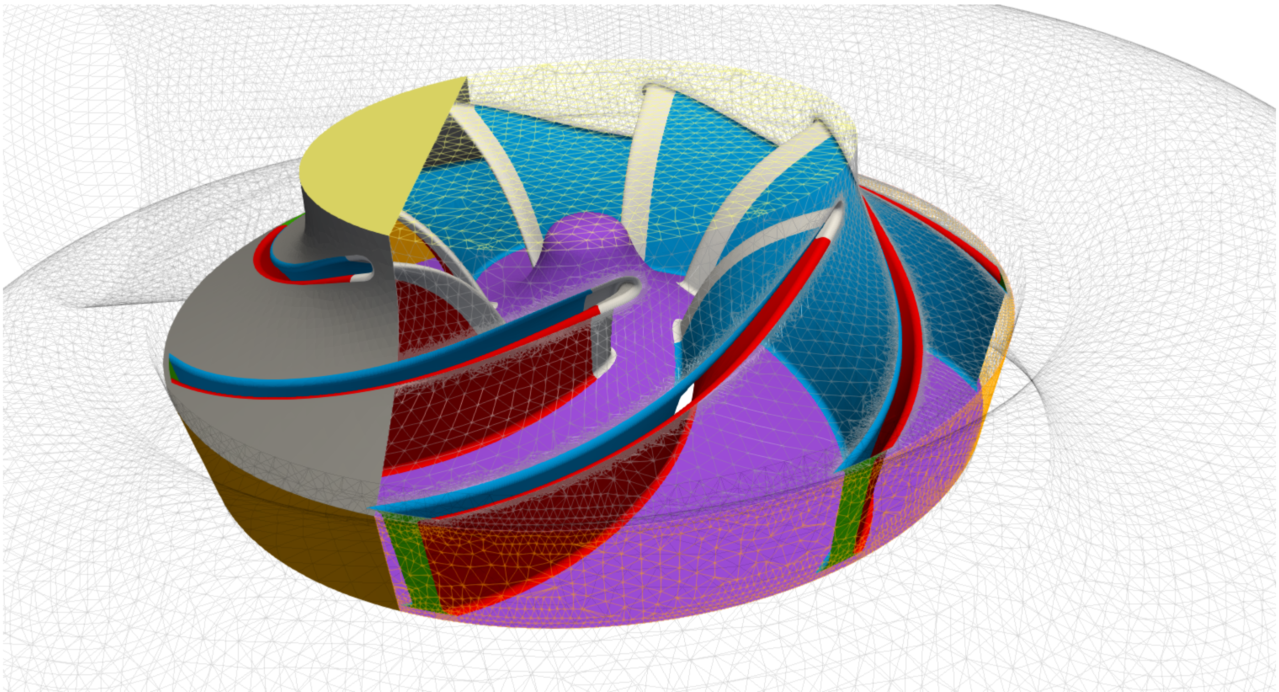


Figure 9.5: Component rotor patches

TCAE offers to impose so called mixing plane approach for components which defines just a segment of a full wheel geometry. In this case, neighboring interfaces have to be perfectly aligned in sense of its full wheel geometry, as depicted in figure 9.7.

- **INPUT RULE 5:** Each component has to be perfectly aligned with its neighbor(s).

Another example of inlet-outlet interface alignment is depicted in figure 9.8. We can see the rotor-volute interface. The rotor outlet (red) is perfectly aligned with the volute inlet (green). The orange color depicts a wall part of the volute component.

A usual mistake is to extended the green part to the whole orange cylindrical part. In this case the red part would be not perfectly aligned with the volute inlet which is not allowed.

9.4.3 Watertight property

Watertight property of the input STL geometry is extremely important for meshing process. The computational mesh is created and bounded by the STLs. Any hole inside, usually at edges of neighboring parts, may cause a crash of a meshing process. It is sometimes referred to as a watertight property or a point-to-point correspondence of vertices. The property is visualized in figure 9.9.

- **INPUT RULE 6:** Each component has to be water tight.

9.5 FEA - Geometry

The meshing process for TFEA simulation requires a single STL file. The geometry describes a surface of a single 3D solid. The representation needs to be clean, with no holes. In case of

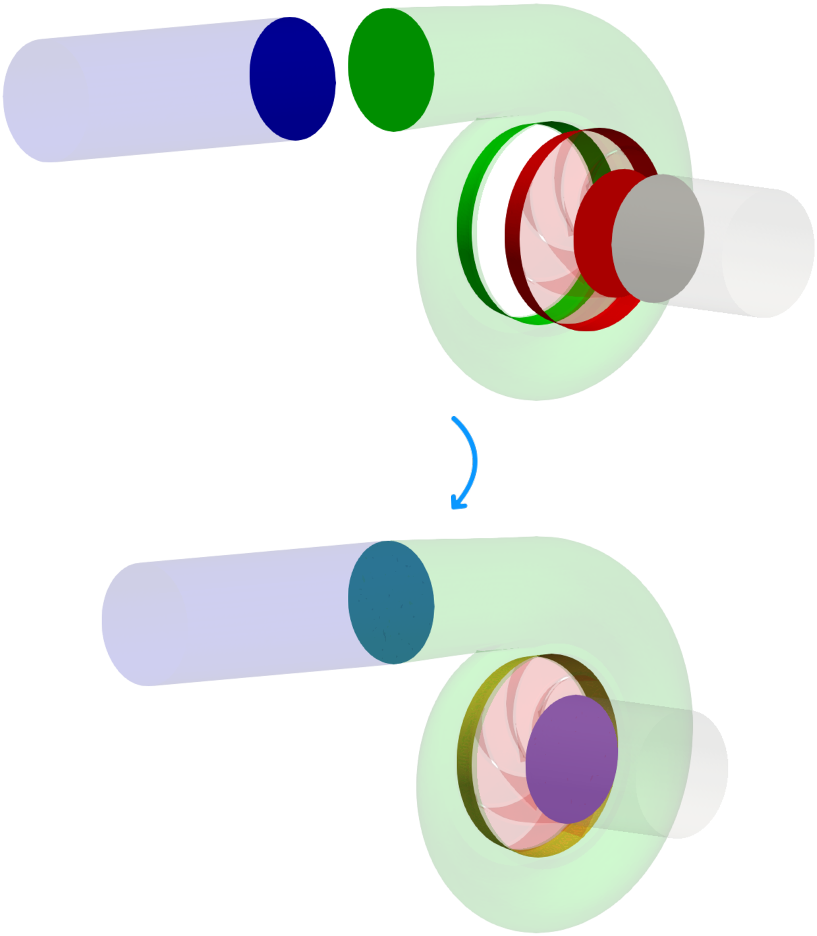


Figure 9.6: Component interface alignment

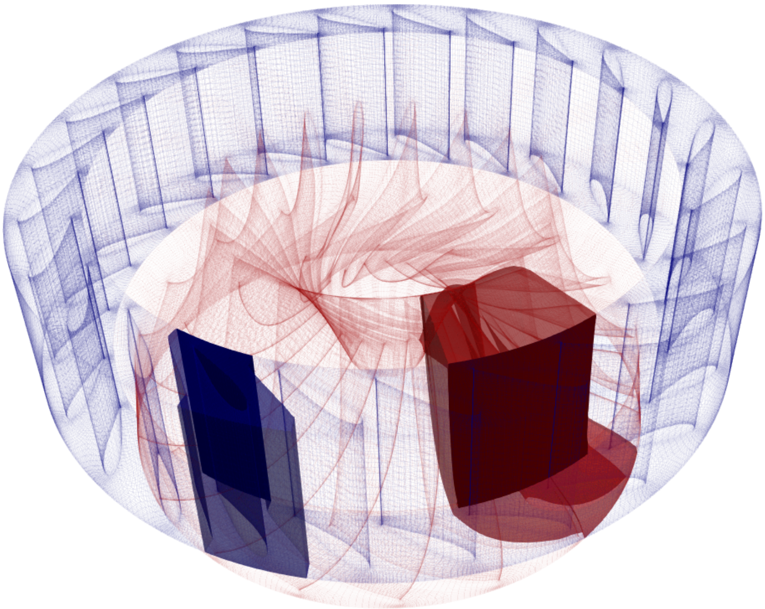


Figure 9.7: Geometry segment alignment

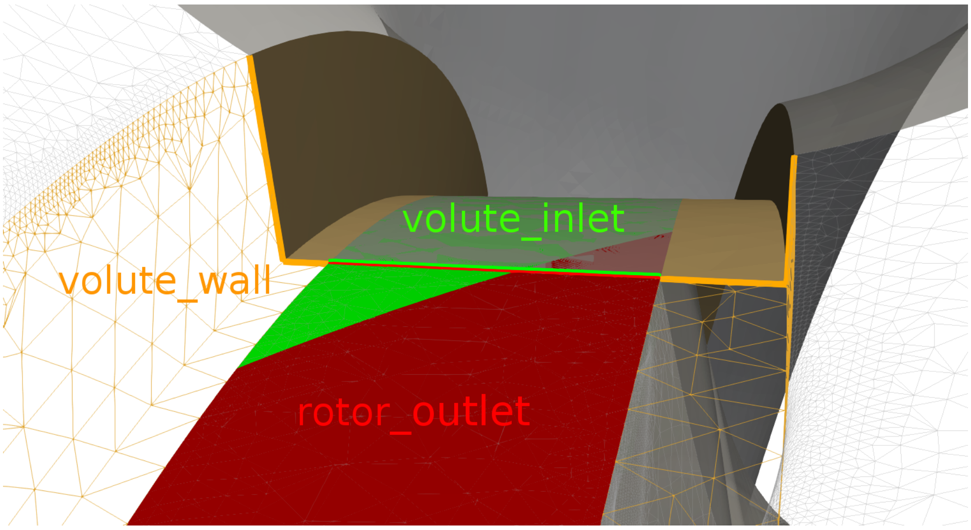
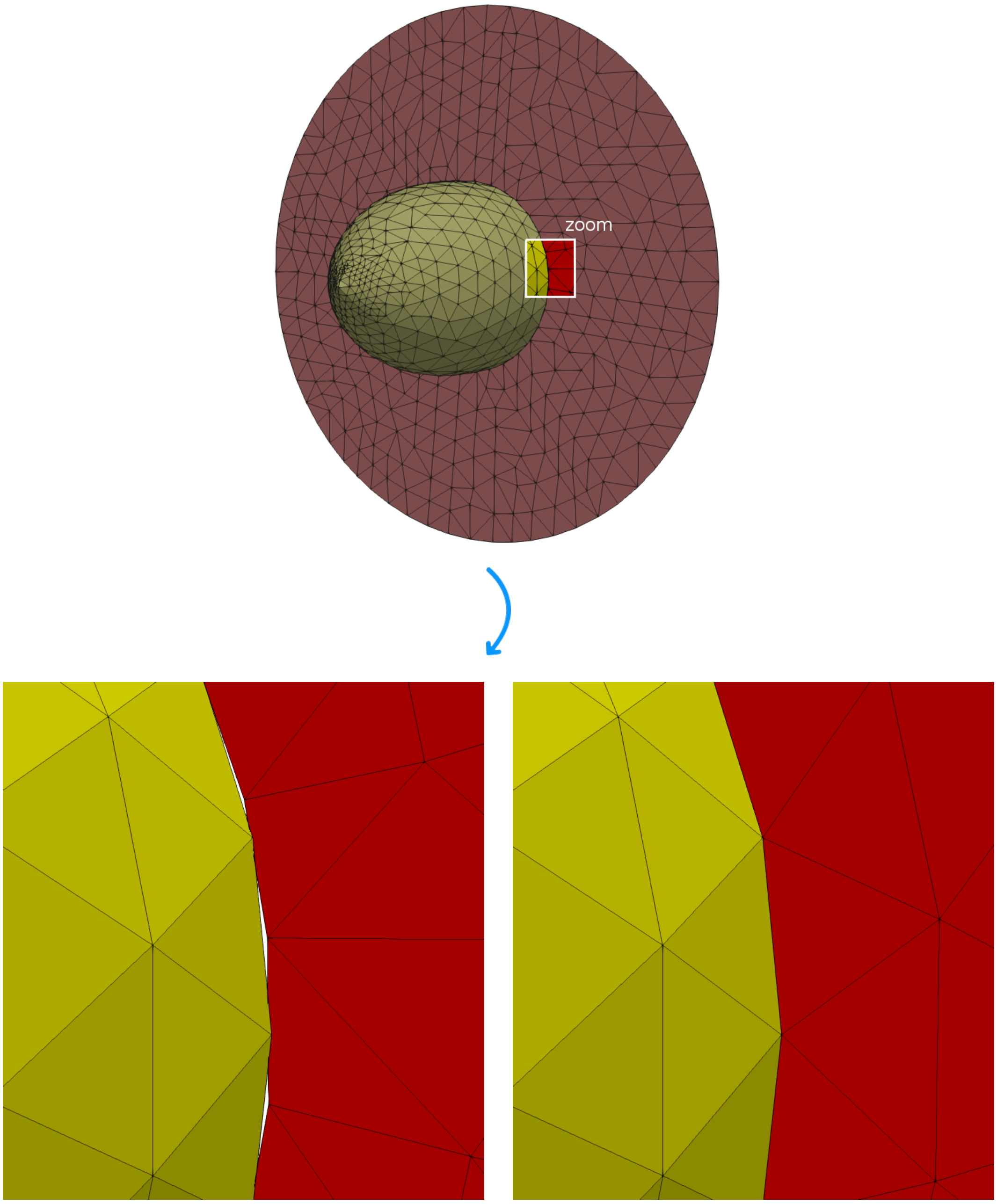


Figure 9.8: Rotor – volute interface



(a) Holes at neighbouring parts

(b) Watertight property

Figure 9.9: Watertight vs. non-waternight geometry

violating these requirements the STL loading will fail.

The NetGen meshing process goes as follows. Based on the given parameters, firstly the edges are meshed and, subsequently, the surface is meshed. Once the surface is meshed, the optimization (smoothing) steps are performed on the surface mesh. Having a nice surface mesh, the meshing process continues with meshing the volume, the meshing is again followed by mesh optimization steps.

Note that some geometries are more difficult to mesh than others. This could be case of geometries of complex shapes or when the geometry contains regions much thinner than the rest (lollypop-like shapes or turbine with thin blades). This (and other) kind of complex geometries can cause some troubles while meshing. By troubles we mean ugly mesh (low quality elements, too fine or too coarse), meshing taking too long (it can happen that NetGen won't give up and will try to mesh the geometry forever), or complete failing of the meshing process. You then need to play with the meshing options - the minimal and maximal size of elements, the grading (willingness to change size of neighbouring elements) or the number of optimization steps.

An example of a TFEA mesh can be seen in Figure 9.10.

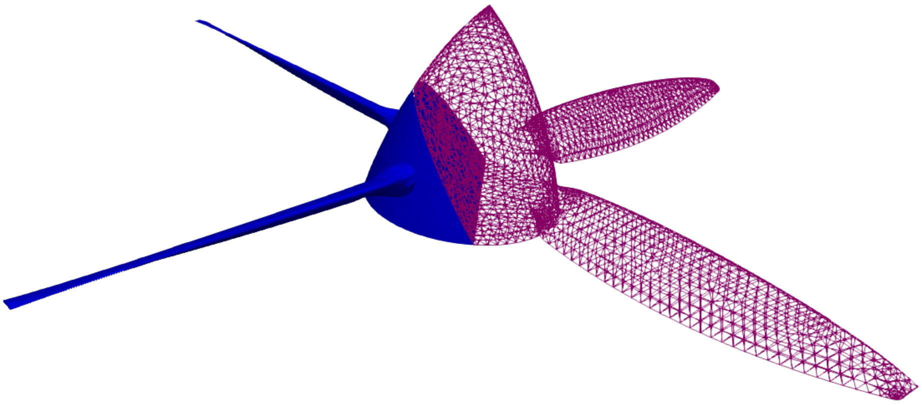


Figure 9.10: Example of meshing a geometry for TFEA calculation.

9.6 General Notes & Reccomendations

- Each component has to be waterproof.
- Watch out the model dimensions, they are critical.
- Ideally, each component surface STLs should fit vertex-to-vertex. Not necessary, but safest.
- All the interfaces between two components should overlap, or at least to fit each other very well.

- When simulating the turbomachinery it is not suitable to have "Trailing edge fixed on outlet" of the Impeller. It is recommended, when simulate, to extend the hub and shroud a little bit.
- Cylindrical background mesh has the following structure: It consist of three parts – outer cylinder, middle cylinder and inner block, which actually blend between square block and cylinder (see 9.11). The size of the cells is controlled by the Background mesh cell size (7.2.2). Other options are Cylindrical radii, Cylindrical grading and Cylindrical warp, which set the following parameters:
  - r_2 – the radius of the outer cylinder (if zero, it is set as 1.01 of the geometry radius)
  - r_1 – the radius of the middle cylinder (if zero, it is set as $0.5r_2$)
  - r_0 – the radius of the inner cylinder (if zero, it is set as $0.5r_1$)
  - g_2 – the mesh grading of the outer cylinder (gradual change of cell sizes)
  - g_1 – the mesh grading of the middle cylinder (gradual change of cell sizes)
  - w – the warp of the inner block, which controls its shape, and is defined by the relation

    $$w = \frac{1}{\sqrt{2}-1}\left(\sqrt{2}\frac{x}{r_0} - 1\right), \tag{9.1}$$

    If $w = 0$, the block collapses into the rectangle, as seen in 9.11).If $w = 1$, the inner blocks transforms to the cylinder with radius $x = r_0$.

  The length of the background mesh is set automatically, so the whole geometry is contained.

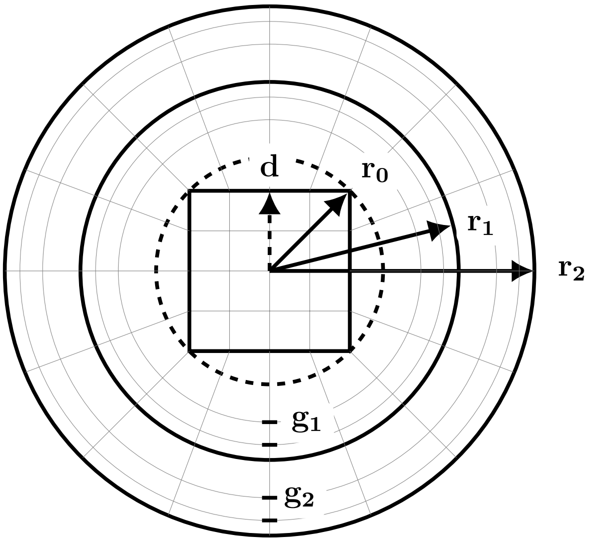


Figure 9.11: TMESH – Cylindrical background mesh (here viewed along the axis)

9.7 Rescue list

If the meshing of the CFD geometry in TMESH, or the following CFD simulation using TCFD goes wrong - please go through this list item by item and check your case settings again. Many typical meshing problems are caused by one of the following reasons:

1. Each TMESH case consists of components. Check the number of components. Check if the components are correctly connected, see the components graph (Figure 7.20) in the Components settings.
2. If you use automatic snappyHexMesh meshing, each component should be watertight. Check if all the components are watertight, see Sec. 9.4.3.
3. All the interfaces between two components must fit perfectly.
4. If you use a periodic case, check if the number of segments is correct.
5. Check if the mesh dimensions are set correctly (mm, m, km, ...).
6. Is the mesh OK visually?
7. If you use automatic snappyHexMesh meshing - check if the internal points are really inside the components (otherwise the "inverse" volume is meshed).
8. If you use automatic snappyHexMesh meshing - check if the background mesh follow the dimensions of the input geometry and if is reasonable fine. For the rotating turbomachinery, a reasonable background mesh cell size is the impeller diameter divided by 75.
9. If you use periodic segment case - check if the periodic patches (rotationAMI type) fit perfectly.
10. If your issue is still unresolved, please contact us at support@cfdsupport.com. The best way to do that is sharing the complete case to our technical support. If your case cannot be shared for any reason - please send us the detailed case description and zipped `TMESH/logRun` folder.
11. Rarely there can be this problem with MPI run. It occurs when MPI finds Loopback, virtual network interface and no other network interface. In this case MPI knows there are other network interfaces than just Loopback and tries to use them. However, MPI is not able to use virtual network interfaces and throws the following error

    ```
    No network interfaces were found for out-of-band communications.  We require
    at least one available network for out-of-band messaging.
    ```

    The solution is to edit $MPI_ARCH_PATH/etc/openmpi-mca-params.conf file and insert here this line oob_tcp_if_include = lo.

Part III

TCFD

Chapter 10

TCFD – Introduction

10.1 What is TCFD?

TCFD is one of the modules of the TCAE software, and is used for setting up the CFD simulation.

TCFD uses OpenFOAM as its computational core, with greatly enhanced capabilities by adding the custom solvers, boundary conditions and postprocessing utilities developed by CFD support.

TCFD is connected in the TCAE environment (Fig. 10.1) to the TMESH module, such that output of the TMESH (CFD mesh) serves as input to the TCFD. At the other end, the TFEA module might be connected to TCFD in the way, that results of the CFD simulation (partiaulary forces and temperature at the relevant surfaces) are passed to TFEA, where they are used for FEA calculation.

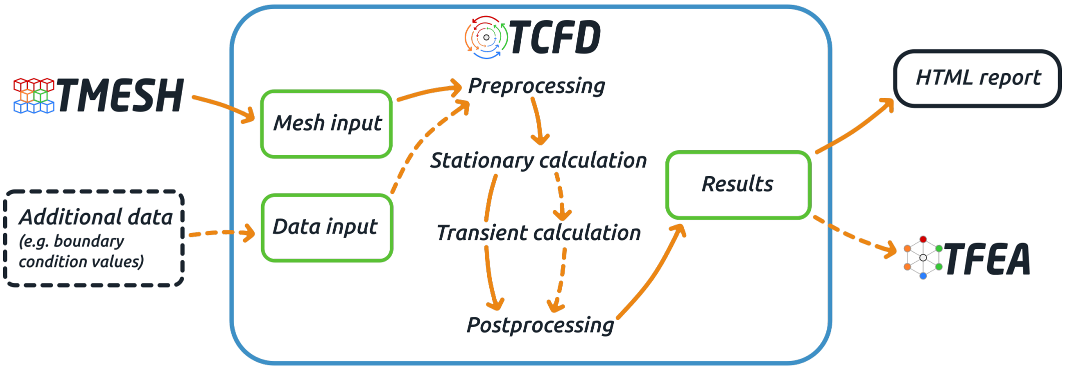


Figure 10.1: TCFD – data workflow

10.2 TCFD directory structure

TCFD directory is subdirectory of TCAE case (see 3.2).

The whole directory should be read-only, and its basic structure is shown in the table 10.1, because user might want to extract some useful data for postprocessing or to look at the logs.

🗀 project_folder	Project directory
└ 🗀 simulationRun	
└ 🗀 TCFD	all files, that belong to TCFD module
└ 🗀 figures	various postprocessing images used in HTML report, and gnuplot scripts used to make them
└ 🗀 logRun	output text logs of all of the applications, that are run by the TCFD
└ 🗀 logs	raw text logs with simulation score data and residuals
└ 🗀 postProcessing	raw data used for postprocessing
└ 🗀 stationary	OpenFOAM directory containing setup and results of stationary calculation
└ 🗀 transient	OpenFOAM directory containing setup and results of transient calculation
└ 🗀 report-steadystate-efficiency	HTML report with its images of the stationary calculation
└ 🗀 report-transient-efficiency	HTML report with its images of the transient calculation
└ 🗋 progress	

Table 10.1: TCFD – Directory structure

Chapter 11

TCFD – GUI Setup & Options

Module TCFD in the GUI can be added / removed by using the modules buttons (4.1) and is selected by the clicking on the *TCFD* item in the *Pipeline Browser*, or on one of its *Output Ports*, which are:

- Settings
  Using this port the keywords and their values of TCFD module are displayed as a table in *SpreadSheet View*.

- Report
  If the CFD calculation if finished and the HTML report is done, it will be displayed in *HTML View* through this port.

- Quantities
  If the CFD calculation is running, visibility of the charts displaying the live values of monitored quantities is controlled via this port (see 4.4.4).

- Residuals
  If the CFD calculation is running, visibility of the charts displaying the residuals for observing the convergence is controlled via this port (see 4.4.4).

After selection the TCFD, user can see what is depicted in the Fig. 11.1. In the *Properties Panel*, there are menus, their contents are thoroughly described in following sections: "GENERAL", "PHYSICS", "SIMULATION", "BOUNDARY CONDITIONS", "POST-PROCESSING".

11.1 GENERAL

The GENERAL section is displayed in Figure 11.2. It consists of only one entry – "Simulation type", which offers selection of the type of the machine that is to be set up. TCFD supports the following machines:

- fan
- propeller
- compressor
- pump

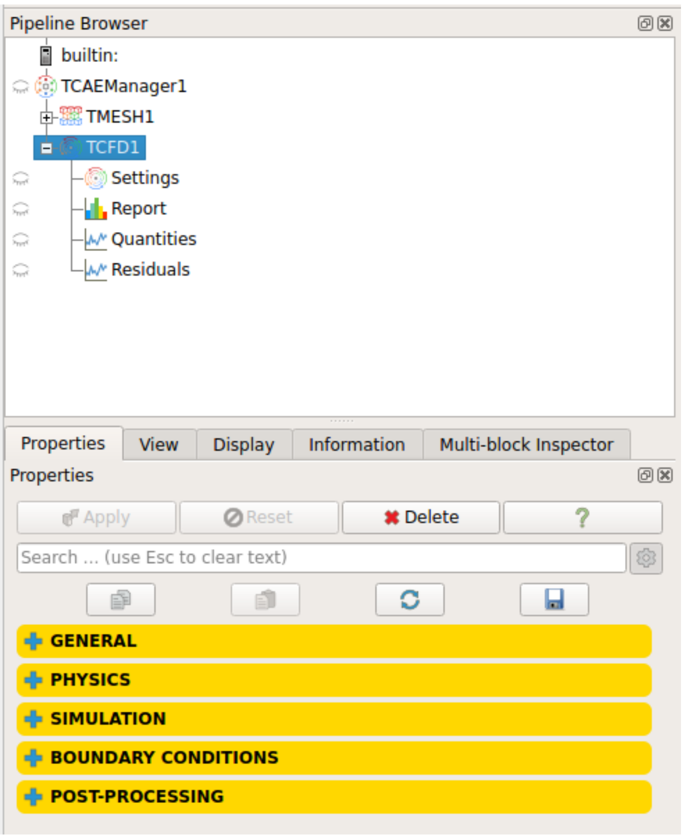
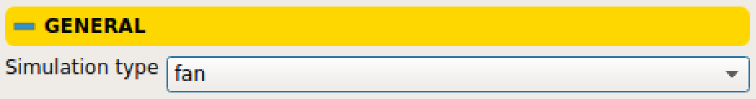


Figure 11.1: TCFD – *Pipeline Browser* and *Properties Panel*

- stator - general type without any rotational part
- virtualTunnel - for external aerodynamics
- turbine
- hydroTurbine
- windTurbine
- closedDomain - type with no inlets and outlets

This choice affects the postprocessing (each of the types employs specific postprocessing formulas for evaluating the efficiency, see Section 14.3), and enables or disables some of the setup features for particular machines (e.g. compressor has fixed physical model Compressible, cavitation can only be set for hydro turbines and pumps, some boundary condition types can be used only for one particular type etc.)



Simulation type: fan

Figure 11.2: TCFD – GENERAL.

11.2 PHYSICS

Menu PHYSICS, shown in Fig. 11.3 contains allx physical settings of the simulated case. It has these subsections: Time management, Fluid properties, Multiphysics, Turbulence and Rotation reference frames.

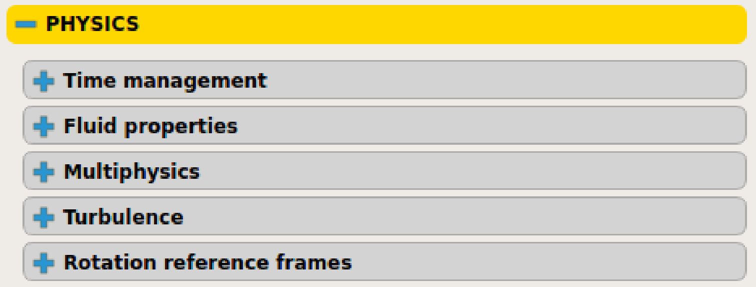
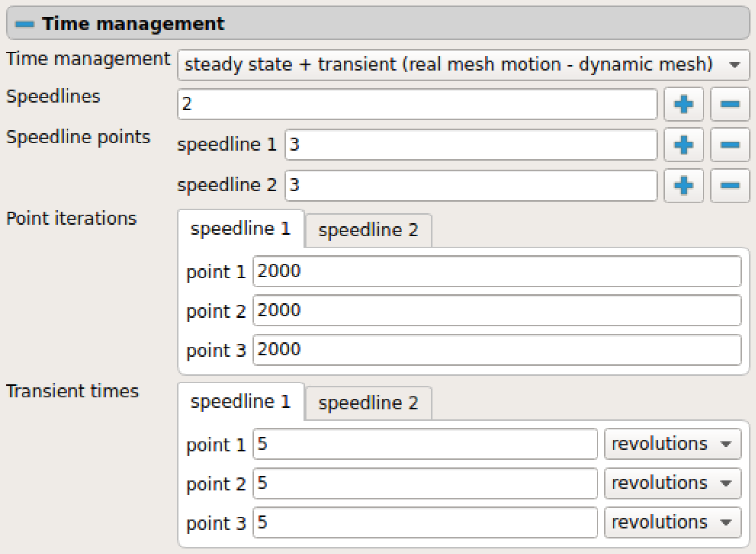


Figure 11.3: TCFD – PHYSICS.

11.2.1 Time management

- The calculation mode can be chosen from the drop-down list "Time management" . Here one can choose between a simple steady-state calculation, or its combination with some of the transient extensions, which will start from the precomputed steady-state results:
  - steady state: a standard steady-state calculation, where rotation (if applied) is simulated by the MRF method.



Time management

Time management: steady state + transient (real mesh motion - dynamic mesh)

Speedlines: 2

Speedline points: speedline 1: 3; speedline 2: 3

Point iterations (speedline 1 | speedline 2):

point 1	2000
point 2	2000
point 3	2000

Transient times (speedline 1 | speedline 2):

point 1	5	revolutions
point 2	5	revolutions
point 3	5	revolutions

Figure 11.4: TCFD – PHYSICS: Time management.

- steady state + transient (real mesh motion - dynamic mesh): classical transient calculation with mesh motion (if applied), where all Mixing planes are replaced by face-weighted interpolation through AMI.
- steady state + transient (static mesh + MRF): a "semi-transient" variant which solves transient equations, but the mesh is static and the rotation (if applied) is simulated by the MRF method as in the case of steady-state calculation.

NOTE: For simulations without rotation and Mixing plane interface connection, the steady state + transient options are identical.

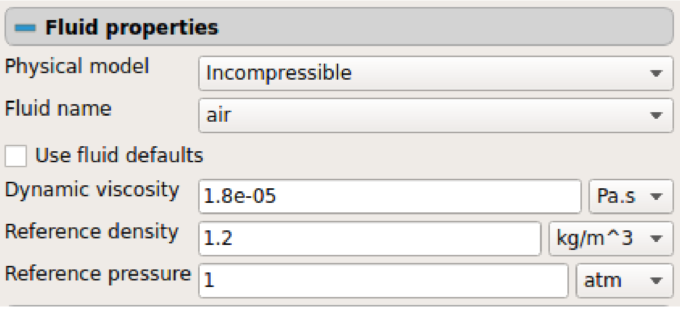
Every simulation consists of individual points with similar setup, which are grouped into speedlines sharing the rotation speed. Number of speedlines and number of points in every speedline is chosen in this section. Some other sections then adjust number of their inputs according to the numbers chosen here. The particular value of the rotation speed is set in "Reference frames" section (11.2.5).

- The field "Speedlines" sets the number of speedlines (groups of points with common rotation speed).
- The sliders in "Speedline points" set the number of points in every speedline.
- The entries in "Point iterations" set the number of iterations of steady-state calculation used for each of the points. This actually sets the maximal number of iterations. If there is some "Convergence check" (see section 11.3.3), the solver may terminate even before the iteration count reaches this number, if it decides that the efficiency and the fluxes are sufficiently converged.
- The entries in "Transient times" set the time span of transient calculation used for each of the points. As in stationary calculation, if there is some "Convergence check" (sec-

tion 11.3.3), the solver may terminate even before the end of the requested time interval, if it decides that the efficiency and the fluxes are sufficiently converged.

11.2.2 Fluid properties

The fluid properties are displayed in Figures 11.5 and 11.6.



Fluid properties

Physical model: Incompressible

Fluid name: air

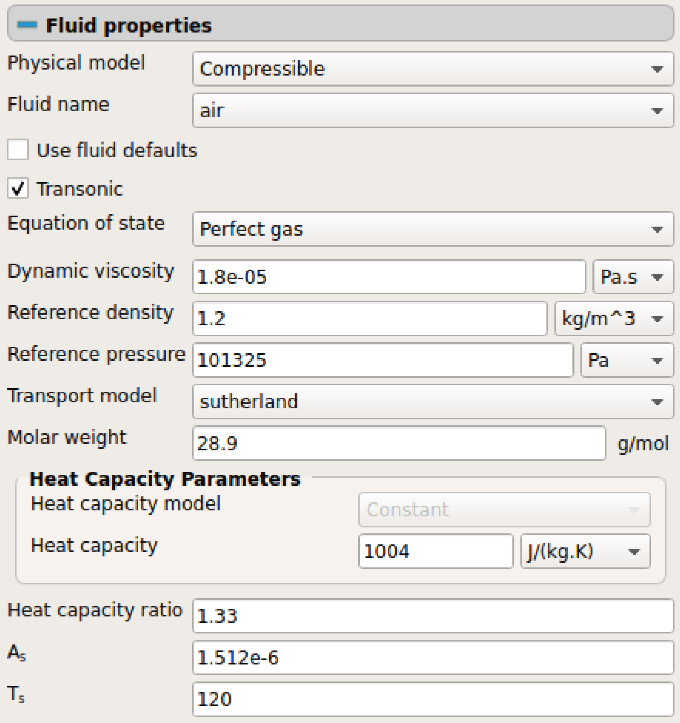
□ Use fluid defaults

Dynamic viscosity: 1.8e-05 Pa.s

Reference density: 1.2 kg/m^3

Reference pressure: 1 atm

Figure 11.5: TCFD – PHYSICS: Fluid properties. Incompressible model.



Fluid properties

Physical model: Compressible

Fluid name: air

□ Use fluid defaults

☑ Transonic

Equation of state: Perfect gas

Dynamic viscosity: 1.8e-05 Pa.s

Reference density: 1.2 kg/m^3

Reference pressure: 101325 Pa

Transport model: sutherland

Molar weight: 28.9 g/mol

Heat Capacity Parameters

Heat capacity model: Constant

Heat capacity: 1004 J/(kg.K)

Heat capacity ratio: 1.33

A_s: 1.512e-6

T_s: 120

Figure 11.6: TCFD – PHYSICS: Fluid properties. Compressible model.

Following options are present in this section:

- The switch "Physical model" determines the physical properties of the flow and the appropriate solver. Possibilities are:

  - "Incompressible" Simplest model for incompressible flows. Only the pressure and velocity fields are solved. Calculation is performed by TCFD solver blue(DyM)Solver, see section 14.1.1. Pump and hydroTurbine simulation types have always this value.
  - "Heat transfer" Model for heat transfer, applicable up to high subsonic speeds, but focused on low velocity flow with buoyant forces (with gravity). Additional temperature and density fields are solved. Calculation is performed by TCFD solver orange(DyM)Solver, see section 14.1.3.
  - "Compressible" Model for high speed compressible flows, both subsonic and transonic. Pressure, velocity, temperature and density fields are solved. Calculation is performed by TCFD solver red(DyM)Solver, see section 14.1.2. Turbine and compressor simulation types have always this value.

  In any case, appropriate turbulent quantities are solved. Choice of solver depends on the value of this switch. The form automatically shows and hides entries that are relevant for the current choice.

- The drop-down list "Fluid name" offers selection of the fluid. This option has effect only with conjunction with "Use fluid defaults" or "Cavitation risk".

- The switch "Use fluid defaults", when checked, hides some further fields and assumes default values for them that are appropriate for the selected fluid at standard conditions. These are summarized in the table 11.1.

- The switch "Transonic" enables transonic mode for compressible physical model. When selected, TCFD will activate transonic formulation of pressure equation. Transonic mode is activated for turbine and compressor simulation types in default.

- The switch "Equation of state" determines equation of state of the fluid, if heat transfer or compressible physical model is chosen. Current options are:

  - "PerfectGas" equation of state has the form

    $$p = \rho r T, \tag{11.1}$$

    where r is specific gas constant.
  - "Boussinesq" equation of state is allowed only for Heat transfer Physical model reads

    $$\rho = \rho_0[1 - \beta(T - T_o)], \tag{11.2}$$

    where β is thermal expansion coefficient, ρ_0 is reference density and T_0 reference temperature. Boussinesq equation of state is used for buoyancy driven flows.
  - "Peng-Robinson" equation of state is allowed only for Compressible Physical model and has the form

    $$\rho = \frac{p}{ZrT}, \tag{11.3}$$

    where Z is so called compressibility factor and is given by the cubic equation

    $$Z^3 - (1 - B)Z^2 + (A - 2B - 3B^2)Z - (AB - B^2 - B^3) = 0, \tag{11.4}$$

where coefficients are defined as:

$$A = \frac{\alpha a p}{R^2 T^2}, \quad B = \frac{bp}{RT} \tag{11.5}$$

$$a = 0.45724 \frac{R^2 T_c^2}{p_C}, \quad b = 0.0778 \frac{R T_c}{p_c} \tag{11.6}$$

$$\alpha = \left(1 + \kappa \left(1 - T_r^{1/2}\right)\right)^2, \quad \kappa = 0.37464 + 1.54226\omega - 0.26992\omega^2 \tag{11.7}$$

If PerfectGas/Peng-Robinson equation of state is selected, we say that the case/setup is compressible, whereas for the Boussinesq the case is incompressible. Compressible means, that density is function of pressure, which is true for PerfectGas/Peng-Robinson, while false for Boussinesq. To avoid confusion - in TCFD *compressible case/setup* is not equivalent to Compressible Physical model, as well as *incompressible case/setup* is not equivalent to Incompressible Physical model. *Incompressible case/setup* means Incompressible physical model or Heat transfer Physical model with Boussinesq equaton of state. *Compressible case/setup* means Compressible Physical model or Heat transfer Physical model with PerfectGas/Peng-Robinson equation of state (see figure 11.7).

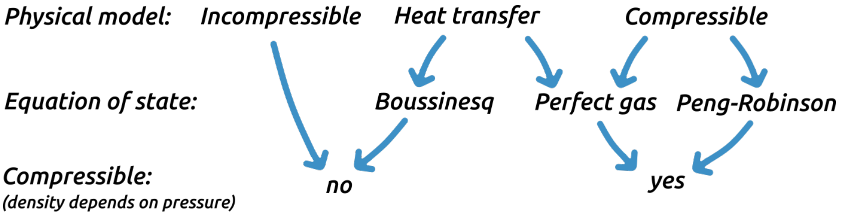


Figure 11.7: TCFD physical models and equations of state.

- The value of "Thermal expansion coefficient" is needed for Boussinesq equation of state, as described above.
- If Peng-Robinson is selected as Equation of state, then the box Peng-Robinson equation of state parameters appears. It contains the coefficients needed for the equations described above.
  - Critical temperature T_c - temperature at the critical point
  - Critical molar volume V_c - molar volume at the critical point
  - Critical pressure p_c - pressure at the critical point
  - Acentric factor ω
- The value of "Dynamic viscosity" specifies the dynamic viscosity (μ) of the fluid. For incompressible cases it is $\mu = \rho\nu$, where ν is the kinematic viscosity.
- The value of "Reference density" is used for three purposes. At first for postprocessing of cases without density (incompressible physical model), where the density is not considered by the solver. At second it is used for setting turbulent quantities at the inlet for cases with compressible/heat transfer physical model, see Section 11.4.1 page 206. And finally it is required by Boussinesq equation of state, if selected.

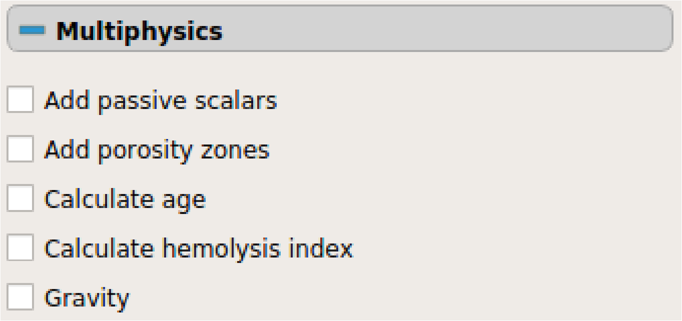
- The value of "Reference pressure" is used as a pressure value with respect to which all other pressures are measured. Typically, in incompressible simulation the user prescribes zero outlet pressure, while he or she actually means the ambient pressure. This entry then sets the ambient pressure.
- The value of "Reference temperature" is the value of the ambient temperature. It is required for postprocessing of "Cavitation risk" and for Boussinesq equation of state, if one of these features is employed.
- The list "Transport model" can be used to select the preferred transport (viscosity) model. There two models available: "constant" and "Sutherland". When "Sutherland" transport model is selected, the entries Ts and As (parameters of the model) will show up

  $$\mu = \frac{A_s T^{\frac{3}{2}}}{T_s + T}\,. \tag{11.8}$$

  In contrast, when "constant" is selected, then one can set the Prandtl number Pr.
- The entry "Molar weight" is used only in simulations with temperature to set up the thermophysical properties of the fluid. Molar weight is a standard chemical property of a substance.
- The box "Heat Capacity Parameters" is shown if the setup and contains parameters describing the treatment of the heat capacity at constant pressure c_p. In the list "Heat capacity model" the model can be selected, but currently the only option is the "Constant", that means, that the heat capacity is independent on the temperature, and its value is set in the entry "Heat capacity".
- The value of "Heat capacity ratio", $\gamma = c_p/c_V$, is used by the boundary conditions Total pressure, Total temperature and Opening for the compressible cases. If one of these boundary condition is not used, this entry is ignored.

11.2.3 Multiphysics

This subsection is shown in figure 11.8 and contains these options:



Multiphysics

□ Add passive scalars

□ Add porosity zones

□ Calculate age

□ Calculate hemolysis index

□ Gravity

Figure 11.8: TCFD – PHYSICS: Multiphysics.

- The switch "Cavitation risk" (See Figure 11.9) can be used to request estimation of the cavitation. This switch is only available for pump, hydroTurbine and propeller machines together with steady-state simulation. The model was implemented according to following literature: [6], [17], [18], [37]. See the section 14.8 for details.

fig/TCFD-cavitationRisk.png

Figure 11.9: TCFD – Cavitation risk parameters.

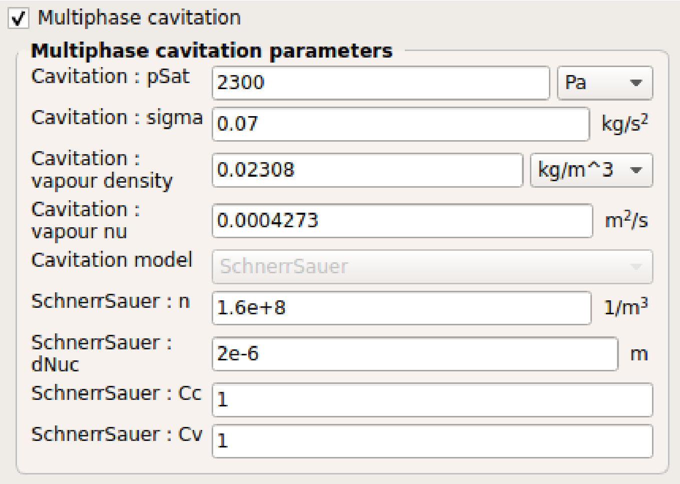
- The switch "Multiphase cavitation" will request use of a specialized cavitation solver greenDyMSolver (section 14.1.4) for the transient phase of the calculation. When this switch is enabled, additional cavitation-related parameters can be set (Figure 11.10). Currently there is only one cavitation model available for this solver – the Schnerr-Sauer model [5]. Multiphase cavitation is only available for water pumps and hydro turbines with transient calculation enabled. See the section 14.8 for details.

- The switch "Calculate comfort(PMV, PPD)" (avaiable only if Physical model: Heat Transfer is selected) switches on calculation of thermal comfort levels PMV and PPD based on DIN ISO EN 7730:2005.

- The switch "Calculate hemolysisIndex" (avaiable only if Physical model: Compressible is selected) swiches on the calculation of hemolysis index.

- The switch "Add passive scalar" allows user to add arbitrary number of passive scalars to the simulation. Their value will be computed with simple convection-diffusion equation. The diffusion coefficient can be prescribed directly ("Diffusivity type" = "constant"), or as a function of dynamic and turbulent viscosity ("Diffusivity type" = "turbulent") as:

$$\mathrm{D} = \mathrm{AlphaD} \cdot \nu + \mathrm{AlphaDt} \cdot \nu_\mathrm{t}. \tag{11.9}$$



☑ Multiphase cavitation

Multiphase cavitation parameters

Cavitation : pSat 2300 Pa

Cavitation : sigma 0.07 kg/s^2

Cavitation : vapour density 0.02308 kg/m^3

Cavitation : vapour nu 0.0004273 m^2/s

Cavitation model SchnerrSauer

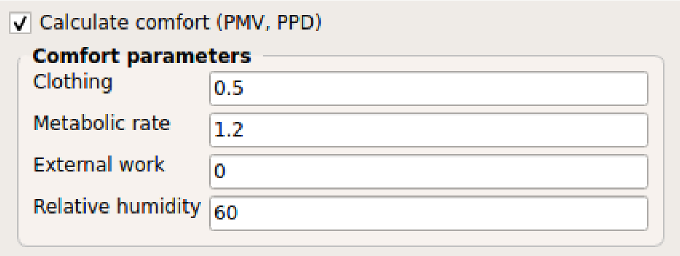
SchnerrSauer : n 1.6e+8 1/m^3

SchnerrSauer : dNuc 2e-6 m

SchnerrSauer : Cc 1

SchnerrSauer : Cv 1

Figure 11.10: TCFD – Multiphase cavitation parameters.



☑ Calculate comfort (PMV, PPD)

Comfort parameters

Clothing 0.5

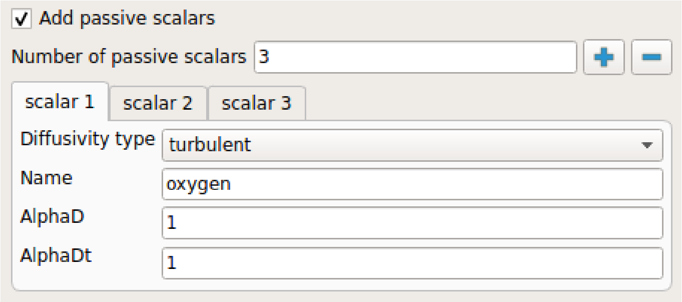
Metabolic rate 1.2

External work 0

Relative humidity 60

Figure 11.11: TCFD – Comfort parameters.

Inlet and initial values for the scalars have to also be defined in BOUNDARY CONDITIONS : Inlet section.



☑ Add passive scalars

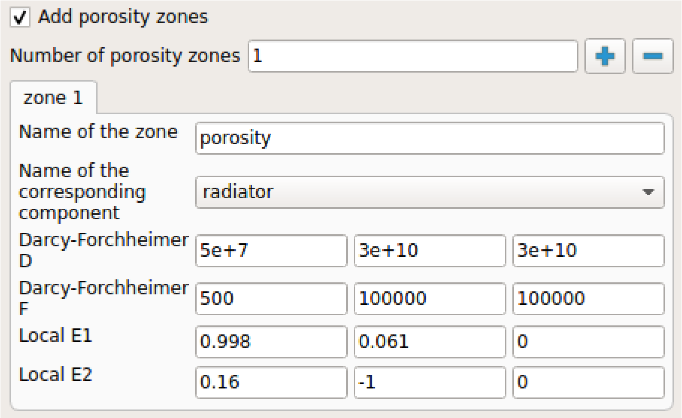
Number of passive scalars: 3

scalar 1 | scalar 2 | scalar 3

Diffusivity type	turbulent
Name	oxygen
AlphaD	1
AlphaDt	1

Figure 11.12: TCFD – Passive scalars.

- The switch "Add porosity zones" allows the user to treat arbitrary component as a porosity zone. The flow in porus meadia is modeled with Darcy-Forchheimer model, here the properties of the porus media are controled by two vectors, **D** and **F** which set the properties along the local coordinate axes. If the vector **F** is set to zero vector the model reduces to the Darcy law. The local coordinate system is determined by the vectors, $\mathbf{E}_1$ and $\mathbf{E}_2$, these are the first two coordinate axes (hence should be orthogonal). The third coordinate axis is then perpendicular to both of them.



☑ Add porosity zones

Number of porosity zones: 1

zone 1

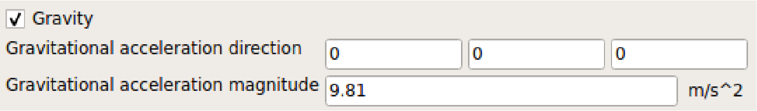
Name of the zone	porosity		
Name of the corresponding component	radiator		
Darcy-Forchheimer D	5e+7	3e+10	3e+10
Darcy-Forchheimer F	500	100000	100000
Local E1	0.998	0.061	0
Local E2	0.16	-1	0

Figure 11.13: TCFD – Porosity zones.

- The switch "Calculate age" switches on the calculation of the age of the fluid present in simulation.
- The switch "Gravity" shows additional variables needed for explicitly adding the gravitational force to the calculation. Section 14.5 gives some hints how to deal with gravity in the simulation.
  - The entry "Gravitational acceleration direction" sets the direction of the gravitational acceleration vector. For most applications the gravitational force is negligible and these components can be zero. The zero direction means no gravitational acceleration

during the computations, no matter what the value of "Gravitational acceleration magnitude" is. For large hydro (water) turbines it may be important. The typical setup with z axis pointing upwards would be the vector $(0, 0, -1.0)$.

– The entry "Gravitational acceleration magnitude" sets the magnitude of the gravitational acceleration vector. If the "Gravitational acceleration direction" is non-zero the gravitational acceleration is computed as "Gravitational acceleration direction" × (normalized to one) × "Gravitational acceleration magnitude". If "Gravitational acceleration direction" is a zero vector the computations run without gravity, the value of "Gravitational acceleration magnitude" can still be used in post-processing. The default value to "Gravitational acceleration magnitude" is $9.81[m/s^2]$.



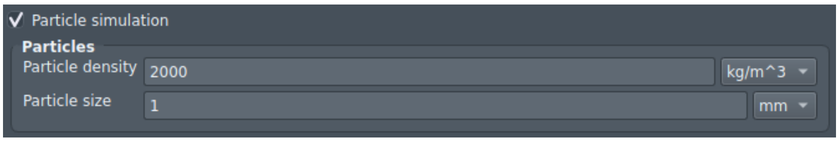
☑ Gravity

Gravitational acceleration direction: 0 | 0 | 0

Gravitational acceleration magnitude: 9.81 m/s^2

Figure 11.14: TCFD – gravity.

- The switch "Particle simulation" allows the user to run simulation of physical particle cloud in the specified velocity field. The user has to specify density and a diemeter of a single particle. Number of particles entering the domain has to also be defined in BOUNDARY CONDITIONS : Inlet section.



☑ Particle simulation

Particles

Particle density: 2000 kg/m^3

Particle size: 1 mm

Figure 11.15: TCFD – particle simulation.

quantity	symbol	unit	value	
			air	water
dynamic viscosity	μ	$\mathrm{Pa \cdot s}$	1.8×10^{-5}	10^{-3}
reference density	ρ	$\mathrm{kg \cdot m^{-3}}$	1.2	998
molar weight	M	$\mathrm{kg \cdot mol^{-1}}$	28.9	18.015
heat capacity	c_p	$\mathrm{J \cdot kg^{-1} \cdot K^{-1}}$	1004	4181.8
Sutherland A_S	A_S	–	$1.512 \cdot 10^{-6}$	$1.512 \cdot 10^{-6}$
Sutherland T_S	T_S	K	120	120
Prandtl number	Pr	–	0.7	7
heat capacity ratio	γ	–	1.4	1.33

Table 11.1: TCFD – Fluid defaults used when Use fluid defaults is checked.

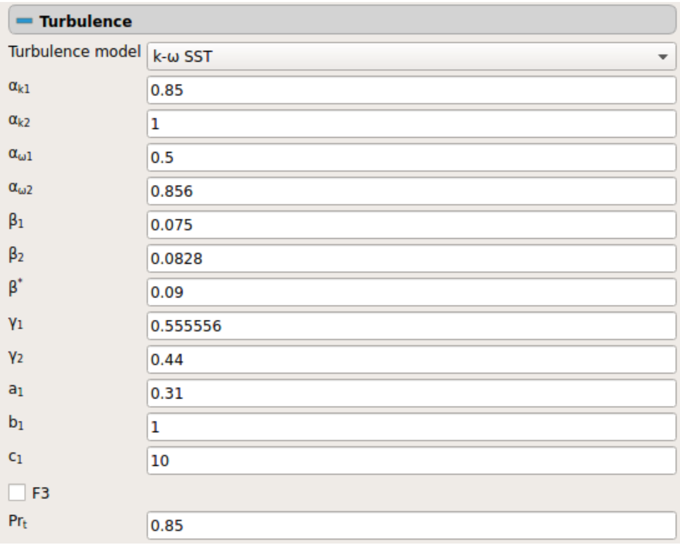
11.2.4 Turbulence

In section "Turbulence" the turbulence modelling is set up (figure 11.16). The "Turbulence model" option switches the model. Possible values are:

- "laminar" : no turbulence
- "kOmegaSST" : $k-\omega$ SST model
- "kOmegaSSTLM" : $k-\omega$ SST-LM model (*Launder-Mentry*)
- "kEpsilon" : $k-\epsilon$ model
- "realizableKE" : realizable $k-\epsilon$ model
- "RNGkEpsilon" : RNG $k-\epsilon$ model
- "Spalart-Allmaras" : Spalart-Allmaras (RANS variant) model
- "Spalart-AllmarasDDES" : Spalart-Allmaras DDES model (implies use of Spalart-Allmaras RANS in stationary calculation and Spalart-Allmaras DDES in the transient one)

If the advanced parameters are enabled, user can also tune parameters of the chosen turbulence model.

Alos, there is an entry "Pr_t" where turbulent Prandtl number is chosen.



Turbulence

Turbulence model: k-ω SST

α_{k1}: 0.85

α_{k2}: 1

$\alpha_{\omega 1}$: 0.5

$\alpha_{\omega 2}$: 0.856

β_1: 0.075

β_2: 0.0828

β^*: 0.09

γ_1: 0.555556

γ_2: 0.44

a_1: 0.31

b_1: 1

c_1: 10

□ F3

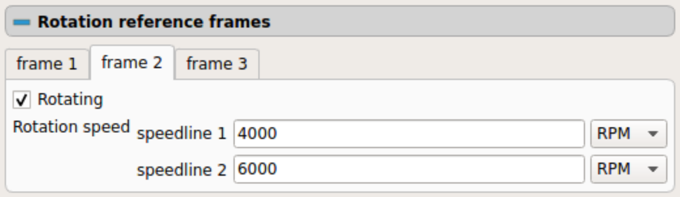
Pr_t: 0.85

Figure 11.16: TCFD – PHYSICS: Turbulence.

11.2.5 Rotation reference frames

This section, depicted in Figure 11.17, prescribes the rotational properties of reference frames. One is able to give desired rotation to the frames created in TMESH, to which these reference frames here correspond (see section 7.2.1).

- Switch "Rotating" turns on/off the rotation of the referece frame around its axis, so the components and patches, which have the particular frame assigned, will rotate.
- The entries in "Rotation speed" set the rotation speed of the frame for every speedline, either in radians per second or in full revolutions per minute (RPM). The unit is chosen in the drop-down box to the right. The number should be positive when the vector of the angular velocity points in the same direction as the rotation axis (right-hand rule) and negative when it points into the opposite direction. The number of speedlines is set in PHYSICS : Time management section (11.2.1).


Rotation reference frames

frame 1 | frame 2 | frame 3

☑ Rotating

Rotation speed

speedline 1: 4000 RPM

speedline 2: 6000 RPM

Figure 11.17: TCFD – PHYSICS: Rotation reference frames.

11.3 SIMULATION

This section (displayed in Figure 11.18) contains the options dedicated to the actual running of the simulation, i.e. the solver settings, running in parallel, numerical setup, scripting and evaluating of the quantities during the run.

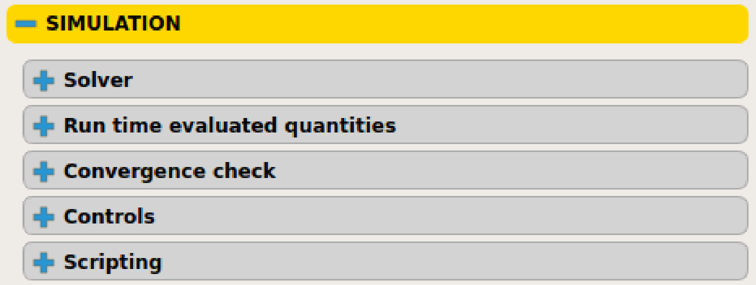


Figure 11.18: TCFD – SIMULATION.

11.3.1 Solver

In this section the basic settings of the solver are to be adjusted (figure 11.19).

- "Time step" – When transient mode is chosen, it is possible to choose two methods for time step definition:
  - "Constant": Uses a constant time step in seconds for the whole simulation, which is set in the field DeltaT.

  - "Adaptive": Computes the maximal value of time step based on the given Courant number value throughout the simulation domain.

- "DeltaT" shows, if the Time step is set to Constant and adjusts the fixed time step of the transient simulation (user can change its value during the calculation). Its unit is second, by default, but additionally, for problems with rotation, it can be set by means of angle in "degrees" of preset rotation speed (if there are more reference frames with rotation, DeltaT is computed from the lowest angular velocity of all of the reference frames (see Section 11.2.5).

- "Courant number" shows, if the Time step is set to Adaptive. The Courant number is a parameter evaluated for each cell and is given as

$$Co = \frac{\Delta t}{2V} \sum_{i \in \text{faces}} |\Phi_i| \ , \tag{11.10}$$

  where V denotes the volume of a cell, Φ_i is the face volumetric flux and faces denotes the set of all faces of the cell. By specifying its maximum value, the appropriate time step Δt can be determined. The transient solver operates in PIMPLE mode to be robust even for large time steps (Courant number $Co > 1$), but still, a reasonably low time step values should be defined (because of both the accuracy and stability).

- The entry "Number of processors" specifies number of processes used to run the solver. TCAE (more precisely OpenFOAM running in background) uses MPI (Message Passing Interface) for communication of the processes.

- The checkbox "Use custom solver" controls the visibility of the two following entries, which are used to switch to custom solvers.

  - The entry "Solver steady state" allows you to use a user-compiled solver within the steady-state TCFD simulation.
  - When transient mode is chosen, the entry "Solver transient" allows you to use a user-compiled solver within the transient TCFD simulation.

Important note:

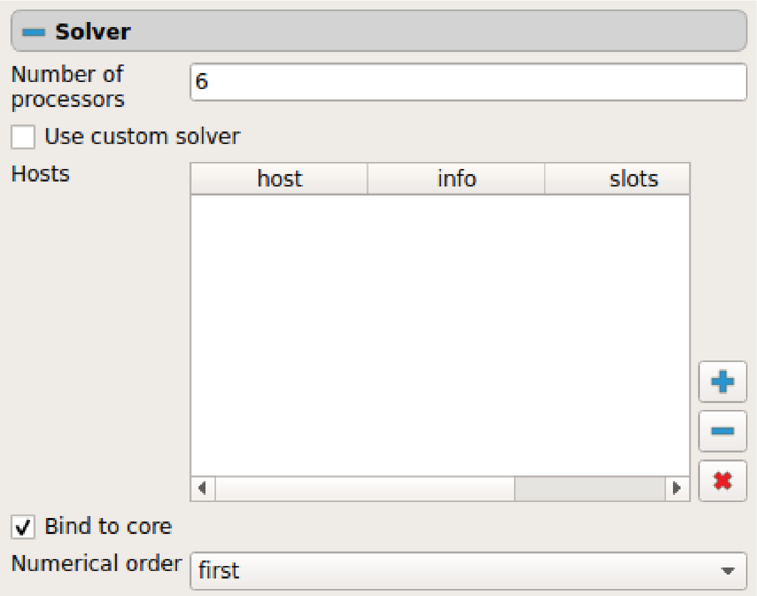
If you need to use your own OpenFOAM solver, it is strongly recommended to take one of our solvers as basis, and then complement it with additional desired features. It is because TCFD puts some demand on the solver (such as some additional parameters are passed to solver on start and the solver output has to adhere to the specific format), and if the solver is incompatible, it will not work.

- The table "Hosts" (advanced) can be used in conjunction with the "Number of Processors" entry to schedule the running of the parallel jobs. This table contains nodes, on which the parallel processes will be launched. The total number of processes given by "Number of Processors" is evenly divided between the nodes. A new node is added by the plus button. This is by default the "localhost" node, but the word "localhost" in the first column can be edited (double-click to enable editing) to any other host name or IP address. The simple local network information provided by the system utility "getent" is then shown in the second column. If the node is unknown, the table row will turn red. Such nodes must be deleted, or the execution will fail. There are several restrictions on the usage of remote nodes:

  - All nodes must be accessible from the workstation where TCAE runs without password for the current user (i.e. using the public key authentication), and the same must be true for access between the nodes.
  - The MPI and TCAE installations on the workstation and all the nodes must be identical. Ideally, there is just one installation on a shared network file system.
  - The TCAE case directory is written on a shared network file system, so that it is accessible both to the front-end workstation used for solution management via TCAE and to the remote calculation nodes.

  Hosts scheduling is currently possible only in Linux systems.

- The switch "Bind to core" (advanced) prevents migration of parallel processes between cores, possibly resulting in some speedup. TCFD takes care for allocation and locking of the processes to the most free CPU cores. (Currently only used in Linux-based systems.)
- The entry "Numerical order" sets the discretization order of the convection term. All calculation should converge with the first order. The second order generally provides more accurate results, but the simulations are often less stable and may require better meshes or other tuning.



Solver

Number of processors: 6

☐ Use custom solver

Hosts

host	info	slots

☑ Bind to core

Numerical order: first

Figure 11.19: TCFD – SIMULATION: Solver.

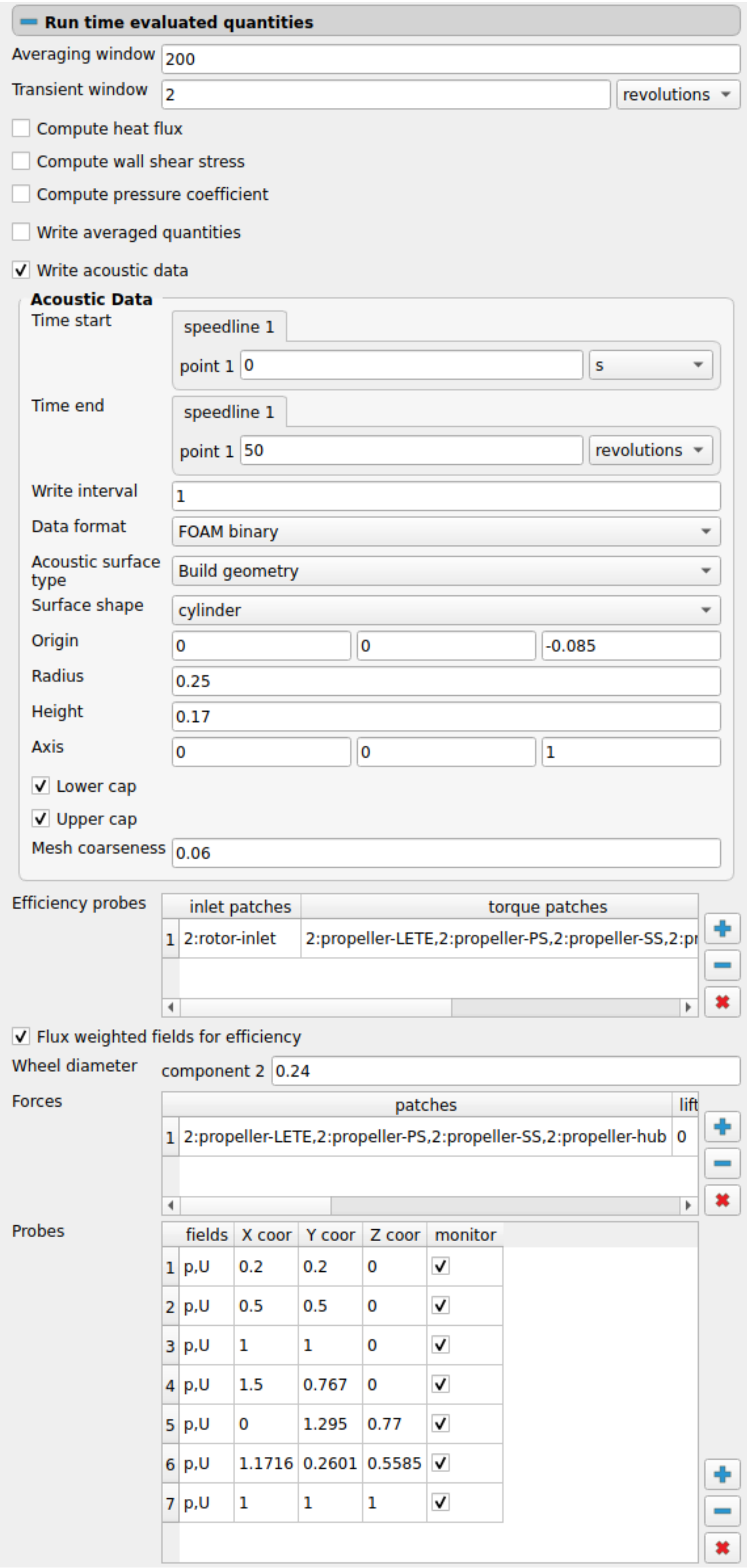
11.3.2 Run time evaluated quantities

In this section (see Figure 11.20) one can determine the quantities, and their properties, which are evaluated at run-time, i.e. each iteration. These are useful for the later post-processing purposes, and for monitoring of the convergece.

- The entry "Averaging window" sets the number of iterations used for calculation of the averaged fields. Also, this interval is used to compute the convergence status (section 11.3.3). Finally, it acts as a smoothing interval for Figures in the resulting report (see section 11.5.1).
- The entry "Transient window" is an analogue of Averaging window in a transient calculation. It can be given either in seconds, or in revolutions.
- The checkbox "Compute heat flux" controls when heat flux over walls is calculated. This is possible only for simulations that take temperature into account.
- The checkbox "Compute wall shear stress" controls when wall shear stress on walls is calculated.
- The checkbox "Write averaged quantities" controls if the averaged are saved. If this option is selected checkbox for each available quantity appears so the user can select only the quantities that are needed. For the sake of memory saving it is required not to save unused quantities.

  The averaging is based on the "Averaging window" or, in the transient case, on the "Transient window". The averaged quantities are saved together with the original ones.

  Note that some averaged quantities might be required for computation of forces exerted on the boundary patches. This is the case when the user selects the "Write surface quantities checkbox" and subsequently "Use averaged quantities" (see Section 11.5.4) or when the averaged quantities are demanded from TFEA (see Section 18.1.1). The force related quantities are force, pressure and, if present in the chosen turbulence model, eddy viscosity with turbulence kinetic energy.
- The checkbox "Write acoustic data" controls when data needed for coustic simulation are written. This is possible only for transient simulations with fixed time step size. On the chosen surface velocity, pressure and density are stored in specific time instants. On top of that also face area and normal are stored too. This data are meant to be read and further processed by TCAA module, see the corresponding pages of this manual for details.
- In the box "Acoustic data" settings for writing acoustic data is located. Namely, "Time start" and "Time end" control in which time the evaluation of data for acoustics starts and ends. "Write interval" then specifies how often the data are being written and "Data format" controls the format (we recommend "FOAM binary"). Then in "Aoucstic surface type" the surface where these quantities are evaluated is specified. You can use "External STL", "CFD patches" or build a new geometry here ("Build geometry"). Furteher you either select CFD patches in "Patches", specify the geometry you want to build or do



Run time evaluated quantities

Averaging window: 200

Transient window: 2 revolutions

□ Compute heat flux

□ Compute wall shear stress

□ Compute pressure coefficient

□ Write averaged quantities

☑ Write acoustic data

Acoustic Data

Time start: speedline 1, point 1: 0 s

Time end: speedline 1, point 1: 50 revolutions

Write interval: 1

Data format: FOAM binary

Acoustic surface type: Build geometry

Surface shape: cylinder

Origin: 0 0 -0.085

Radius: 0.25

Height: 0.17

Axis: 0 0 1

☑ Lower cap

☑ Upper cap

Mesh coarseness: 0.06

Efficiency probes

	inlet patches	torque patches
1	2:rotor-inlet	2:propeller-LETE,2:propeller-PS,2:propeller-SS,2:pr

☑ Flux weighted fields for efficiency

Wheel diameter: component 2 0.24

Forces

	patches	lift
1	2:propeller-LETE,2:propeller-PS,2:propeller-SS,2:propeller-hub	0

Probes

	fields	X coor	Y coor	Z coor	monitor
1	p,U	0.2	0.2	0	☑
2	p,U	0.5	0.5	0	☑
3	p,U	1	1	0	☑
4	p,U	1.5	0.767	0	☑
5	p,U	0	1.295	0.77	☑
6	p,U	1.1716	0.2601	0.5585	☑
7	p,U	1	1	1	☑

Figure 11.20: TCFD – SIMULATION: Run-time evaluated quantities.

nothing (if "External STL" is used). Apart from the geometrical choices, the last entry is "Mesh coarseness" that controls coarseness of the surface that is being created. The same value controls coarseness of the surface described by STL file if the checkbox "Remesh" is selected.

- The table "Efficiency probes" maintains the evaluation of the efficiency of the simulated machine, as well as many other integral parameters, which are printed in the HTML report after the calculation. The table has by default a single generic row, which corresponds to the default report being generated. Every row of the table defines inlet, torque and outlet patches. It is possible to add more rows to this table with different contents, so that more reports will be generated for the chosen evaluation method. The inlet and outlet patches are used to calculate mass flow and total pressure difference, the torque patches are used to calculate the torque expended by the fluid/blade (depending on the machine). Together, they are used to evaluate efficiency and other quantities (for efficiency formulas, see section 14.3). The switch "monitor" makes all the evaluated quantities at the inlet/outlet patches available for the Quantity monitor (see section 4.4.4).
- The switch "Flux weighted fields for efficiency" determines whether the quantities used to calculate efficiency are calculated from area-averaged or mass-flow-averaged quantities.
- The table "Forces" contains the settings for evaluation of the forces and force coefficients.

patches	defines boundary parts for evaluation
liftX,*Y,*Z	lift direction ($\vec{L}$) for the lift coefficient evaluation (C_l)
dragX,*Y,*Z	drag direction ($\vec{D}$) for the drag coefficient evaluation (C_d)
CofRX,*Y,*Z	center of rotation for the momentum coefficient evaluation
pitchX,*Y,*Z	pitch axis ($\vec{P}$) for the momentum coefficient evaluation (C_m)
torqueX,*Y,*Z	custom axis for torque evaluation (T_axis column in output file)
ref. area	reference area (A_{ref}) for lift and drag coefficient evaluation
ref. length	reference length (l_{ref}) for momentum coefficient evaluation
ref. Umag	reference velocity ($\lvert U_{\mathrm{ref}}\rvert$) for force coefficient evaluation
monitor	whether to include calculated quantities in the Quantity monitor (see section 4.4.4)

The force coefficients are evaluated as follows:

$$
\begin{aligned}
C_l &= \frac{\vec{F}_t \cdot \vec{L}}{\frac{1}{2}\rho_{\mathrm{ref}}|U_{\mathrm{ref}}|^2 A_{\mathrm{ref}}} \\
C_d &= \frac{\vec{F}_t \cdot \vec{D}}{\frac{1}{2}\rho_{\mathrm{ref}}|U_{\mathrm{ref}}|^2 A_{\mathrm{ref}}} \\
C_m &= \frac{\vec{M}_t \cdot \vec{P}}{\frac{1}{2}\rho_{\mathrm{ref}}|U_{\mathrm{ref}}|^2 l_{\mathrm{ref}} A_{\mathrm{ref}}} ,
\end{aligned}
\tag{11.11}
$$

where $\vec{F}_t$ denotes the total force and $\vec{M}_t$ the total moment. For incompressible physical model the reference pressure ρ_{ref} is taken from the value defined in Physics menu, see section 11.2.2.

Besides the values described above, the output additionally includes force components in x, y, z direction as well as torque components for x, y, z axes.

- The table "Probes" is used for probing fields in the computational domain. User can add arbitrary number of probes. Quantities, which will be probed, are selected in column "fields". Columns "X coor", "Y coor" and "Z coor" determine coordinates of the probe. Results are written to files in the directory `case_name/postProcessing/probe-%` (see TCFD directory structure 10.1). The switch "monitor" makes all the evaluated fields available for the Quantity monitor (see section 4.4.4).

- The checkbox "Compute pressure coefficient" (figure 11.21) enables calculation of the pressure coefficient C_p (mainly for the purposes of the external aerodynamics). The C_p is evaluated at given patches during the calculation for each iteration/time step and is written to the results, so it is possible to view the field in the TCAEManager (see 4.4.1).

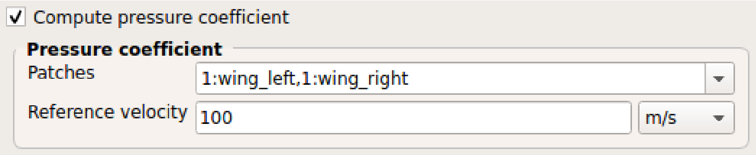
  The pressure coefficient is given by the relation

$$C_p = \frac{p - p_\infty}{/}\frac{1}{2}\rho_\infty v_\infty^2 \tag{11.12}$$

  where

  - p is static pressure
  - p_∞ is freestream pressure, which is given by the value of the Reference pressure ()
  - ρ_∞ is freestream density, which is given by the value of the Reference density
  - v_∞ is freestream velocity, which is set in the entry "Reference velocity".

  User can select the patches, on which C_p will be evaluated by the "Patches" combo box (only the wall patches are available).



☑ Compute pressure coefficient

Pressure coefficient

Patches: 1:wing_left,1:wing_right

Reference velocity: 100 m/s

Figure 11.21: TCFD – SIMULATION: Run time evaluated quantities. Pressure coefficient.

11.3.3 Convergence check

TCFD offers the possibility to monitor the convergence of the calculation, and to terminate the calculation of each particular point, when it converges to save computational resources. There might be an arbitrary number of "Convergence checks", which are shown in the table of the same name (see Figure 11.22). Each Convergence check monitors the convergence of one particular quantity, and if there are more than one checks present, the calculation will be terminated, when all of the quantities are converged.

The convergence checks are triggered once per minute during the calculation.

By clicking on the ✚, ━, ✖ buttons the individual checks can be added or deleted.

Each Convergence check has these options:

- "Monitored probe" depends on the number of Efficiency and Forces probes specified in the preceding section (11.3.2), and it determines the source of the data, that will be feed into the Convergence check.

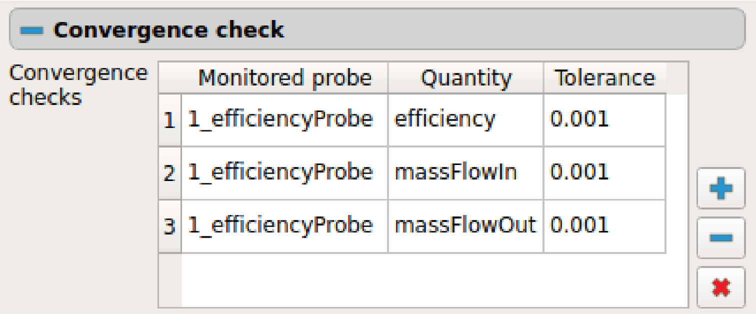
- "Quantity" sets one specific quantity, which convergence will be monitored. The available options are controlled by the value of Monitored probe, so the appropriate quantities are shown.

- "Tolerance" sets the tolerance parameter (ξ) of the Convergence check, which is used to obtain the convergence status in the equation below.

Each quantity is converged, if the the following criterion holds:

$$\max_i |\phi_i - \langle\phi\rangle| \leq \xi\langle\phi\rangle, \tag{11.13}$$

where ϕ is the selected quantity evaluated in the i-th iteration, ξ is the convergence tolerance, the angle brackets denote the average over the last Averaging window iterations (or over Transient window time interval in transient simulations), and the maximal value is taken from the same time window.



Convergence checks

	Monitored probe	Quantity	Tolerance
1	1_efficiencyProbe	efficiency	0.001
2	1_efficiencyProbe	massFlowIn	0.001
3	1_efficiencyProbe	massFlowOut	0.001

Figure 11.22: TCFD – SIMULATION: Convergence check.

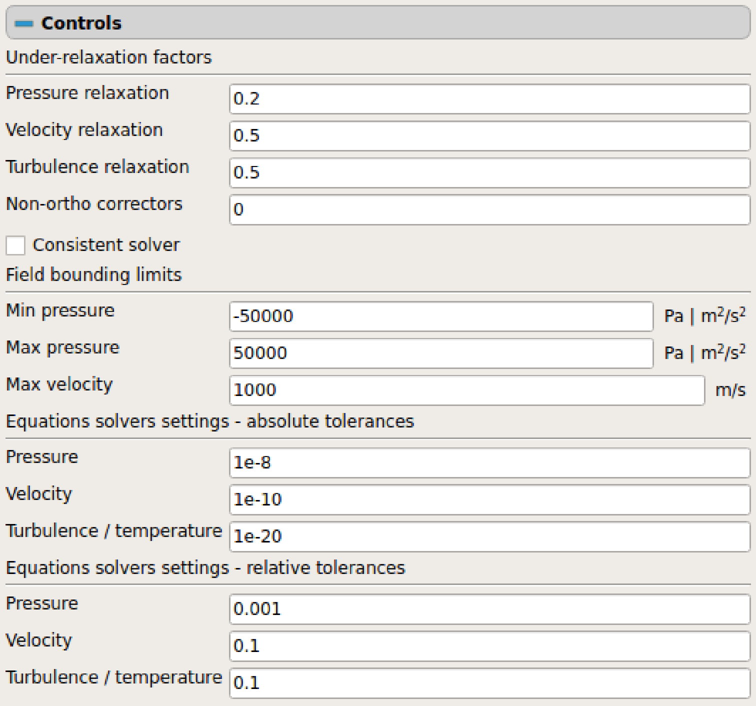
11.3.4 Controls

In this section one can adjust various parameters of the solver for fine tuning of the calculation (figure 11.23).

These options are available:

- Under-relaxation factors (see section 14.4.6) can be changed in this section. Depending on the selection of physical model, the section offers modification of some or all of the following factors: pressure, velocity, density, temperature and turbulence. The defaults are concisely summarized in the table 11.2.

- The entry "Non-ortho correctors" (advanced) sets the number of non-orthogonal correctors. Should be increased up to 3 if simulation shows large time step continuity error.



Controls

Under-relaxation factors

Pressure relaxation	0.2	
Velocity relaxation	0.5	
Turbulence relaxation	0.5	
Non-ortho correctors	0	

□ Consistent solver

Field bounding limits

Min pressure	-50000	Pa | m^2/s^2
Max pressure	50000	Pa | m^2/s^2
Max velocity	1000	m/s

Equations solvers settings - absolute tolerances

Pressure	1e-8
Velocity	1e-10
Turbulence / temperature	1e-20

Equations solvers settings - relative tolerances

Pressure	0.001
Velocity	0.1
Turbulence / temperature	0.1

Figure 11.23: TCFD – SIMULATION: Controls. Incompressible steady-state setup.

- The checkbox "Consistent solver" (advanced) enables SIMPLEC (SIMPLE-Consistent) algorithm. For some application it allows to set relaxation factors to higher values. It is highly experimental choice, but for some problems (steady flows with low turbulence and/or coarse meshes and first-order schemes of accuracy) it allows to set relaxations up to 1.0.

- Field bounding limits section defines lower and upper bounds for solving fields. If there is a cell with a lower value then the lower bound (or higher than the upper bound) after a solver iteration, it will be adjusted to this value. This bounding can help in the beginning of the solution process, when the fields wildly oscillate iteration-to-iteration before stabilization. The minimal and maximal field bounds are always given in numerical units (SI). The unit of pressure is Pa for compressible/heat transfer physical model and $\mathrm{Pa}/(\mathrm{kg}/\mathrm{m}^3) = \mathrm{m}^2/\mathrm{s}^2$ (see section 14.1.1 about kinematic pressure) for the incompressible one, in both cases without contribution of the reference pressure.

  - The entry "Minimal pressure" sets the lower bound for the pressure.
  - The entry "Maximal pressure" sets the upper bound for the pressure.
  - The entry "Maximal velocity" sets the upper bound for the magnitude of the velocity. Larger vectors are scaled to this magnitude.
  - The entry "Minimal temperature" sets the lower bound for the temperature. This is only available for the setup with temperature.
  - The entry "Maximal temperature" sets the upper bound for the temperature. This is only available for the setup with temperature.

<table>
<tr><th>physical model</th><th>incompressible</th><th colspan="2">heat transfer</th><th colspan="2">compressible</th></tr>
<tr><th>equation of state</th><td>—</td><td>Boussinesq</td><td colspan="3">perfect gas</td></tr>
<tr><th>transonic</th><td>—</td><td>—</td><td>—</td><td>no</td><td>yes</td></tr>
<tr><th>pressure</th><td>0.2</td><td>0.5</td><td colspan="2">0.3</td><td>0.8</td></tr>
<tr><th>velocity</th><td>0.5</td><td>0.3</td><td colspan="2">0.7</td><td>0.2</td></tr>
<tr><th>density</th><td>—</td><td>0.1</td><td colspan="2">0.01</td><td>1.0</td></tr>
<tr><th>temperature</th><td>—</td><td>0.2</td><td colspan="2">0.7</td><td>0.2</td></tr>
<tr><th>turbulence</th><td>0.5</td><td>0.3</td><td colspan="2">0.8</td><td>0.2</td></tr>
</table>

Table 11.2: TCFD – default relaxation factors for all possible setups. The compressible subsonic setup is used for fans, the compressible transonic setup is used for compressors and compressible turbines.

  - The entry "Minimal density" sets the lower bound for the density. This is only available for the setup with temperature.
  - The entry "Maximal density" sets the upper bound for the density. This is only available for the setup with temperature.

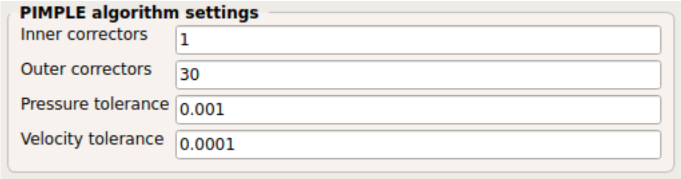
- If the calculation is transient, another box "PIMPLE algorithm settings" (figure 11.24) appears at the bottom of the section, and contains the the adjustments of the PIMPLE algorithm parameters:

  - The entry "Inner correctors" sets a number of iterations for correcting pressure field without re-calculation of a momentum matrix.
  - The entry "Outer correctors" sets a number of iterations for re-calculation of the pressure-momentum coupling within one time step. The outer correctors loop basically corresponds to SIMPLE (or SIMPLEC if "Consistent solver" is checked) algorithm loop within one time step. Value 1 enables PISO mode, whereas higher numbers set the maximum number of SIMPLE iterations to be performed if the tolerances below are not satisfied.
  - "Pressure tolerance"
  - "Velocity tolerance"

  The outer loop is stopped before the maximum number of iterations is reached, if both pressure and velocity residuals drops below the tolerance thresholds.

11.3.5 Scripting

For experienced users it is possible to extend the TCFD workflow by custom scripts (`advanced`) (see figure 11.25). These scripts are expected to be written in basic Python 3.7 and they are executed at specific moments during the workflow. These "Execution points" are:



PIMPLE algorithm settings

Inner correctors: 1

Outer correctors: 30

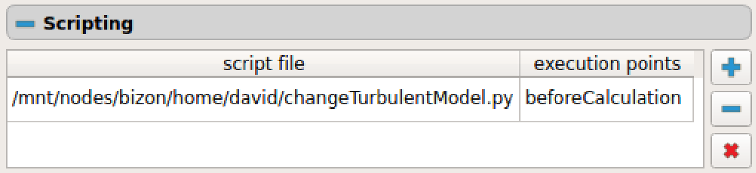
Pressure tolerance: 0.001

Velocity tolerance: 0.0001

Figure 11.24: TCFD – PIMPLE algorithm settings.

- "afterWrite"
- "beforeCalculation"
- "afterCalculation"
- "beforeEverySpeedline"
- "afterEverySpeedline"
- "beforeTransient"
- "beforeReport"
- "afterReport"

Their meaning is obvious. It is allowed to assign multiple execution points to a single script.



Scripting

script file	execution points
/mnt/nodes/bizon/home/david/changeTurbulentModel.py	beforeCalculation

Figure 11.25: TCFD – SIMULATION: Scripting.

The script can use the predefined variable CaseDirectory, which contain the full absolute path to the TCAE case directory (in the figure 10.1 it is the top level directory tcaecase). Besides the standard Python functions one can also use the TCAE-specific functions `SetEntry` and `WriteFile`. For example the following tiny script

```
SetEntry("TCFD/system/fvSolution", "solvers/p/nCellsInCoarsestLevel", "10")
WriteFile("TCFD/system/fvSolution")
```

will change the coarsest-level cell number in pressure GAMG solver to 10, followed by writing the modified file. The function `SetEntry` does all modifications in memory and the result is written to disk only when the function `WriteFile` is used, or during writing requested by the workflow. Apart from the two them, there are also two another access functions

```
RenameEntry(<file>, <entry>, <newname>)
DeleteEntry(<file>, <entry>)
```

with obvious purpose: The former changes the name of an entry (i.e. the keyword that introduces the entry), whereas the latter erases the whole entry from the file.

Scripting Examples

Scripting in TCFD requires an advanced knowledge of OpenFOAM case and dictionaries structure as well as the TCFD case structure (table 10.1). Additionaly, it allows to run any command during the TCFD workflow which gives the user a high variability of customization.

Changing Turbulence Model

TCFD offers three turbulence models from $k - \varepsilon$ family of turbulence models, namely kEpsilon, realizableKE and RNGkEpsilon. With TCFD, user can use other turbulence models which OpenFOAM implements as well. This example shows how to employ LaunderSharmaKE low Reynolds turbulence model by a scripting framework.

Let's have a look into a case written by TCAE for a standard kEpsilon turbulence model, particularly into the file: `tcaecase0\TCFD\constant\turbulenceModel`:

```
simulationType RAS;

RAS
{
    RASModel kEpsilon;
    turbulence on;
    printCoeffs on;
    kEpsilonCoeffs
    {
        Cmu 0.09;
        C1 1.44;
        C2 1.92;
        C3 -0.33;
        sigmak 1;
        sigmaEps 1.3;
    }
}
```

To use a different turbulence model, this file have to be modified accordingly. In this case, the `RASModel` has to be changed to `LaunderSharmaKE`. User can do this by creating the following script

```
SetEntry("constant/turbulenceModel", "RAS/RASModel", "LaunderSharmaKE")
WriteFile("constant/turbulenceModel")
```

and executing it at afterWrite, see Figure 11.25.

It is always important to modify or add all the parameters and files compulsory for a particular setup.

User cannot change kEpsilon model to v2f turbulence model just by changing the `RASModel` parameter. v2f is a 4-equation model requiring additional fields to be defined (2v,f) which is also possible to implement by a scripting framework.

Running Application

Within the scripting framework, it is possible to execute any command, e.g., OpenFOAM application. As an example, the user wants to automatically evaluate `vorticity` and `Q` (Q-criterion) field after the simulation is done. In a standard OpenFOAM it can be done by a command:

```
» postProcess -funcs "(vorticity Q)"
```

The script should be executed at beforeReport when all the calculations and tasks are done. The script is printed below:

```
import os

# PATH to TCAE in WIN:

tcaeInstDir = 'C:/TCAE/21.09'

# IMPORTANT: in WIN, new environmental variable has to be defined.
if os.name == 'nt':
    os.environ['OPENFOAM_INSTALL_PATH']= tcaeInstDir + '/cygwin64/opt/OpenFOAM/OpenFOAM-dev'
```

```
10
11 postProcessExec = 'postProcess'
12 if os.name == 'nt':
13     postProcessExec = tcaeInstDir + '/cygwin64/opt/OpenFOAM/OpenFOAM-dev/platforms/cygwin64mingw-
         w64DPInt32Opt/bin/postProcess.exe'
14
15 myCmd = postProcessExec + ' -funcs "(vorticity Q)" -case ' + CaseDirectory.decode('ascii') + ' > ' +
      CaseDirectory.decode('ascii') + '/log.vorticityAndQ'
16
17 os.system(myCmd)
```

1 : loads 'Miscellaneous operating system interfaces' python module needed for `os.name` and `os.system` functions

4 : defines installation directory for Windows version of TCAE

7-8: in Windows, a new environmental variable has to be defined to run OpenFOAM binaries directly (i.e., out of cygwin environment)

10: definition of executable in Linux

11-12: definition of executable in Windows

14: definition of the command string to be executed; the output is re-directed into the file inside the case directory

16: executing the command

11.4 BOUNDARY CONDITIONS

Thi section deals with setting of the boundary, interface and initial conditions and is therefore divided into following submenus (fig. 11.26):

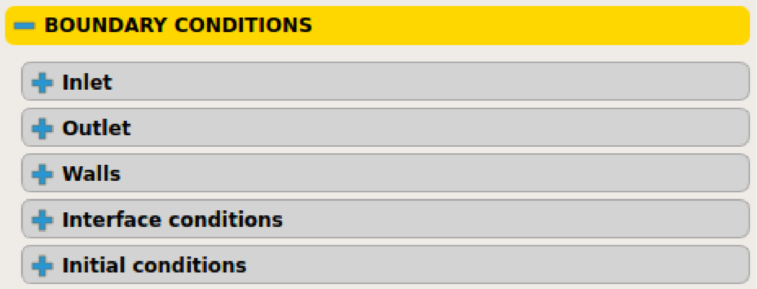
- "Inlet boundary conditions"
- "Outlet boundary conditions"
- "Wall boundary conditions"
- "Interface conditions"
- "Initial conditions"

In TCFD, each Inlet, Outlet and Wall patch has its own boundary condition. If user loads a geometry in TMESH (see section 7.2.2) and assigns individual patches to the components, the patches appear automatically in one of the submenus in this section.

Based on the type of the patch selected in the Patches table (see 7.2.2), each patch is shown here in the correct subsection (Inlet, Outlet, Wall, Interface).

For better clarity during the boundary conditions setup it is possible to highlight the patch in *RenderView*, which boundary condition is being adjusted. This works in all of the following sections (Inlet / Outlet / Walls / Interface).

Each patch has small "flashlight" icon, by default "turned off". If user *double*-clicks on the tab bar of some patch (i.e. on the icon or on the patch name), the patch is highlighted in *RenderView* and the flashlight icon "turns on" (see Fig. 11.27). Next double-click disables the highlighting.



BOUNDARY CONDITIONS

Inlet

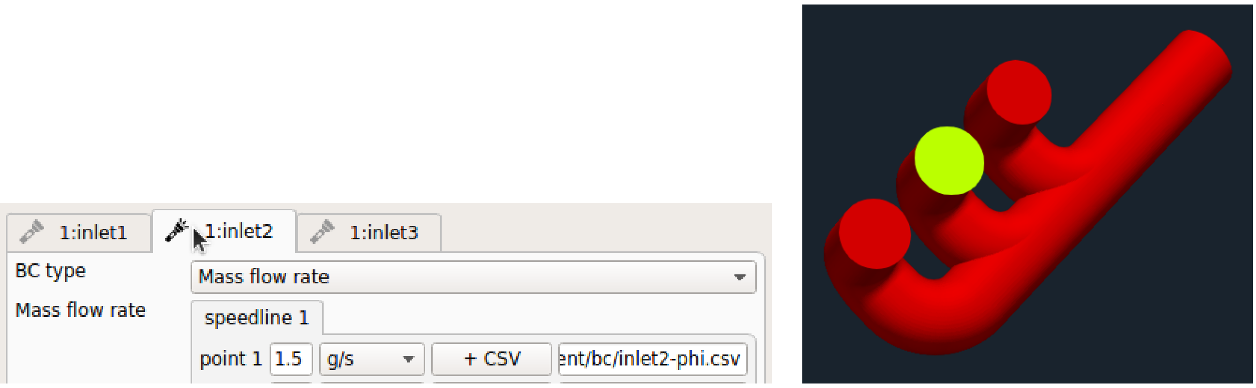
Outlet

Walls

Interface conditions

Initial conditions

Figure 11.26: TCFD – BOUNDARY CONDITIONS.



BC type: Mass flow rate

Mass flow rate: speedline 1, point 1 1.5 g/s + CSV ent/bc/inlet2-phi.csv

(a) Selected patch `1:inlet2`

(b) Highlighted patch `1:inlet2` in *RenderView*

Figure 11.27: TCFD – BOUNDARY CONDITIONS: Patch highlighting

11.4.1 Inlet

In this subsection the boundary condition for Inlet patches are set up.

First of all, there is one option "Turbulent inlet quantities", which is common for all inlet boundary conditions. It defines, which turbulent quantities will be entered later by the user in each inlet turbulent boundary condition. The values are:

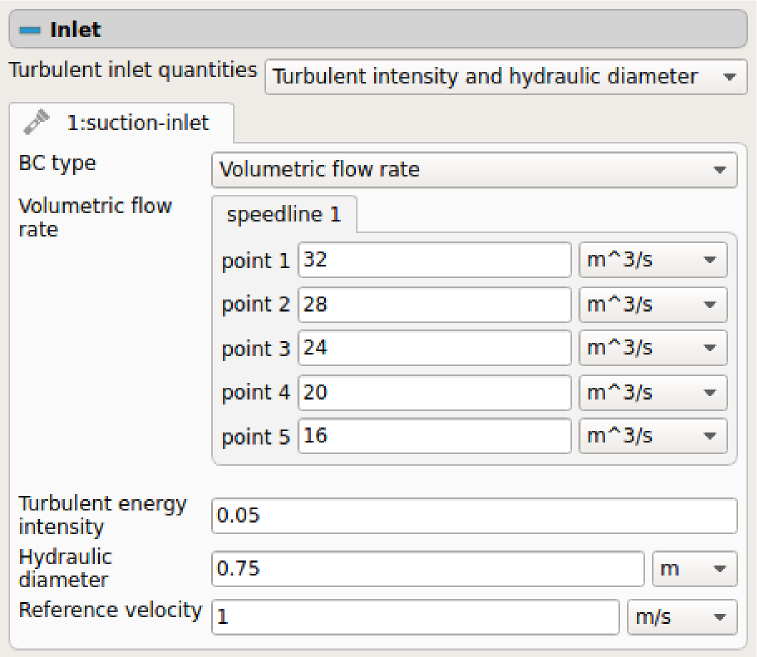
- "Turbulent intensity and hydraulic diameter" – Turbulent energy intensity, hydraulic diameter and reference velocity will be prescribed.
- "Turbulent intensity and length scale" – Turbulent energy intensity, turbulent length scale and reference velocity will be prescribed.
- "Turbulent intensity and viscosity ratio" – Turbulent energy intensity, turbulent viscosity ratio and reference velocity will be prescribed.
- "Turbulent viscosity ratio" (Available only for models SpalartAllmaras and SpalartAllmaras-DDES) – Turbulent viscosity ratio will be prescribed.
- "Model quantities" – the quantities of the selected turbulent model will be prescribed, that means:
  - turbulent kinetic energy k for all of the kOmegaSST* and kEpsilon* models
  - turbulent dissipation rate ϵ for all of the kEpsilon* models
  - turbulent specific dissipation rate ω for all of the kOmegaSST* models
  - transition momentum thickness Reylonds number $Re_{\theta t}$ for the kOmegaSSTLM model

If there is no turbulence, i.e. the Turbulence model is laminar, above mentioned entry is hidden, as well as any other parameters of the turbulent boundary conditions.

Then, each Inlet patch has its own tab with settings, which has title in the form `<component-index>:<patch-name>`.

The choice of the type of the inlet boundary condition for each particular patch is done by selection of one item of the drop-down list "BC type". By changing its value the appropriate entries appear in the menu. For many of the quantities one can prescribe different values for each point in each speedline, while some have just one fixed value for whole calculation. The types of the inlet boundary conditions are:

- Mass flow rate.
  Values of mass flow rate are prescribed.
- Directed mass flow rate
  Directed variant of Mass flow rate type, see below.
- Volumetric flow rate.
  Values of volumetric flow rate are prescribed.
- Directed volumetric flow rate
  Directed variant of Volumetric flow rate type, see below.



Inlet

Turbulent inlet quantities: Turbulent intensity and hydraulic diameter

1:suction-inlet

BC type: Volumetric flow rate

Volumetric flow rate: speedline 1

point 1	32	m^3/s
point 2	28	m^3/s
point 3	24	m^3/s
point 4	20	m^3/s
point 5	16	m^3/s

Turbulent energy intensity: 0.05

Hydraulic diameter: 0.75 m

Reference velocity: 1 m/s

Figure 11.28: TCFD – BOUNDARY CONDITIONS: Inlet. Example of Volumetric flow rate boundary condition.

- Total pressure
  Values of inlet total pressure p_{tot} are prescribed.
  - Non-uniform inlet velocity
    This option enables non-uniform distribution of inlet velocity vectors magnitudes (direction is always normal to the inlet patch). Typically, it is applicable to inlet components where the flow direction near the inlet does not follow the inlet patch normal direction.
- Fixed velocity
  Components of the velocity vector are prescribed (in m/s).
- Velocity profile
  This boundary condition is useful to represent boundary layer profile for wind tunnel and wind turbine simulations. Two entries are available: "Velocity profile CSV file" and "Velocity profile direction". The vector determines the line, along which the profile is evaluated. The CSV file shall contain four columns - first column specifies the distance from the point $[0, 0, 0]$ along the mentioned line, while other three columns store components of velocity vector:

```
        -47.5,     0    , 0, 0
        -47.25,  -12.04, 0, 0
        -47,     -13.30, 0, 0
        -46,     -15.56, 0, 0
        -45,     -16.74, 0, 0
        -40,     -19.59, 0, 0
        -30,     -22.11, 0, 0
        -20,     -23.59, 0, 0
        -10,     -24.66, 0, 0
        0,     −25.51, 0, 0
        10,     −26.21, 0, 0
```

The visualization of the velocity profile in the case of boundary layer is shown in the Fig. 11.29.

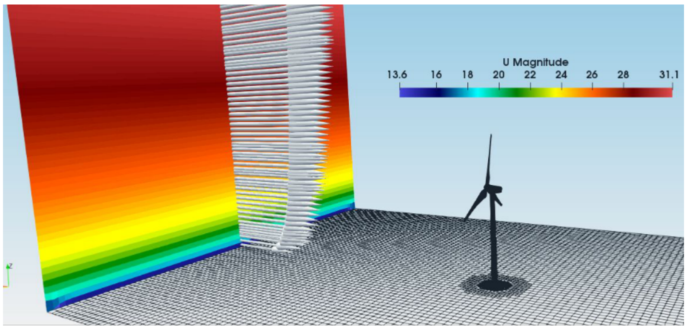


Figure 11.29: TCFD – Velocity profile visualization.

- Fan pressure

  This boundary condition simulate the fan behaviour, such as there were fan at inlet. This condition sets total pressure to

$$p_{\mathrm{tot,BC}} = p_0 + \Delta p_{\mathrm{tot}}. \tag{11.14}$$

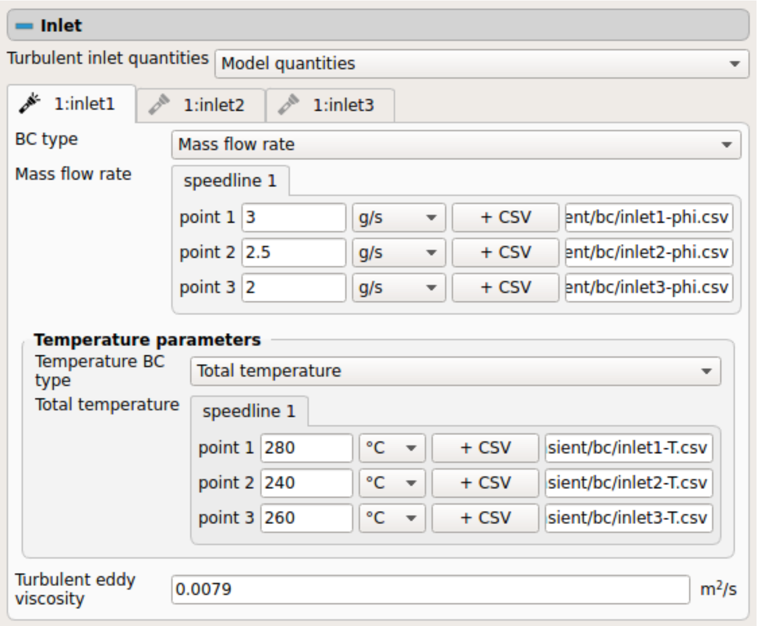
  Base pressure p_0 is set by entry "Fan pressure p0". Total pressure increase Δp_{tot} depend on the volumetric flow rate. This dependency (fan characteristic) is contained in a text file, which is loaded using the entry "Fan pressure file" and must have the OpenFOAM list syntax:

```
        9
        (
        (1.4    2000)
        (1.6    1900)
        (1.8    1750)
        (2.0    1580)
        (2.2    1400)
        (2.4    1200)
        (2.6    960)
        (2.8    720)
        (3.0    450)
        )
```

  First line defines the number of lines, while each row contains volumetric flow rate $[m^3/s]$ in the first column and total pressure differences Δp_{tot} in the second. The Δp_{tot} has units $[Pa]$ for Compressible or Heat transfer physical models, and $[m^2 s^{-2}]$ for Incompressible model (see section 14.1.1 about kinematic pressure). This boundary condition is available only for stator machine type.

- Opening

  This boundary condition shall be selected if the flow direction is unknown. The total pressure (along with total temperature for compressible cases) and freestream values of turbulent quantities must be set, therefore appropriate entries in the menu appear.



Inlet

Turbulent inlet quantities: Model quantities

BC type: Mass flow rate

Mass flow rate

speedline 1

point 1	3	g/s	+ CSV	ent/bc/inlet1-phi.csv
point 2	2.5	g/s	+ CSV	ent/bc/inlet2-phi.csv
point 3	2	g/s	+ CSV	ent/bc/inlet3-phi.csv

Temperature parameters

Temperature BC type: Total temperature

Total temperature

speedline 1

point 1	280	°C	+ CSV	sient/bc/inlet1-T.csv
point 2	240	°C	+ CSV	sient/bc/inlet2-T.csv
point 3	260	°C	+ CSV	sient/bc/inlet3-T.csv

Turbulent eddy viscosity: 0.0079 m^2/s

Figure 11.30: TCFD – BOUNDARY CONDITIONS: Inlet. Example of Mass flow rate boundary condition with additional CSV files with transient values and temperature boundary condition.

- Freestream

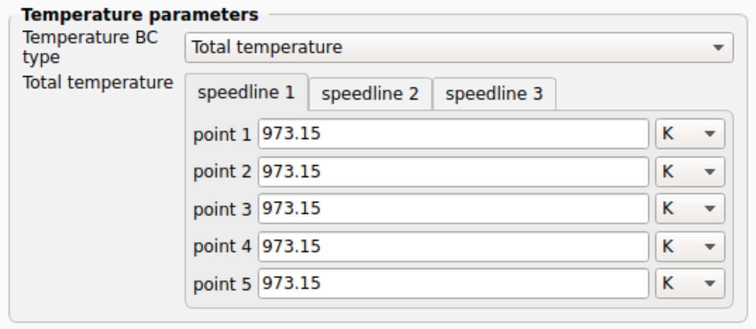
Directed boundary conditions ("Directed mass flow rate" and "Directed volumetric flow rate") allow specification of the angle of the velocity vectors at inlet. This is done by setting the additional fields "Meridional angle" and "Circumferential angle". The inlet direction is a vector that points typically outside from the geometry and the velocity vectors will be oriented to be antiparallel with the inlet direction vectors (i.e. to point inwards). The Meridional angle is the angle between the inlet direction vectors and the direction of the rotation axis. Allowed values of the Meridional angle are between 0 and 180 degrees. The Circumferential angle of the inlet direction vector at a given face is the angle between the inlet direction vector and the plane formed by the axis and the position vector of the given face. The range of the Circumferential angle is from -90 degrees to +90 degrees, where positive angles correspond to positive orientation with respect to the rotation axis (right hand rule). For exact geometric meaning of the angles see section 15.1.

If Compressible or Heat transfer physical model is selected, additional parameters, which control the boundary condition for temperature are shown in each tab (see Fig. 11.31).

Switch "Temperature BC type" changes the type of boundary condition for temperature. Avialable options are "Total temperature" and "Fixed temperature". The first alternative is enabled only for compressible calculations (perfectGas equation of state is chosen). Then, the Total or Fixed temperature is set up for each speedline and point.

Turbulent boundary condition

If the Turbulence model isn't laminar, several parameters of the turbulence model appear. Based on the value of Turbulent inlet quantities, described above, proper parameters appear in each



Temperature parameters

Temperature BC type: Total temperature

Total temperature

speedline 1 | speedline 2 | speedline 3

point 1 973.15 K

point 2 973.15 K

point 3 973.15 K

point 4 973.15 K

point 5 973.15 K

Figure 11.31: TCFD – Temperature boundary condition.

inlet boundary condition box. The turbulent model specific parameters are then computed from them. Following options might be present, along with the contitions, that have to be fulfilled to make the particular option visible and usable:

- "Turbulent kinetic energy" (k)
  – Turbulent inlet quantities = Model quantities
  – Turbulence model = kOmegaSST / kOmegaSSTLM /kEpsilon / RNGkEpsilon / RealizableKE

- "Turbulent dissipation rate" (ϵ)
  – Turbulent inlet quantities = Model quantities
  – Turbulence model = kEpsilon / RNGkEpsilon / RealizableKE

- "Turbulent specific dissipation rate" (ω)
  – Turbulent inlet quantities = Model quantities
  – Turbulence model = kOmegaSST / kOmegaSSTLM

- "Turbulent eddy viscosity" $(\tilde{\nu})$
  – Turbulent inlet quantities = Model quantities
  – Turbulence model = SpalartAllmaras / SpalartAllmarasDDES

- "ReThetaT" $(Re_{\theta t})$
  – Turbulent inlet quantities = Model quantities
  – Turbulence model = kOmegaSSTLM

- "Turbulent energy intensity" (I)
  – Turbulent inlet quantities = Turbulent intensity and length scale / Turbulent intensity and hydraulic diameter / Turbulent intensity and viscosity ratio

- "Turbulent length scale" (l)
  – Turbulent inlet quantities = Turbulent intensity and length scale

- "Turbulent viscosity ratio" ()
  – Turbulent inlet quantities = Turbulent intensity and viscosity ratio

- "Hydraulic diameter" (L)
  – Turbulent inlet quantities = Turbulent intensity and hydraulic diameter

- "Reference velocity" (U_{ref})
  – Turbulent inlet quantities = Turbulent intensity and length scale / Turbulent intensity and hydraulic diameter / Turbulent intensity and viscosity ratio

If the Turbulent inlet quantities isn't Model quantities, the actual model quantities are then computed by the TCFD using the relations given in the section 15.1.

If there are passive scalars added to the simulation as described in the chapter PHYSICS : Multiphysics, the user can specify Inlet values of scalars.

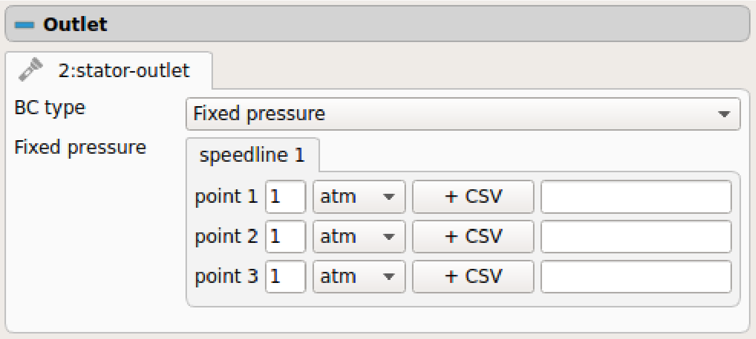
If there is particle simulation added to the simulation as described in the chapter PHYSICS : Multiphysics, the user can specify Particle injection rate.

When the calculation is set up as transient *and* the advanced mode is switched on, then some of the boundary condition parameters accept a CSV file with time-dependent values of the parameter. The first column of the CSV file must contain the times, the second column must contain the values of the parameter (e.g. mass flow or total pressure), in the same units as chosen for the steady-state initial calculation. Clicking on "+ CSV" button will open a file selection dialog; alternatively, the path to the file can be put by hand into the edit field next to the button. When no time-dependent CSV data are specified, then the transient calculation will use the same value for the boundary condition parameter as the stationary calculation. If the calculation is set to run longer than what times are available in the CSV file, it will use the last available value of the parameter once the maximal time defined in CSV is exceeded.

Examples of inlet boundary conditions settings are displayed in Figures 11.28 and 11.30.

11.4.2 Outlet

In this subsection the boundary condition for Outlet patches are set up.



Outlet

2:stator-outlet

BC type: Fixed pressure

Fixed pressure: speedline 1

point 1 1 atm + CSV

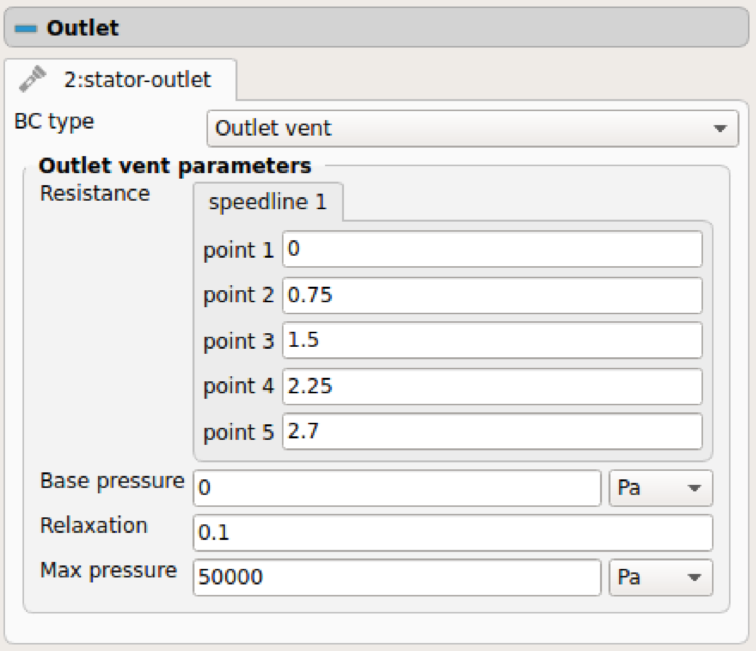
point 2 1 atm + CSV

point 3 1 atm + CSV

Figure 11.32: TCFD – BOUNDARY CONDITIONS: Outlet. Example of Fixed pressure boundary condition.

Each Outlet patch has its own tab with settings, which has title in the form `<component-index>:<patch-name>`.

The choice of the type of the inlet boundary condition for each particular patch is done by selection of one item of the drop-down list "BC type":



Outlet

2:stator-outlet

BC type: Outlet vent

Outlet vent parameters

Resistance: speedline 1

point 1: 0

point 2: 0.75

point 3: 1.5

point 4: 2.25

point 5: 2.7

Base pressure: 0 Pa

Relaxation: 0.1

Max pressure: 50000 Pa

Figure 11.33: TCFD – BOUDNARY CONDITIONS: Outlet. Example of Outlet vent boundary condition.

- Fixed pressure
  Uniform value of static pressure is prescribed.

- Fixed mean pressure
  This BC differs from the Fixed pressure in that it imposes the average value of the static pressure, i.e., the particular distribution is dependent on the flow at the boundary patch. A comparison of Fixed pressure and Fixed mean pressure boundary conditions for axial fan geometry is presented in Figure 11.34.

- Outlet vent
  This boundary condition adjusts the pressure based on the velocity:

$$p_{BC} = p_0 + \frac{1}{2}\rho R|\mathbf{u}|^2, \tag{11.15}$$

  where R is the so called *resistance*. Density and velocity fields are taken from the solver ($\rho = 1$ for incompressible physical model). The parameters are

  - "Resistance", which specifies the outlet vent resistance R; it can contain multiple resistance if multiple points are to be computed;
  - "Base pressure", which specifies the outlet static pressure $p0$ for resistance $R = 0$;
  - "Relaxation", which specifies the relaxation of the pressure fields between the iterations of the solver;
  - "Max pressure", which sets a pressure limit.

- Mass flow rate
  Values of mass flow rate are prescribed.

- Volumetric flow rate
  Values of volumetric flow rate are prescribed.

- Fixed velocity
  Components of the velocity vector are prescribed (in m/s).

- Fan pressure
  This boundary condition works the same way as its counterpart for the inlet boundary condition, with one difference: the pressure on outlet is prescribed as $p_{tot,BC} = p_0 - \Delta p_{tot}$, so the actual total pressure is lower than base pressure p_0.

- Opening
  This boundary condition shall be selected if the flow direction is unknown. The Total pressure (along with Total temperature for compressible cases) and freestream values of turbulent quantities (depends on the Turbulence model – Turbulent kinetic energy, Turbulent dissipation rate, Turbulent specific dissipation rate, Turbulent eddy viscosity, ReThetaT) must be set, therefore appropriate entries in the menu appear.

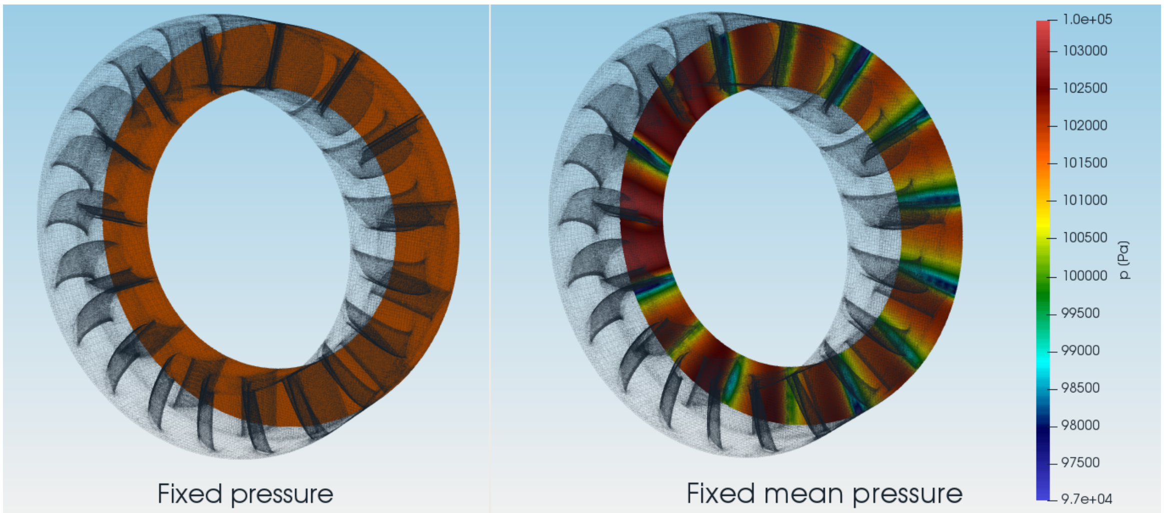


Figure 11.34: TCFD – Outlet boundary conditions: Fixed pressure vs Fixed mean pressure.

The boundary conditions Mass flow rate, Volumetric flow rate and Fixed velocity can be set for:

- Total pressure boundary condition at the inlet

- One-component mesh or frozen rotor setup, i.e., number of there is no Interface condition with type MixingPlane (see section 11.4.4).

Just as in the case of Inlet boundary conditions, when the calculation is set up as transient *and* the advanced mode is switched on, then some of the boundary condition parameters accept a CSV file with time-dependent values of the parameter. The first column of the CSV file must contain the times, the second column must contain the values of the parameter (e.g. mass flow or total pressure), in the same units as chosen for the steady-state initial calculation. Clicking on "+ CSV" button will open a file selection dialog; alternatively, the path to the file can be put by hand into the edit field next to the button. When no time-dependent CSV data are specified, then the transient calculation will use the same value for the boundary condition parameter as the stationary calculation. If the calculation is set to run longer than what times are available in the CSV file, it will use the last available value of the parameter once the maximal time defined in CSV is exceeded.

Examples of outlet boundary conditions settings are displayed in Figures 11.32 and 11.33.

11.4.3 Walls

TCFD assigns wall boundary condition to each patch of types wall, hub, shroud, cutWater and all blade*, defined in the section Components (see 7.2.2).

For better clarity, the boundary conditions are groupped by the component, in which the boundary patch is contained. Option "Patches from component" controls the active component, which patches are shown.

Each Wall patch has its own tab with settings, which has title in the form `<component-index>:<patch-name>`.

There are three types of wall boundary conditions for velocity, and the selection is done by the drop-down list "Wall type":

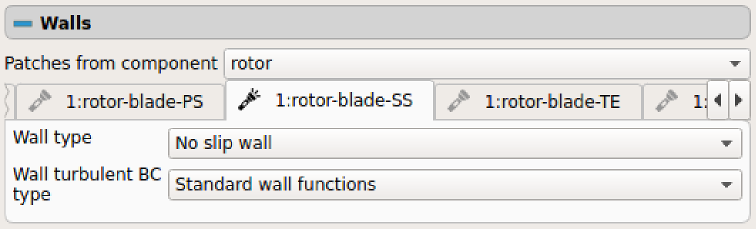
- "No slip wall" - velocity at the wall is zero
- "Slip wall" - velocity vector at the wall is parallel with the boundary
- "Fixed velocity" - the actual velocity vector is prescribed

There is another choice for the boundary condition for the turbulent quantities, which is chosen via entry "Wall turbulent BC type":

- "Standard wall functions"
- "Low-Reynolds wall functions"
- "Rough walls" If selected, another two boxes appear:
  - "Sand-grain roughness"
  - "Roughness constant"

If the temperature is computed, additional boundary condition for temperature have to be prescribed and additional section "Wall temperature parameters" is displayed. There are five types of wall temperature boundary conditions, which are selectable by the drop-down list "Temperature BC type":

- "Adiabatic wall"
  The heat flux is zero, in other words the normal derivative of the temperature is zero.



Walls

Patches from component: rotor

Wall type: No slip wall

Wall turbulent BC type: Standard wall functions

Figure 11.35: TCFD – BOUNDARY CONDITIONS: Walls. Example of boundary condition with velocity and turbulent type.

- "Fixed temperature"
  The value of temperature is prescribed. The value can be set for each point for each speedline, and for transient simulations the CSV files with time-dependent values can be used*.
- "Fixed mapped temperature" The temperature value is read from a .csv file and mapped on the CFD mesh. This is useful if your temperature is spatially nonuniform. This boundary condition is temporally constant (same for all speedlines).
- "Fixed power"
  Heating power in Watts is specified. There is only one value for every point and speedline.
- "Fixed heat flux"
  Heat flux $[Wm^{-2}]$ is prescribed. There is only one value for every point and speedline.
- "Fixed heat transfer coefficient"
  This boundary condition is intended to simulate the case, where inner computational domain is separated from the environment by solid wall. The "Ambient temperature" is specified (for each point and speedline, optionaly with CSV files for transient case*), along with "Heat transfer coefficient" at the outer surface of the wall. Wall may consist of different layers, and their properties are set in table "Wall layers". By adding and removing the rows one sets the number of layers - each row corresponds to one layer, which has its own thickness and thermal conductivity.

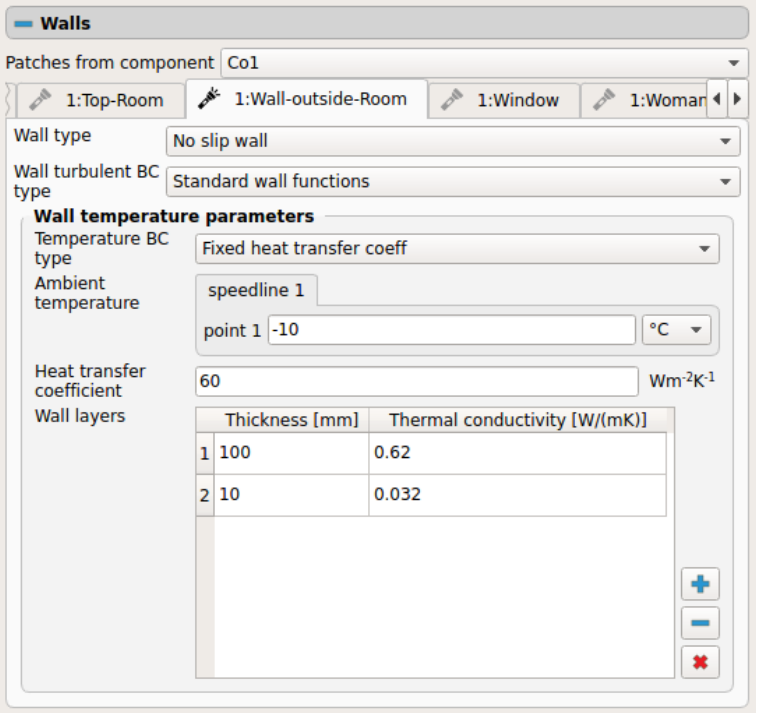
** Look at the sections above (Inlet, Outlet) for the description of adding the time-dependent values of boundary conditions via the CSV files.*

Examples of wall boundary conditions settings are displayed in Figures 11.35 and 11.36.

11.4.4 Interface conditions

In this section, the interface conditions, which are the conditions prescribed on interface patches, are set up. They affect the way the solution behaves on the boundaries between the components, where fluid travels from one component to another (typicaly between the stator and rotor in turbumachinery simulations, where fluid enters the rotating component).

An example of the interface conditions is displayed in the Fig. 11.37.



Walls

Patches from component: Co1

1:Top-Room | 1:Wall-outside-Room | 1:Window | 1:Woman

Wall type: No slip wall

Wall turbulent BC type: Standard wall functions

Wall temperature parameters

Temperature BC type: Fixed heat transfer coeff

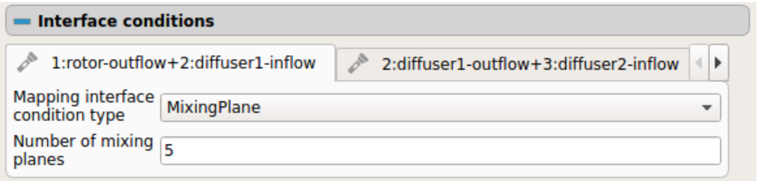
Ambient temperature: speedline 1 — point 1: -10 °C

Heat transfer coefficient: 60 $Wm^{-2}K^{-1}$

Wall layers:

	Thickness [mm]	Thermal conductivity [W/(mK)]
1	100	0.62
2	10	0.032

Figure 11.36: TCFD – BOUNDARY CONDITIONS: Walls. Example of temperature boundary condition Fixed head transfer coeff



Interface conditions

1:rotor-outflow+2:diffuser1-inflow | 2:diffuser1-outflow+3:diffuser2-inflow

Mapping interface condition type: MixingPlane

Number of mixing planes: 5

Figure 11.37: TCFD – BOUNDARY CONDITIONS: Interface conditions.

As user adds or removes inletInterface/outletInterface[1] patches in the Patches table, and connects them properly to the corresponding neighbour patches in other components (see Fig 7.2.2), the interface patches automatically appears in this section. In contrast to Inlet/Outlet/Wall Boundary conditions, these don't appear individualy, but in pairs, because each interface is made up from two interface patches, each one in separate component.

Each interface (i.e. the pair of patches) has one tab with its settings, and has title in the form `<1st-component-index>:<1st-patch-name> + <2nd-component-index>:<2nd-patch-name>`.

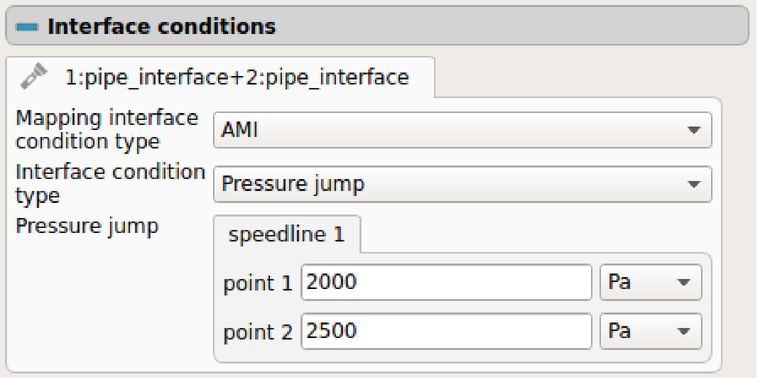
For each interface condition, the following settings are available:

- "Mapping interface condition type" sets the method of interpolating the values from one interface patch to the other. Its options are:
  - "AMI" (*Arbitrary Mesh Interface*, direct weighted interpolation) maps the physical fields directly from the first patch to the second.
  - "MixingPlane" computes the averages on the first patch (over given number of circular strips) and then maps these averages the second patch.

  For the better understanding see section 14.7.1.

- "Number of mixing planes" sets the number of mixing plane strips (circular averaging) to use.

- "Interface condition type" enables the additional condition, which affects the physical fields directly. Currently, there is only one type - Pressure jump, and is visible, only if the Mapping interface condition type is set to AMI. If chosen, the pressure of the fluid moving through the interface will rise by the desired value. The value may be set individually for each point and speedline with optional CSV files containing time-depenedent values (see Fig. 11.38). If the geometry is periodic, it is forbidden to add pressure jump condition, so if user do so, the Check Setup (4.1) will notify him/her to solve this discrepancy.


Interface conditions

1:pipe_interface+2:pipe_interface

Mapping interface condition type: AMI

Interface condition type: Pressure jump

Pressure jump

speedline 1

point 1: 2000 Pa

point 2: 2500 Pa

Figure 11.38: TCFD – Interface conditions - pressure jump.

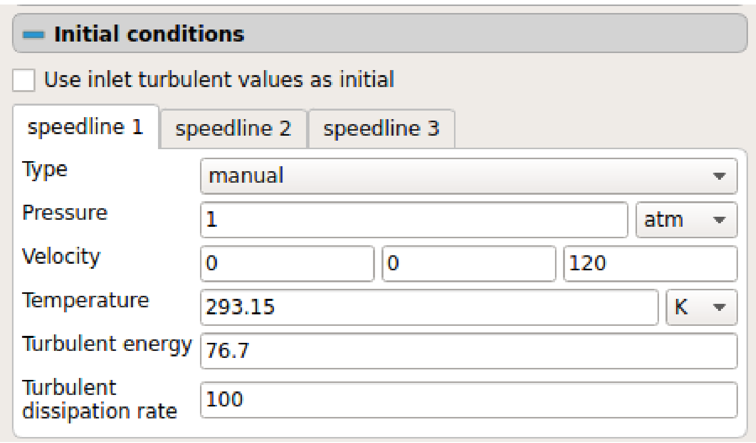
[1] freestreamInterface patches don't feature interface conditions, therefore don't appear here

11.4.5 Initial conditions

This subsection deals with setup of the initial conditions. The initial conditions are set for each speedline independently. Typical setup is displayed in Figures. 11.39 and 11.40.

By checking the "Use inlet turbulent values as initial" the values of the inlet turbulent quantities are copied from the first inlet boundary condition.

For each initial condition (e.g. for each speedline) there is first option "Type" switches between "manual" and "mapped" type of initial condition.



Initial conditions

□ Use inlet turbulent values as initial

speedline 1 | speedline 2 | speedline 3

Type: manual

Pressure: 1 atm

Velocity: 0 0 120

Temperature: 293.15 K

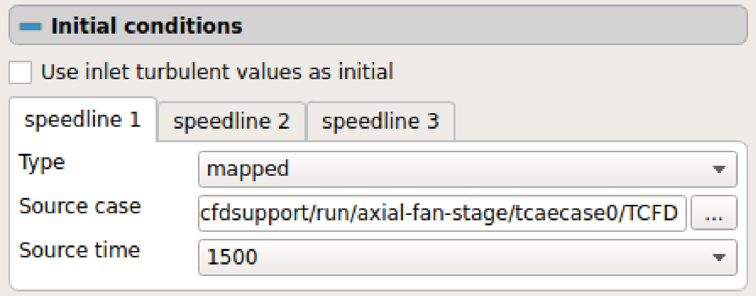
Turbulent energy: 76.7

Turbulent dissipation rate: 100

Figure 11.39: TCFD – BOUNDARY CONDITIONS: Initial conditions - manual variant.

If type is selected to manual, one has to specify uniform initial values for individual fields that will be solved for and following options are displayed (Fig. 11.39):

- The entry "Pressure" sets a constant value of initial pressure throughout the computational domain. The solver will then iteratively improve this initial estimate.
- The entry "Velocity" sets a constant value of initial velocity throughout the computational domain. It is recommended that the initial flow follows the overall expected flow direction, typically along the rotation axis.
- The entry "Temperature" sets a constant value of initial temperature (if temperature is present in the calculation) throughout the computational domain.
- The entry "Turbulent energy"* sets a constant value of initial turbulent energy k throughout the computational domain. This is only needed when using kOmegaSST, kOmegaSSTLM, kEpsilon, realizableKE, RNGkEpsilon turbulence models.
- The entry "Turbulent dissipation"* sets a constant value of initial turbulent dissipation rate ϵ throughout the computational domain. This is only needed when using kEpsilon, realizableKE, RNGkEpsilon turbulence models.
- The entry "Turbulent dissipation rate"* sets a constant value of initial turbulent dissipation rate ω throughout the computational domain. This is only needed when using kOmegaSST(LM) turbulence models.



Initial conditions

□ Use inlet turbulent values as initial

Type: mapped

Source case: cfdsupport/run/axial-fan-stage/tcaecase0/TCFD

Source time: 1500

Figure 11.40: TCFD – BOUNDARY CONDITIONS: Initial conditions - mapped variant.

- The entry "ReThetaT"* sets a constant value of initial $Re_{\theta t}$ throughout the computational domain. This is only needed when using kOmegaSSTLM turbulence model.

** Not visible if Use inlet turbulent as initial is enabled.*

On the contrary, if type is selected to mapped, the fields will be initialized by results from some other TCFD (or generally OpenFOAM) calculation. That means, that to map results from another TCAE case, user has to select directory `case_name/TCFD`. The meshes of the old and the new calculation need to be very similar and the boundary patches in both cases must have the same names. Two options are displayed (Fig. 11.40):

- The entry "Source case" is a path to a directory with a completed TCFD (or OpenFOAM) calculation to be used.
- If the selected case is a valid, the selection list "Source time" can be used to select a particular (stationary) time result directory from the chosen case.

11.5 POST-PROCESSING

This section allows user to define how the post-processing (evaluation of results and generation of a report) will be done (Fig. 11.41).

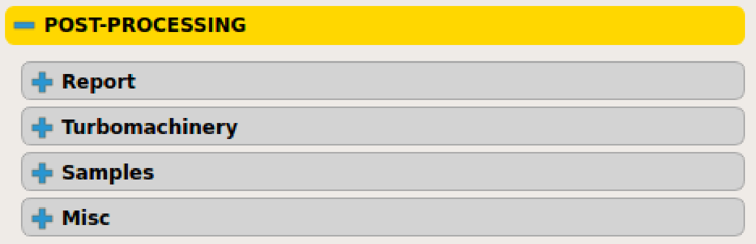
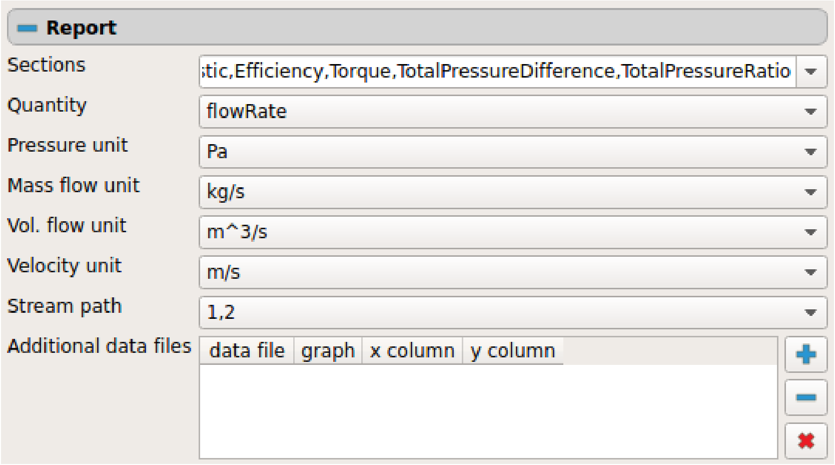


Figure 11.41: TCFD – POST-PROCESSING.

11.5.1 Report

In this section the properties of the HTML report, which is created when the calculation is finished, are to be set. The section is depicted in Fig 11.42.

- The selection "Sections" defines sections to appear in the calculation report.
- The selection "Quantity" defines the reference quantity for the calculation report.
- The selection "Pressure unit" defines the pressure unit that will be used in the calculation report.
- The selection "Temperature unit" defines the temperature unit that will be used in the calculation report.
- The selection "Mass flow unit" defines the mass flow unit that will be used in the calculation report.
- The selection "Vol. flow unit" defines the volumetric flow unit that will be used in the calculation report.
- The selection "Velocity unit" defines the velocity unit that will be used in the calculation report.
- "Wheel diameter" is a mandatory parameter only for the fan machine and is used for post-processing.
- The list "Stream path" offers the possibility to select a path though the simulated machine (in terms of component indices) that will be used when plotting the total pressure along the flow (one point per interface encountered on the way). For linear component topology with single inlet to the first component and single outlet from the last component there will be only one path and the resulting total pressure graph will simply correspond to the total pressure variations between the inlet and the outlet. But for more complicated topologies with cycles of multiple inlets/outlets there are generally more ways how to choose the path.
- The table "Additional data files" is used to add user data to the graphs in the final report from the calculation. The data should be provided in the form of Gnuplot-readable data files, i.e. text files with white-space-separated equal-length columns of numbers. The graph into which the data are to be added is chosen in the second column, the column indices to be used in the remaining two (first column has index 1). Several data-lines within a single data file can be achieved by interrupting the columns by an empty row; this is useful particularly in the multi-speedline summary diagrams (their name ends with "-all").



Report

Sections: tic,Efficiency,Torque,TotalPressureDifference,TotalPressureRatio

Quantity: flowRate

Pressure unit: Pa

Mass flow unit: kg/s

Vol. flow unit: m^3/s

Velocity unit: m/s

Stream path: 1,2

Additional data files:

data file	graph	x column	y column

Figure 11.42: TCFD – POST-PROCESSING: Report.

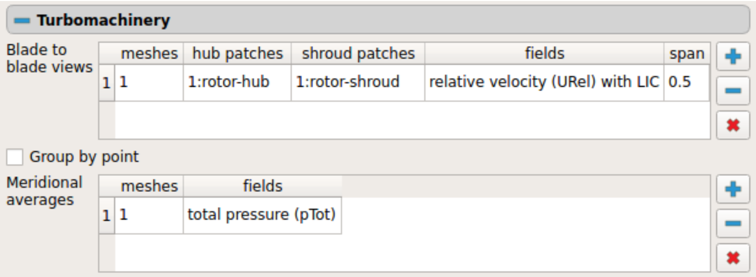
11.5.2 Turbomachinery

This subsection deals with some additional post-processing features, which are useful for turbomachinery simulations (Fig. 11.43).

- The table "Blade to blade views" contains a list of requested blade-to-blade views (circularly unwrapped and slices meshes) to be generated by Turbo Blade Post. The column "meshes" contains a list of meshes that will be unwrapped. It is possible to use both the internal meshes and individual patches. The hub and shroud patches must be provided in the next two columns as they define the requested transformation of the mesh. The column "fields" is used to select the field that will be displayed on the blade-to-blade view. Finally, "span" contains a space separated list of positions between the hub and the shroud where the blade-to-blade views will be taken.
- The switch "Group by point" reorders the blade-to-blade views in the report so that figures corresponding to the same calculation point are grouped together; otherwise figures corresponding to the same span are grouped together. The former option is good for visualisation of how the field changes in space, whereas the latter option is good for visualisation of how the field changes with the boundary condition.
- The table "Meridional averages" contains a list of meridional averages to be generated by Turbo Blade Post. It can be used only with component indices, but otherwise the meaning of its columns is identical to Blade to blade views.

11.5.3 Samples

This section (Fig. 11.44) contains just the table "Surface samples" , which controls sampling of the pressure (and temperature, if present) fields on selected patches for the latter use in FEM analysis in some third-party SW. The patches are selected in column "patches" (the wall and interface patches are available), while in column "fields" the sampled fields are selected. Results are written in raw format and stored in `case_name/TCFD/postProcessing/surfaceSample-%`. (see TCFD directory structure 10.1).



Turbomachinery

Blade to blade views

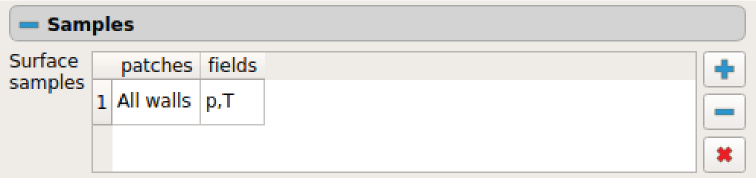
	meshes	hub patches	shroud patches	fields	span
1	1	1:rotor-hub	1:rotor-shroud	relative velocity (URel) with LIC	0.5

☐ Group by point

Meridional averages

	meshes	fields
1	1	total pressure (pTot)

Figure 11.43: TCFD – POST-PROCESSING: Turbomachinery.



Samples

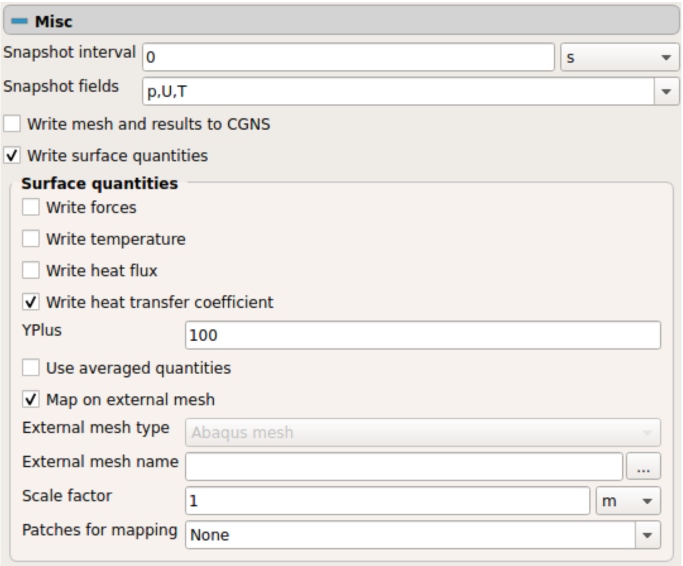
Surface samples

	patches	fields
1	All walls	p,T

Figure 11.44: TCFD – POST-PROCESSING: Samples.

11.5.4 Misc

Here some another postprocessing tools can be enabled (Fig. 11.45).



Misc

Snapshot interval: 0 s

Snapshot fields: p,U,T

☐ Write mesh and results to CGNS

☑ Write surface quantities

Surface quantities

☐ Write forces

☐ Write temperature

☐ Write heat flux

☑ Write heat transfer coefficient

YPlus: 100

☐ Use averaged quantities

☑ Map on external mesh

External mesh type: Abaqus mesh

External mesh name:

Scale factor: 1 m

Patches for mapping: None

Figure 11.45: TCFD – POST-PROCESSING: Misc.

- The entry "Snapshot interval" has only effect in a transient calculation. It serves for regular write-out of the results during the calculation, so that they can be later used e.g. to construct an animation.

- The element "Snapshot fields" can be used to select fields that will be written during a snapshot. Quantities other than velocity and pressure are rarely needed.

- If the switch "Write mesh and results to CGNS" is enabled, the mesh and results is written on the disk in *CGNS* format. Stationary results are saved in file `case_name/TCFD/case_stationary.cgns` and the transient ones in file `case_name/TCFD/case_transient.cgns` (see TCFD directory structure 10.1).

- The checkbox "Write surface quantities" gives another way of exporting the data for the FEM analysis in some third-party SW. In contrast to Surface samples (11.5.3), the forces exerted by the fluid on wall patches are calculated, and then (with temperature, if present) written in the files in directory `tcae_case/TCFD/postProcessing/surfaceQuantities/`, where there is one file `surfaceQuantitie` with results for each selected patch and each solution point. This feature currently works in the stationary calculation.

- The chekbox "Write forces" lets the user choose saving the forces exerted by the fluid onthe boundary patches.

- "Use relative pressure" is available only if checkbox "Write forces" is enabled. It determines, whether to compute forces (for Write surface quantities) from the absolute, or the relative pressure. The latter option comes beneficial in the case, when the geometry is "one-sided", i.e. the geometry is surrounded by the fluid on just one side. Considering absolute pressure would lead to unphysical values of forces, because there is no pressure from the surroundings of the machine.

- The checkbox "Write temperature" gives the user the possibility to save the temperature field on boundary patches. This checkbox is selectable only if thermal simulation is being performed.

- The chekbox "Write heat flux" gives the possibility to calculate and store the heat flux over wall patches. In the same time, power (heat flux integrated over the patch) is computed too. This checkbox is selectable only if thermal simulation is being performed. The power values are stored on the path `case_name/TCFD/postProcessing/wallHeatFlux1/0/wallHeatFlux.dat`.

- If the checkbox "Write heat transfer coefficient" is selected, the heat transfer coefficient is computed, as well as temperature in choosen *yPlus* - this is set in the entry "YPlus". After the succesful computation, both values are written to the `surfaceQuantities.csv` file. The heat transfer coefficient calculation is performed in post processing and requires some quantities to load. If you choose the surface data to be time-averaged ("Use averaged quantities" is on), you need to save these. They are namely *Velocity*, *Temperature*, *Density*, *Eddy viscosity*, *Dynamic viscosity*, *Thermal diffusivity*. Of course, this checkbox is selectable only if the simulation takes into account the thermal effects.

- The entry "YPlus" is related to the "Write heat transfer coefficient" checkbox. It says in which *yPlus* the temperature is evaluated. This checkbox is selectable only if thermal simulation is being performed.

- "Use averaged quantities" determines whether we use averaged or instantaneous values for calculating and saving the "Write surface quantities". In the case of instanteous values it can happen that the current iteration oscilates and the values are of. This danger can be minimized by the usage of averaged quantities. However, to be able to use the averaged quantities, the user needs to save them (see "Write averaged quantities" in Section 11.3.2).

For the force calculation velocity, pressure and, if present in the chosen turbulence model, eddy viscosity with turbulence kinetic energy. So if the user chooses to use averaged quantities all these quantities (and temperature) need to be saved.

- The option "Map on external mesh" allows you to map the surface quantities specified by the options above on an external mesh. (So far only temperature and heat transfer coefficient are possible to map). The resulting files (for each quantity is created a separate file) are stored on the path `TCFD/postProcess/extractedData/<time>/`. The temperature file is called `temperature.nam` and the file containing heat transfer coefficient and corresponding reference temperature is called `htc.nam`.
- The entry "External mesh type" what the formata of the mesh on which we are about to map is. The only supported format is currently *Abaqus mesh*. This entry shows up only if "Map on external mesh" is selected.
- "External mesh name" specifies the path to the external mesh. This entry shows up only if "Map on external mesh" is selected.
- There is also an option "Scale factor" that will scale the external mesh before the mapping. The mapped values are stored for the original unscaled mesh.
- The entry "Patches for mapping" requires the user to select the patches that are used for the mapping on external mesh. These usually are the patches on the interface between the two meshes. This entry shows up only if "Map on external mesh" is selected.

Chapter 12

TCFD – Configuration File Options

This section contains the list of all keywords of the TCFD module, that might appear in the *.tcae file. General format of the *.tcae file is described in section 3.3.

Keyword ▶ ***Description***	Allowed / sample values	Units	Mandatory
type ▶ *Machine class, one of: compressor, fan, propeller, pump, stator, turbine, hydroTurbine, windTurbine, virtualTunnel, closedDomain.*	fan	–	yes
TCFD-numberOfProcessors ▶ *Number of CPU cores used for simulation. Default = 1. If processors > 1, then a domain-decomposition solution with MPI communication is used.*	4	–	no
solverSteadyState ▶ *Custom solver for steady state calculation.*	blueSolver	–	no
solverTransient ▶ *Custom solver for transient calculation.*	blueDyMSolver	–	no
bindToCore ▶ *Lock processes to cores to prevent their migration. This is currently active only in Linux.*	true	–	no
TCFD-hosts ▶ *List of remote machines for scheduling parallel processes. Paswordless login must be available. Currently active only in Linux.*	node1 node2 node3	–	no
transient ▶ *Append transient simulation after stationary. Default is “no”. The semi-transient modes “semiAMI” and “semiMXP” use MRF method instead of physical rotation, albeit with time derivatives.*	no | yes | semiAMI | semiMXP	–	no
timeStep ▶ *Time step options (transient simulations only). Default is “adaptive”.*	adaptive | constant	–	no
deltaT ▶ *Value for timestep constant*	1_s	–	no
Co ▶ *Courant number (transient simulations only).*	0.9	–	no
fluidName ▶ *Name of the fluid. Used for various material properties defaults.*	water | air | custom	–	yes
physicalModel ▶ *Physical model of the fluid.*	incompressible | compressible | heatTransfer	–	yes
equationOfState ▶ *Equation of state for compressible and heatTransfer physical models.*	perfectGas | Boussinesq | PengRobinson	–	no

Keyword ▶ ***Description***	Allowed / sample values	Units	Mandatory
PengRobinson-Tc	132.5203	K	no
▶ *Critical temparature T_c for Peng-Robinson equation of state.*			
PengRobinson-Vc	0.084	dm^3/mol	no
▶ *Critical molar volume V_c for Peng-Robinson equation of state.*			
PengRobinson-Pc	3.7860e+06	Pa	no
▶ *Critical pressure p_c for Peng-Robinson equation of state*			
PengRobinson-omega	0.0335	–	no
▶ *Acentric factor ω for Peng-Robinson equation of state.*			
Boussinesq-beta	0.003	–	yes*
▶ *Volumetric thermal expansion coefficient for Boussinesq equation of state.*			
compressibility	incompressible | compressible	–	no
▶ *Deprecated. Fluid flow nature: compressible or incompressible.*			
referenceDensity	1.2	kg/m^3	yes
▶ *Reference density of the fluid. Selectable unit.*			
dynamicViscosity	0.000018	$Pa \cdot s$	yes
▶ *Dynamic viscosity of the fluid. Selectable unit.*			
gravitationalAccelerationDirection	0 0 0	–	no
▶ *Gravity direction in Cartesian coordinates (x y z components).*			
gravitationalAccelerationMagnitude	9.81	m/s^2	no
▶ *Gravity magnitude.*			
referencePressure	101325	Pa	yes
▶ *Reference pressure. All other pressures will be considered relative to this one. Selectable unit.*			
referencePressurePoint	no	–	no
▶ *Reference pressure point for incompressible calculations (the equations involve pressure only in derivatives, so it needs to be fixed by value taken from this point)*			
referenceTemperature	293.15	K	yes*
▶ *Reference temperature. *) Only used in compressible calculations. Selectable unit.*			
molarWeight	28.9	kg/mol	no

Keyword ▶ ***Description***	**Allowed / sample values**	**Units**	**Mandatory**
▶ *Molar weight (air = 28.9, water = 18.015).*			
heatCapacityRatio	1.4	–	no
▶ *Heat capacity ratio Cp/Cv for "totalPressure", "opening" and "totalTemperature" boundary conditions.*			
heatCapacityModel	constant	–	no
▶ *Select heat capacity model. Currently only option "constant" is available.*			
Cp	1004	$\mathrm{J/(kg \cdot K)}$	no
▶ *Speficic heat capacity for "constant" model.*			
transport	sutherland | constant	–	no
▶ *Viscous transport model.*			
Pr	0.7	–	no
▶ *Constant transport parameter (Prandl number, default air: 0.7, water: 7)*			
As	1.512e-06	–	no
▶ *Sutherland transport parameter.*			
Ts	120	K	no
▶ *Sutherland transport parameter.*			
numberOfPorosityZones	120	K	no
▶ *Number of the porosity zones.*			
N_porosityZone-name	porosity_zone_1	K	no
▶ *Name of the porosity zone.*			
N_porosityZone-component	1	–	no
▶ *Index of the corresponding component where the porosity will be applied.*			
N_porosityZone-d	5e+07 3e+10 3e+10	–	no
▶ *Darcy-Forchheimer vector D for the Nth porosity zone.*			
N_porosityZone-f	500 100000 100000	–	no
▶ *Darcy-ForchHeimer vector F for the Nth porosity zone.*			
N_porosityZone-e1	0.998 0.061 0	–	no
▶ *Local directional vector E1 for the Nth porosity zone.*			
N_porosityZone-e2	0.016 -1 0	–	no

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
▶ *Local directional vector E2 for the Nth porosity zone.*			
cavitationRisk	false	–	no
▶ *Whether to evaluate cavitation properties.*			
multiphaseCavitation-use	yes	–	no
▶ *Wheter to include multiphase cavitation in calculation. Only for water machines.*			
multiphaseCavitation-model	SchnerrSauer	–	no
▶ *Which model of multiphase cavitation to use. Only for water machines.*			
multiphaseCavitation-pSat	2300	Pa	no
▶ *Saturation pressure for multiphase cavitation.*			
multiphaseCavitation-sigma	0.075	kg/s^2	no
▶ *Surface tension for multiphase cavitation.*			
multiphaseCavitation-vapourRho	0.02308	kg/m^3	no
▶ *Vapour density for multiphase cavitation.*			
multiphaseCavitation-vapourNu	4.273e-04	m^2/s	no
▶ *Vapour kinematic viscosity for multiphase cavitation.*			
multiphaseCavitation-SchnerrSauer-n	1.6e+13	1/m^3	no
▶ *Bubble number density (parameter of Schnerr-Sauer multiphase cavitation model).*			
multiphaseCavitation-SchnerrSauer-dNuc	2.0e-06	m	no
▶ *Nucleation site diameter (parameter of Schnerr-Sauer multiphase cavitation model).*			
multiphaseCavitation-SchnerrSauer-Cc	1	–	no
▶ *Condensation rate coefficient (parameter of Schnerr-Sauer multiphase cavitation model).*			
multiphaseCavitation-SchnerrSauer-Cv	1	–	no
▶ *Vapourisation rate coefficient (parameter of Schnerr-Sauer multiphase cavitation model).*			
calcAge	false	–	no
▶ *Whether to calculate age of the fluid.*			
calcComfort	false	–	no
▶ *Whether to calculate comfort levels PMV and PPD.*			
comfort-clothing	0.5	–	no

Keyword ▶ ***Description***	**Allowed / sample values**	**Units**	**Mandatory**
▶ *Parameter to calculate comfort levels*			
comfort-externalWork	0	W/m^2	no
▶ *Parameter to calculate comfort levels*			
comfort-metabolicRate	1.2	W/m^2	no
▶ *Parameter to calculate comfort levels*			
comfort-relativeHumidity	60	%	no
▶ *Parameter to calculate comfort levels*			
numberOfPassiveScalars	2	–	no
▶ *How many passive scalars to add to the calculation.*			
N_passiveScalar-name	oxygen	–	no
▶ *Name of the Nth passive scalar.*			
N_passiveScalar-diffusivityType	constant/turbulent	–	no
▶ *Nth passive scalar: Whether to specify constant value of diffusivity or calculate it from dynamic and turbulent viscosity.*			
N_passiveScalar-alphaD	1	–	no
▶ *Nth passive scalar: Parameter for turbulent diffusivity calculation.*			
N_passiveScalar-alphaDt	1	–	no
▶ *Nth passive scalar: Parameter for turbulent diffusivity calculation.*			
N_passiveScalar-diffusivity	1	–	no
▶ *Nth passive scalar: Constant diffusivity.*			
numberOfSpeedlines	3	–	yes
▶ *Number of speedlines (different rotation speeds).*			
N_numberOfPoints	3	–	yes
▶ *Number of points for the Nth speedline. All speedline points share rotational speed.*			
N_iterations	1000 1000 1000	–	yes
▶ *Number of iterations for each point of the Nth speedline.*			
N_transientTimes	1_s 1_s 1_revolution	–	no
▶ *How much time to compute in transient simulation in respective points of Nth speedline. Choice of unit.*			
pMin	-2000	Pa (m^2/s^2)	no

Keyword ▶ ***Description***	Allowed / sample values	Units	Mandatory
▶ *Bounding value for pressure for robust convergence. Pascals assumed in compressible case, kinematic pressure in incompressible case.*			
pMax	5000	Pa (m^2/s^2)	no
▶ *Bounding value for pressure for robust convergence. Pascals assumed in compressible case, kinematic pressure in incompressible case.*			
UMax	1000	m/s	no
▶ *Bounding value for robust convergence.*			
rhoMin	0.1	kg/m^3	no
▶ *Bounding value for robust convergence.*			
rhoMax	20	kg/m^3	no
▶ *Bounding value for robust convergence.*			
TMin	273.15	K	no
▶ *Bounding value for robust convergence.*			
TMax	1000	K	no
▶ *Bounding value for robust convergence.*			
p_relax	0.2	–	no
▶ *Pressure under-relaxation factor.*			
U_relax	0.5	–	no
▶ *Velocity under-relaxation factor.*			
rho_relax	1	–	no
▶ *Density under-relaxation factor.*			
t_relax	0.2	–	no
▶ *Turbulence under-relaxation factor.*			
T_relax	0.2	–	no
▶ *Temperature under-relaxation factor.*			
X_tolerance	1E-8	–	no
▶ *Linear solver tolerance for quantity $X = rho, p, U, tt$ (tt includes temperature and turbulent quantities)*			
X_finalTolerance	1E-8	–	no
▶ *Linear solver tolerance for quantity $X = rho, p, U, tt$ (tt includes temperature and turbulent quantities)*			

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
X_relTol	1E-3	–	no
▶ *Linear solver relative tolerance of quantity X = rho, p, U, tt (tt includes temperature and turbulent quantities)*			
nonOrthoCorrectors	0	–	no
▶ *Non-orthogonality-compensating nested pressure iterations.*			
SIMPLE-consistent	false	–	no
▶ *Enables SIMPLEC algorithm. Default is SIMPLE.*			
PIMPLE-nCorrectors	1	–	no
▶ *Number of inner corrector iterations for PIMPLE algorithm.*			
PIMPLE-nOuterCorrectors	100	–	no
▶ *Maximum number of outer corrector iterations (SIMPLE/C) within a single time iteration.*			
PIMPLE-UTolerance	1e-5	–	no
▶ *Convergence tolerance for velocity components initial residuals during the outer corrector loop.*			
PIMPLE-pTolerance	0.001	–	no
▶ *Convergence tolerance for pressure initial residuals during the outer corrector loop.*			
numericalOrder	first	–	no
▶ *Either "first" or "second" numerical order.*			
transonic	1	–	no
▶ *Transonic solver mode "1" or subsonic mode "0".*			
averagingWindow	100	–	no
▶ *Averaging window for stationary calculation. Does the averages over last n iterations. Checks convergence for last n iterations.*			
transientWindow	0.1_revolutions	– (s)	no
▶ *Averaging window for transient calculation. Use the suffix _revolutions or _seconds to determine units for the time interval.*			
averagedQuantities-save	yes	–	no
▶ *If set to yes, we are you can select quantities that will be averaged overtime and saved.*			
averagedQuantities-U	yes	–	no
▶ *Averaged velocity will be saved.*			
averagedQuantities-p	yes	–	no

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
▶ *Averaged pressure will be saved.*			
averagedQuantities-mu	yes	–	no
▶ *Averaged dynamic viscosity will be saved.*			
averagedQuantities-alpha	yes	–	no
▶ *Averaged thermal diffusivity will be saved.*			
averagedQuantities-rho	yes	–	no
▶ *Averaged density will be saved.*			
averagedQuantities-nut	yes	–	no
▶ *Averaged eddy viscosity will be saved.*			
averagedQuantities-k	yes	–	no
▶ *Averaged turbulence kinetic energy will be saved.*			
averagedQuantities-omega	yes	–	no
▶ *Averaged specific rate of dissipation will be saved.*			
averagedQuantities-ReThetat	yes	–	no
▶ *Averaged transition momentum thickness Reynolds number will be saved.*			
averagedQuantities-gammaInt	yes	–	no
▶ *Averaged intermittency will be saved.*			
averagedQuantities-epsilon	yes	–	no
▶ *Averaged rate of dissipation of turbulent kinetic energy will be saved.*			
averagedQuantities-nuTilda	yes	–	no
▶ *Averaged nuTilda from Spalart-Allmaras turbulence model will be saved.*			
averagedQuantities-T	yes	–	no
▶ *Averaged temperature will be saved.*			
snapshotInterval	0	s	no
▶ *Secondary write for transient calculation.*			
snapshotFields	p U	–	no
▶ *Which fields to write during snapshots.*			
reportSections	Efficiency Troque ForceCeoffs	–	no

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
▶ *Which section should report include.*			
convergenceCheck	true	–	no
▶ *Monitor convergence and auto-skip to next point when convergence has been reached.*			
convergenceCheck-tolerance	0.001	–	no
▶ *Relative convergence threshold for efficiency and mass flow.*			
userDefinedFunctions-TCFD	scripts/run.py afterWrite	–	no
▶ *Custom user script setup.*			
N_referenceFrame-angularVelocity	200_RPM 300_RPM 400_RPM	–	no
▶ *Definition of the Nth reference frame, angular velocity for all speedlines, either rad/s or RPM*			
N_referenceFrame-rotating	1	–	no
▶ *Definition of the Nth reference frame, switch rotation on (0) or off (1).*			
N_*patchX*-boundaryCondition-type	massFlowRate	–	yes
▶ *Boundary condition to use for patch* patchX *in the Nth component. For inlet patch the possible options are: massFlowRate, directedMassFlowRate, volumetricFlowRate, directedVolumetricFlowRate, totalPressure, fixedVelocity, velocityProfile, fanPressure, opening, freestream. For outlet patch: fixedPressure, fixedMeanPressure, outletVent, volumetricFlowRate, massFlowRate, fixedVelocity, fanPressure, opening, freestream. For wall patch: noSlip, slip, fixedVelocity.*			
N_*patchX*-boundaryCondition-M_massFlowRate	0.126 0.124 0.120	kg/s	yes*
▶ *Patch* patchX, Nth *component: Boundary condition values for all points of the Mth speedline of type "massFlowRate". Selectable unit.*			
N_*patchX*-boundaryCondition-M_massFlowRate-K_csv	bc/mphi.csv	–	no
▶ *Patch* patchX, Nth *component: Transient boundary condition data for Mth speedline, Kth point.*			
N_*patchX*-boundaryCondition-M_volumetricFlowRate	0.126 0.124 0.120	m^3/s	yes*
▶ *As above, for BC of type "volumetricFlowRate". Used for incompressible calculations. Selectable unit.*			
N_*patchX*-boundaryCondition-M_volumetricFlowRate-K_csv	bc/phi.csv	–	no
▶ *Patch* patchX, Nth *component: Transient boundary condition data for the Mth speedline, Kth point.*			
N_*patchX*-boundaryCondition-M_totalPressure	200000 150000 100000	Pa	yes*
▶ *As above, for BC of type "totalPressure". Selectable unit.*			
N_*patchX*-boundaryCondition-M_totalPressure-K_csv	bc/pTot.csv	–	no
▶ *Patch* patchX, Nth *component: Transient boundary condition data for th Mth speedline, Kth point.*			

Keyword ▶ ***Description***	**Allowed / sample values**	**Units**	**Mandatory**
N_*patchX*-boundaryCondition-M_fixedVelocity	0 0 1 0 0 2 0 0 3	m/s	yes*
▶ *As above, for BC of type "FixedVelocity".*			
N_*patchX*-boundaryCondition-M_fixedVelocity-K_csv	bc/U.csv	–	no
▶ *Patch* patchX, Nth *component: Transient boundary condition data for the Mth speedline, Kth point.*			
N_*patchX*-boundaryCondition-velocityProfileFile	Uprofile.csv	–	yes*
▶ *Patch* patchX, Nth *component: CSV file with velocity profile values for BC of type "velocityProfile". Selectable unit.*			
N_*patchX*-boundaryCondition-velocityProfileDirection	0 0 1	–	yes*
▶ *Patch* patchX, Nth *component: Direction vector, that determines the line, along which is velocity profile evaluated for BC of type "velocityProfile".*			
N_*patchX*-boundaryCondition-fanPressureP0	100000_Pa	–	yes*
▶ *Patch* patchX, Nth *component: Total pressure value for BC of type "fanPressure". Selectable unit.*			
N_*patchX*-boundaryCondition-M_meridionalAngle	90	deg	yes*
▶ *Patch* patchX, Nth *component: Additional parameter for "directedMassFlowRate" and "directedVolumetricFlowRate" boundary conditions.*			
N_*patchX*-boundaryCondition-M_circumferentialAngle	0	deg	yes*
▶ *Patch* patchX, Nth *component: Additional parameter for "directedMassFlowRate" and "directedVolumetricFlowRate" boundary conditions.*			
N_*patchX*-boundaryCondition-M_outletVentResistance	0 4 8	–	yes*
▶ *Patch* patchX, Nth *component: Outlet vent boundary condition resistance factor values for individual points of the Mth speedline.*			
N_*patchX*-boundaryCondition-outletVentP0	101325	–	yes*
▶ *Patch* patchX, Nth *component: Outlet vent static pressure for zero resitance.*			
N_*patchX*-boundaryCondition-outletVentRelaxation	0.1	–	yes*
▶ *Patch* patchX, Nth *component: Outlet vent boundary condition relaxation factor value. Only for experienced users.*			
N_*patchX*-boundaryCondition-outletVentMaxPressure	200000	Pa	yes*
▶ *Patch* patchX, Nth *component: Outlet vent boundary condition maximum outlet pressure value. Selectable unit.*			
N_*patchX*-boundaryCondition-M_fixedPressure	100000 110000 120000	Pa	yes*
▶ *Patch* patchX, Nth *component: Uniform/Average (following the type *-type fixedPressure/fixedMeanPressure) outlet pressure value for all points of the Mth speedline. Selectable unit.*			
N_*patchX*-temperatureBoundaryCondition-type	totalTemperature	–	yes*

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
▶ *Patch* patchX, Nth *component: Temperature boundary condition to use. Possible options for inlet patch are: totalTemperature, fixedTemperature, freestreamTemperature. For wall patch: adiabaticWall, fixedTemperature, fixedPower, fixedHeatFlux, fixedHeatTransferCoeff.*			
N_*patchX*-temperatureBoundaryCondition-M_totalTemperature	1000_K 900_K 800_K	–	yes*
▶ *Patch* patchX, Nth *component: Boundary condition values for total temperature in compressible calculations for the Mth speedline. Selectable unit.*			
N_*patchX*-temperatureBoundaryCondition-M_totalTemperature-K_csv	../bc/temperatures.csv	–	no
▶ *Patch* patchX, Nth *component: Transient boundary condition data for Mth speedline, Kth point.*			
N_*patchX*-temperatureBoundaryCondition-M_fixedTemperature	280_K 290_K 300_K	–	yes*
▶ *Patch* patchX, Nth *component: Boundary condition values for static temperature in heatTransfer/compressible calculations for the Mth speedline. Selectable unit.*			
N_*patchX*-temperatureBoundaryCondition-M_fixedTemperature-K_csv	../bc/temperatures.csv	–	no
▶ *Patch* patchX, Nth *component: Transient boundary condition data for Mth speedline, Kth point.*			
N_*patchX*-temperatureBoundaryCondition-freestreamTemperature	290_K	–	yes*
▶ *Patch* patchX, Nth *component: Boundary condition values for freestream temperature in calculations where temperature field is present. Selectable unit.*			
turbulenceInletQuantities	turbulentIntensityAndHydraulicDiameter	–	no
▶ *Which quantities to use for defining turbulent inlet conditions.*			
N_*patchX*-turbulentBoundaryCondition-type	standardWalls	–	no
▶ *Wall patch* patchX, Nth *component: Turbulent boundary condition type. Possible options are standardWalls, lowReWalls and roughWalls.*			
N_*patchX*-turbulentBoundaryCondition-turbulentEddyViscosity	0.01	m^2/s	yes*
▶ *Patch* patchX, Nth *component: Parameter of the Spalart-Allmaras turbulence model, not needed for other models.*			
N_*patchX*-turbulentBoundaryCondition-hydraulicDiameter	1	m	yes*
▶ *Patch* patchX, Nth *component: Hydraulic diameter at inlet, needed only if option turbulentIntensityAndHydraulicDiameter is chosen as turbulenceInletQuantities. Selectable unit.*			
N_*patchX*-turbulentBoundaryCondition-turbulentLengthScale	0.07	m	yes*
▶ *Patch* patchX, Nth *component: Turbulent length scale at inlet, needed only if option turbulentIntensityAndLengthScale is chosen as turbulenceInletQuantities. Selectable unit.*			
N_*patchX*-turbulentBoundaryCondition-turbulentViscosityRatio	1	m	yes*
▶ *Patch* patchX, Nth *component: Ratio of turbulent and physical viscosity at inlet, needed only if option turbulentIntensityAndViscosityRatio or turbulentViscosityRatio is chosen as turbulenceInletQuantities.*			

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
N_*patchX*-turbulentBoundaryCondition-referenceVelocity	2	m/s	yes*
▶ *Patch* patchX, Nth *component: Reference velocity at inlet, needed only if option turbulentIntensityAndViscosityRatio, turbulentIntensityAndLengthScale or turbulentIntensityAndHydraulicDiameter is chosen as turbulenceInletQuantities. Selectable unit.*			
N_*patchX*-turbulentBoundaryCondition-turbulentDissipation	100	$J/kg * s$	no
▶ *Patch* patchX, Nth *component: Boundary condition for turbulent dissipation (epsilon).*			
N_*patchX*-turbulentBoundaryCondition-turbulentDissipationRate	100	1/s	no
▶ *Patch* patchX, Nth *component: Boundary condition for turbulent dissipation rate (omega).*			
N_*patchX*-turbulentBoundaryCondition-turbulentEnergyIntensity	0.05	–	no
▶ *Patch* patchX, Nth *component: Turbulent energy intensity, 0.05 means 5*			
N_*patchX*-turbulentBoundaryCondition-turbulentKineticEnergy	0.01	–	no
▶ *Patch* patchX, Nth *component: Turbulent kinetic energy on inlet.*			
N_*patchX*-turbulentBoundaryCondition-roughWalls-Ks	0	–	no
▶ *Patch* patchX, Nth *component: Sand-grain roughness value.*			
N_*patchX*-turbulentBoundaryCondition-roughWalls-Ks	0	–	no
▶ *Patch* patchX, Nth *component: Roughness constans.*			
N_*patchX*-boundaryCondition-M_passiveScalar-K	0.01	–	no
▶ *Patch* patchX, Nth *component: Inlet value for Mth speedline, Kth passive scalar.*			
N_*patchX*-temperatureBoundaryCondition-power	100	W	yes*
▶ *Patch* patchX, Nth *component: Heating power for BC of type "fixedPower".*			
N_*patchX*-temperatureBoundaryCondition-heatFlux	400	W/m^2	yes*
▶ *Patch* patchX, Nth *component: Heat flux for BC of type "fixedHeatFlux".*			
N_*patchX*-temperatureBoundaryCondition-M_ambientTemperature	280_K 285_K 290_K	–	yes*
▶ *Patch* patchX, Nth *component: Ambient fluid temperature values for all points of the Mth speedline for BC of type "fixedHeatTransferCoeff". Selectable unit*			
N_*patchX*-temperatureBoundaryCondition-M_ambientTemperature-K_csv	bc/ambTemp.csv	–	no
▶ *Patch* patchX, Nth *component: Transient boundary condition data for Mth speedline, Kth point.*			
N_*patchX*-temperatureBoundaryCondition-heatTransferCoeff	100	$W/(m^2K)$	yes*
▶ *Patch* patchX, Nth *component: Heat transfer coefficient on the outer side of the wall for BC of type "fixedHeatTransferCoeff".*			

Keyword ▶ ***Description***	**Allowed / sample values**	**Units**	**Mandatory**
N_*patchX*-temperatureBoundaryCondition-thicknessLayers	10 20 15	mm	yes*
▶ *Patch* patchX, Nth *component: Thicknesses of wall layers, that constitute the wall between inner domain and ambient fluid for BC of type "fixedHeatTransferCoeff".*			
N_*patchX*-temperatureBoundaryCondition-kappaLayers	100 180 60	W/m^2	yes*
▶ *Patch* patchX, Nth *component: Thermal conductivities of wall layers for BC of type "fixedHeatTransferCoeff".*			
N_*patchX*-temperatureBoundaryCondition-temperatureFile	temp.csv	–	yes*
▶ *Name of the file containing (spatially non-constant) values of temperature on the boundary. For BC of type "mappedTemperature".*			
N_*patchX*-temperatureBoundaryCondition-temperatureOffset	0_K	–	yes*
▶ *Offset value that is added to the entries from "temperatureFile". For BC of type "mappedTemperature".*			
numberOfInterfaceConditions	1	–	no
▶ *How many different interface condtitions are there (conditions on interfaces between the components).*			
N_interfaceCondition-type	pressureJump	–	no
▶ *Interface condition to use. Possible options are: pressureJump.*			
N_interfaceCondition-M_pressureJump	5000_Pa 6000_Pa 8000_Pa	–	yes*
▶ *Pressure jump values for all points of the Mth speedline for the Nth interface condition of typ "pressureJump".*			
N_interfaceCondition-M_pressureJump-K_csv	bc/pJump.csv	–	no
▶ *Transient boundary condition data for Nth interface condition, Mth speedline, Kth point.*			
N_interfaceCondition-interfaces	K:stator-outlet|2:outflow-inlet	–	yes*
▶ *Interface condition is applied to interface, which consist of patch 'stator-outlet' in component 1 and patch 'outflow-inlet' in component 2.*			
fixedPressure	101325	Pa	no
▶ *Needed for calculations without inlets and outlets (closedDomain type). Works similarly as the boundary condition - sets the pressure value at some point in the domain.*			
fixedPressurePoint	0 100 -200	–	no
▶ *Coordinates of the point with fixed pressure.*			
N_initialPressure	101325	Pa	yes
▶ *Initial condition for static pressure. Selectable unit.*			
N_initialVelocity	0 0 10	m/s	yes
▶ *Initial condition for velocity vector (x y z components).*			

Keyword ▶ ***Description***	Allowed / sample values	Units	Mandatory
N_initialTemperature ▶ *Initial condition for temperature. Only used in compressible calculations. Selectable unit.*	290	K	yes*
N_initialTurbulentEnergy ▶ *Initial condition for turbulent kinetic energy (k).*	1.5	m^2/s^2	yes
N_initialTurbulentDissipation ▶ *Initial condition for turbulent kinentic energy dissipation (epsilon).*	100	$J/kg * s$	yes
N_initialTurbulentDissipationRate ▶ *Initial condition for turbulent kinentic energy dissipation (omega).*	100	$1/s$	yes
N_initialTurbulentEddyViscosity ▶ *Initial condition for eddy viscosity(Spalart Allmaras model).*	0.001	–	no
N_initialPassiveScalar-2 ▶ *Initial condition 1. speedline, passive scalar number 2.*	0	–	no
initialTurbConditionsFromInlet ▶ *Whether to set initial turbulent conditions from their respective inlet values*	yes	–	no
turbulence ▶ *Turbulence model one of: laminar, kOmegaSST, kOmegaSSTLM, kEpsilon, realizableKE, RNGkEpsilon, SpalartAllmaras, SpalartAllmarasDDES.*	kOmegaSST	–	no
kOmegaSST-alphaK1 ▶ *Parameter of the k-omega SST model.*	0.85	–	no
kOmegaSST-alphaK2 ▶ *Parameter of the k-omega SST model.*	1	–	no
kOmegaSST-alphaOmega1 ▶ *Parameter of the k-omega SST model.*	0.5	–	no
kOmegaSST-alphaOmega2 ▶ *Parameter of the k-omega SST model.*	0.856	–	no
kOmegaSST-beta1 ▶ *Parameter of the k-omega SST model.*	0.075	–	no
kOmegaSST-beta2	0.0828	–	no

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
▶ *Parameter of the k-omega SST model.*			
kOmegaSST-betaStar	0.09	–	no
▶ *Parameter of the k-omega SST model.*			
kOmegaSST-gamma1	0.5555556	–	no
▶ *Parameter of the k-omega SST model.*			
kOmegaSST-gamma2	0.44	–	no
▶ *Parameter of the k-omega SST model.*			
kOmegaSST-a1	0.31	–	no
▶ *Parameter of the k-omega SST model.*			
kOmegaSST-b1	1	–	no
▶ *Parameter of the k-omega SST model.*			
kOmegaSST-c1	10	–	no
▶ *Parameter of the k-omega SST model.*			
kOmegaSST-F3	0	–	no
▶ *Parameter of the k-omega SST model.*			
kOmegaSSTLM-ca1	2	–	no
▶ *Parameter of the k-omega SSTLM model.*			
kOmegaSSTLM-ca2	0.06	–	no
▶ *Parameter of the k-omega SSTLM model.*			
kOmegaSSTLM-ce1	1	–	no
▶ *Parameter of the k-omega SSTLM model.*			
kOmegaSSTLM-ce2	50	–	no
▶ *Parameter of the k-omega SSTLM model.*			
kOmegaSSTLM-cThetat	0.03	–	no
▶ *Parameter of the k-omega SSTLM model.*			
kOmegaSSTLM-sigmaThetat	2	–	no
▶ *Parameter of the k-omega SSTLM model.*			
kEpsilon-Cmu	0.09	–	no

Keyword ▶ ***Description***	**Allowed / sample values**	**Units**	**Mandatory**
▶ *Parameter of the k-epsilon model.*			
kEpsilon-C1	1.44	–	no
▶ *Parameter of the k-epsilon model.*			
kEpsilon-C2	1.92	–	no
▶ *Parameter of the k-epsilon model.*			
kEpsilon-C3	-0.33	–	no
▶ *Parameter of the k-epsilon model.*			
kEpsilon-sigmak	1	–	no
▶ *Parameter of the k-epsilon model.*			
kEpsilon-sigmaEps	1.3	–	no
▶ *Parameter of the k-epsilon model.*			
realizableKE-A0	0.09	–	no
▶ *Parameter of the realizible k-epsilon model.*			
realizableKE-C2	1.92	–	no
▶ *Parameter of the realizible k-epsilon model.*			
realizableKE-sigma	1.44	–	no
▶ *Parameter of the realizible k-epsilon model.*			
realizableKE-sigmaEp	-0.33	–	no
▶ *Parameter of the realizible k-epsilon model.*			
RNGkEpsilon-C1	1.44	–	no
▶ *Parameter of the renormalization group k-epsilon model.*			
RNGkEpsilon-C2	1.92	–	no
▶ *Parameter of the renormalization group k-epsilon model.*			
RNGkEpsilon-C3	-0.33	–	no
▶ *Parameter of the renormalization group k-epsilon model.*			
RNGkEpsilon-Cmu	0.09	–	no
▶ *Parameter of the renormalization group k-epsilon model.*			
RNGkEpsilon-beta	1.3	–	no

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
▶ *Parameter of the renormalization group k-epsilon model.*			
RNGkEpsilon-eta0	1	–	no
▶ *Parameter of the renormalization group k-epsilon model.*			
RNGkEpsilon-sigmaEps	1.3	–	no
▶ *Parameter of the renormalization group k-epsilon model.*			
RNGkEpsilon-sigmak	1	–	no
▶ *Parameter of the renormalization group k-epsilon model.*			
LESDelta-cubeRootVol-delta	1	–	no
▶ *Parameter of the Spalart-Allmaras turbulence model, not needed for other models.*			
Prt	0.85	–	no
▶ *Turbulent Prandtl number.*			
N_wheelDiameter	0	scaleFactor	no
▶ *Wheel / Impeller / Rotor diameter for the Nth component. Used to post-process fans.*			
numberOfEfficiencyProbes	1	–	no
▶ *How many reports to generate (default: 1).*			
N_efficiencyProbe-inletPatches	1:pump_pipe_inlet	–	no
▶ *Inlet patches for the Nth report (default: inlet of first component). Specify patches as "<component>:<patch>".*			
N_efficiencyProbe-torquePatches	2:pump_rotor_PS 2:pump_rotor_SS	–	no
▶ *Torque patches for the Nth report (default: blades of rotating comps.).*			
N_efficiencyProbe-outletPatches	3:pump_spiral_outlet	–	no
▶ *Outlet patches for the Nth report (default: outlet of last component).*			
N_efficiencyProbe-convergenceCheck	yes	–	no
▶ *Whether to include this efficiency probe in the convergence check.*			
N_efficiencyProbe-monitored	yes	–	no
▶ *If the efficiency probe should be added to Quantity monitor (default: yes).*			
efficiencyProbesFluxWeighted	yes	–	no
▶ *Use mass flow weighting when evaluating quantities needed for efficiency calculation.*			
numberOfForcesProbes	1	–	no

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
▶ *How many forces to evaluate (default: 0).*			
N_forcesProbe-inletPatches	1:hull 2:hull-tail	–	no
▶ *Reference velocity for force coefficients is evaluated on these patches. Specify patches as "<component>:<patch>".*			
N_forcesProbe-patches	2:blade	–	no
▶ *Patches used for the force evaluation. Specify patches as "<component>:<patch>".*			
N_forcesProbe-liftDirection	0 0 1	–	no
▶ *Lift direction.*			
N_forcesProbe-dragDirection	1 0 0	–	no
▶ *Drag direction.*			
N_forcesProbe-pitchAxis	1 0 0	–	no
▶ *Axis for moment coefficient evaluation.*			
N_forcesProbe-torqueAxis	0 0 1	–	no
▶ *Axis for torque evaluation.*			
N_forcesProbe-CofR	0 0 0	–	no
▶ *Center of rotation for moment calculations.*			
N_forcesProbe-referenceArea	1	–	no
▶ *Reference area for coefficients evaluation.*			
N_forcesProbe-referenceLength	1	–	no
▶ *Reference length for moment coefficient evaluation.*			
N_forcesProbe-referenceVelocity	1	–	no
▶ *Reference (freestream) velocity magnitude.*			
N_forcesProbe-monitored	yes	–	no
▶ *Whether to include value from this probe to live monitoring during calculation.*			
numberOfBladeToBladeViews	1	–	no
▶ *Number of distinct blade-to-blade views to include in the report (default: 0).*			
bladeToBladeView-groupByPoint	no	–	no
▶ *Reorder blade-to-blade view figures in report so that they are grouped by point and not by span.*			
N_bladeToBladeView-meshes	2	–	no

Keyword	Allowed / sample values	Units	Mandatory
▶ ***Description***			
▶ *Meshes to unwrap for the Nth blade-to-blade view, mostly whole components or blade patches. Specify component by name or number, patch by "<component>:<patch>".*			
N_bladeToBladeView-hubs	1:pump_rotor_hub	–	no
▶ *Hub patches needed to define the Nth blade-to-blade view.*			
N_bladeToBladeView-shrouds	1:pump_rotor_shroud	–	no
▶ *Shroud patches needed to define the Nth blade-to-blade view.*			
N_bladeToBladeView-field	U	–	no
▶ *Field to display on the Nth blade-to-blade view.*			
N_bladeToBladeView-span	0.25 0.5 0.75	–	no
▶ *Distances between hub and shroud for the Nth set of blade-to-blade views.*			
numberOfMeridionalAverages	1	–	no
▶ *Number of distinct meridional average views to include in the report (default: 0).*			
N_meridionalAverage-meshes	2	–	no
▶ *Meshes to average for the Nth meridional average view. Specify component by its name or number.*			
N_meridionalAverage-field	p	–	no
▶ *Field to display on the Nth meridional average view.*			
numberOfProbes	1	–	no
▶ *Number of distinct probes (point locations) for specific fields value evaluation (default: 0).*			
N_probe-fields	p T	–	no
▶ *Fields to be evaluated by Nth probe.*			
N_probe-location	0.15 0.113 -0.04	–	no
▶ *Point location.*			
N_probe-monitored	no	–	no
▶ *If the probe should be added to Quantity monitor (default: yes).*			
computePressureCoefficient	yes	–	no
▶ *Compute pressure coefficient c_p on selected patches.*			
pressureCoefficient-patches	1:wing_left 1:wing_right	–	no
▶ *Specifies on which wall patches should the pressure coefficient be computed.*			

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
pressureCoefficient-referenceVelocity	120_m/s	–	no
▶ *Reference velocity of the fluid needed to calculate pressure coefficient.*			
writeCGNS	no	–	no
▶ *Whether to save mesh with results in CGNS format after calculation.*			
calcSurfaceQuantities-perform	no	–	no
▶ *Whether to save chosen quantities and surface areas of all faces on surfaces. The results are written to csv file.*			
calcSurfaceQuantities-writeForces	no	–	no
▶ *Whether to include forces to the surface saved quantities.*			
calcSurfaceQuantities-writeTemperature	no	–	no
▶ *Whether to include temperature to the surface saved quantities.*			
calcSurfaceQuantities-writeHeatFlux	no	–	no
▶ *Whether to include heat flux to the surface saved quantities.*			
calcSurfaceQuantities-writeHTC	no	–	no
▶ *Whether to include heat transfer coefficient to the surface saved quantities.*			
calcSurfaceQuantities-HTCYPlus	no	–	no
▶ *Value of the yPlus where the ambient temperature is evaluated.*			
calcSurfaceQuantities-writeWallShearStress	no	–	no
▶ *Whether to include wall shear stress to the surface saved quantities.*			
calcSurfaceQuantities-density	no	–	no
▶ *Whether to include density to the surface saved quantities.*			
calcSurfaceQuantities-useRelativePressure	no	–	no
▶ *Whether to use relative pressure for computing forces, that are written to csv file.*			
calcSurfaceQuantities-useAveragedQuantities	no	–	no
▶ *Whether to use averaged or instantaneous values of selected quantities.*			
calcSurfaceQuantities-mapOnExternalMesh	no	–	no
▶ *Whether to map the surface quantities on an external mesh.*			
calcSurfaceQuantities-externalMeshType	abaqus	–	no
▶ *External mesh format. Currently only option is abaqus.*			

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
calcSurfaceQuantities-externalMeshName	data/Abaqus-mesh-to-map-to.inp	–	no
▶ *Path to external mesh for mapping.*			
calcSurfaceQuantities-patchesForMapping	1:wall	–	no
▶ *List of patches that are used for the mapping on external mesh .*			
calcSurfaceQuantities-scaleFactor	no	–	no
▶ *The external mesh will be scaled according to this value.*			
numberOfSurfaceSamples	1	–	no
▶ *Number of distinct surface samples for specific fields and patches to be exported (default: 0).*			
N_surfaceSample-fields	p	–	no
▶ *Fields to be sampled.*			
N_surfaceSample-patches	2:impeller-blades 2:impeller-hub	–	no
▶ *Patches to be sampled, "<component>:<patch>".*			
reportUnit-massFlowRate	1_kg/s	–	no
▶ *Choice of unit for mass flow rate in report,choice of kg/s, kg/min, kg/h, g/s, g/min, g/h.*			
reportUnit-volumetricFlowRate	1_m^3/s	–	no
▶ *Choice of unit for volumetric flow rate in report, choice of m^3/s, m^3/min, m^3/h, l/s, l/min, l/h, USgal/s, USgal/min, USgal/h.*			
reportUnit-pressure	1_Pa	–	no
▶ *Choice of unit for pressure in report, choice of Pa, bar, mbar, atm, torr, psi.*			
reportUnit-temperature	1_K	–	no
▶ *Choice of unit for temperature in report, choice of*			
reportUnit-velocity	1_m/s	–	no
▶ *Choice of unit for speed in report.*			
streamPath	1 2 3	–	no
▶ *Route through the machine for total pressure, velocity and total temperature plot.*			
additionalGraphDataFiles	data/eta.dat flowRateVsEfficiency,1,2	–	no
▶ *Supplies additional data sources for report graphs.*			
reportQuantity	flowRate	–	no

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
▶ *Choice to base report either on flowRate or velocity.*			
TCAA-reportSections	Head Efficiency Torque	–	no
▶ *Which sections to include in report, choice of Efficiency, Torque, TotalPressureDifference, TotalPressureRatio, TotalPressurePerInterfaces, VelocityMagnitudePerInterfaces, CircumferentialMeridionalAngle, CavitationRisk, TotalTemperatureDifference, TotalTemperaturePerInterfaces, Head, DimensionlessCompression, TurboBladePost, PerformanceCurve, ForceCoeffs.*			

Chapter 13

TCFD – Comparing reports

Results from a TCFD simulation are presented in a HTML report that can be displayed in the TCAE GUI as well as in any common HTML viewer. Often it is useful to be able to compare reports from separate calculations, for instance with different blade geometries, fluid parameters etc. TCAE can generate a comparative HTML report consisting of two or more separate TCFD reports. The tables of the two (or more) reports will be places side by side for easier analysis, and some plots will be combined into one. This can be done in GUI, or in the command line.

Note that even though the below methods seem to require the HTML files only, actually the whole report directories (with CSV and XML data files) are required for producing the comparative report.

13.1 Comparing in GUI

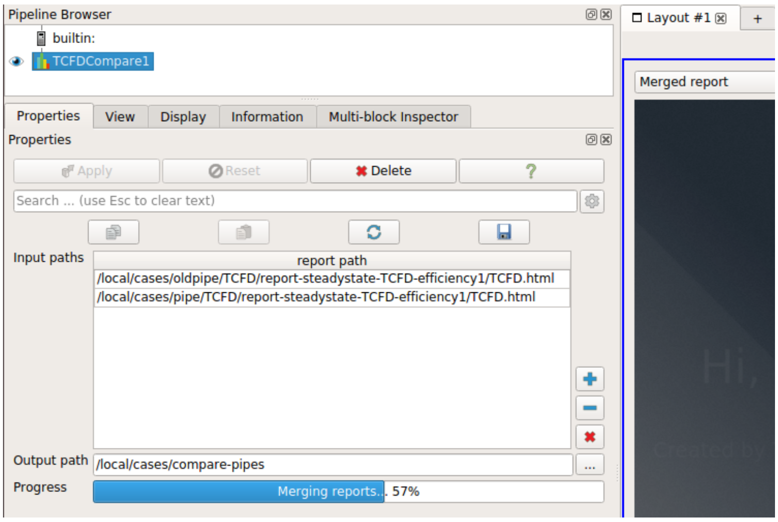
In TCAE GUI, one can use the TCFD Compare source. It can be selected from the menu (*Sources* → *TCFD Sources*) or by clicking on a dedicated icon in the toolbar:

A simple table appears in the *Properties* panel. Using the plus button next to the table add both (or all) reports that are to be combined, choose the output directory where to write the resulting comparative report and press *Apply*, see figure below. Once the report is done, it will also immediately appear in the built-in HTML viewer in ParaView.

13.2 Comparing in command line

TCAE offers a command-line program `compareCFDReports` that can be directly used to generate the comparative report. Simply run it from a directory where the report is to be generated and give it paths to all TCFD HTML files to be merged:

```
compareCFDReports report1.html report2.html
```



Input paths

report path
/local/cases/oldpipe/TCFD/report-steadystate-TCFD-efficiency1/TCFD.html
/local/cases/pipe/TCFD/report-steadystate-TCFD-efficiency1/TCFD.html

Output path /local/cases/compare-pipes

Progress Merging reports... 57%

Figure 13.1: TCFD Compare - generation of a comparative report, in progress

In Linux, the program `compareCFDReports` is directly available in the command line. In Windows, its full path needs to be specified, which is

“`C:\TCAE\20.09\bin\compareCFDReports.exe`”

or in a different directory specified during the installation of TCAE.

Chapter 14

TCFD – CFD Theory & Formulas

14.1 TCFD Solvers

14.1.1 blueSolver, blueDyMSolver

TCFD solver for steady-state [transient], incompressible fluid flow is called blueSolver [blueDyMSolver]. It was gradually developed during the time from the simpleFoam [pimpleFoam] solver. In any matters the blueSolver behaves the same way as any standard OpenFOAM solver. It is compatible with all OpenFOAM applications and libraries. Solver is modified to be more robust, it can use MRF method, the limits for variables can be specified and many other changes have been done.

Governing equations for blueSolver - incompressible steady-state Navier-Stokes equations:

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 \\ \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) &= \nu_{\text{eff}} \nabla^2 \mathbf{u} - \nabla p_k + \mathbf{g} \end{aligned} \tag{14.1}$$

Governing equations for blueDyMSolver - incompressible transient Navier-Stokes equations:

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0 \\ \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) &= \nu_{\text{eff}} \nabla^2 \mathbf{u} - \nabla p_k + \mathbf{g} \end{aligned} \tag{14.2}$$

In both cases the system consist of mass conservation equation and momentum conservation equation.

Following symbols are used: ∂ is partial derivative, ∇ is operator nabla $\left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)$, $\mathbf{u}$ is velocity vector, $\otimes$ is outer product, t is time, p_k is kinematic pressure $[\text{Pa}/(\text{kg}\,\text{m}^{-3}) = \text{m}^2\text{s}^{-2}]$, $\mathbf{g}$ is gravitational acceleration vector and ν_{eff} is effective kinematic viscosity. If the flow is considered as laminar, the effective kinematic viscosity is simply kinematic viscosity $\nu_{\text{eff}} = \nu$. If turbulence is modelled using RANS or LES approach, following relation holds $\nu_{\text{eff}} = \nu + \nu_{\text{t}}$, where ν_{t} is turbulent kinematic viscosity. Its calculation is performed by the turbulent model.

Kinematic pressure

Incompressible N-S equations 14.1, 14.2 are derived from the more general counterparts 14.3, 14.4. In the form above, they doesn't contain density explicitly. Density (which is constant through the computational domain) is included in the so called kinematic pressure $p_k = p/\rho_0$. This way are the equations simplified. As equations above indicate, TCFD blueSolver and blueDyMsolver use the kinematic density, but bear in mind, when running TCFD both in GUI and terminal, input values like boundary and initial conditions are expected to be in "standard" pressure. The conversion is done automatically.

14.1.2 redSolver, redDyMSolver

TCFD solver for steady-state [transient], compressible fluid flow is called redSolver [redDyMSolver]. It was gradually developed during the time from the rhoSimpleFoam [sonicFoam] solver. In any matters the redSolver [redDyMSolver] behaves the same way as any standard OpenFOAM solver. It is compatible with all OpenFOAM applications and libraries. Solver is modified to be more robust, it can use MRF method, the limits for variables can be specified and many other changes have been done.

Governing equations for redSolver are compressible steady-state Navier-Stokes equations

$$\begin{aligned}
\nabla \cdot (\rho \mathbf{u}) &= 0 \\
\nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) &= -\nabla p + \nabla \cdot \left[\mu_{\text{eff}} \left((\nabla \mathbf{u})^T - \frac{2}{3}(\nabla \cdot \mathbf{u})\mathbf{I}\right)\right] + \\
&\quad + \nabla \cdot (\mu_{\text{eff}} \nabla \mathbf{u}) \\
\nabla \cdot (\rho \mathbf{u} h) + \nabla \cdot (\rho \mathbf{u} K) &= \nabla \cdot \left[\mu_{\text{eff}} \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T - \frac{2}{3}(\nabla \cdot \mathbf{u})\mathbf{I}\right) \cdot \mathbf{u}\right] + \\
&\quad + \nabla \cdot (\alpha_{\text{eff}} \nabla h)
\end{aligned} \tag{14.3}$$

Governing equations for redDyMSolver are compressible transient Navier-Stokes equations

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) &= 0 \\
\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) &= \nabla \cdot \left[\mu_{\text{eff}} \left((\nabla \mathbf{u})^T - \frac{2}{3}(\nabla \cdot \mathbf{u})\mathbf{I}\right)\right] - \\
&\quad - \nabla p + \nabla \cdot (\mu_{\text{eff}} \nabla \mathbf{u}) \\
\frac{\partial \rho e}{\partial t} + \nabla \cdot (\rho \mathbf{u} e) + \frac{\partial \rho K}{\partial t} + \nabla \cdot (\rho \mathbf{u} K) &= \nabla \cdot (\alpha_{\text{eff}} \nabla e) - \nabla \cdot (\mathbf{u} p)
\end{aligned} \tag{14.4}$$

Additional symbols: ρ is density, $\otimes$ is outer product, $\mathbf{I}$ is identity matrix, ^{T} is transpose operator, p is pressure, μ_{eff} is effective dynamic viscosity, α_{eff} is effective thermal diffusivity, h is specific enthalpy (per unit mass), K is specific kinetic energy (per unit mass) $K = \frac{1}{2}|\mathbf{u}|^2$ and e is specific internal energy (per unit mass).

Just as in the incompressible case, if the flow is considered as laminar, the effective dynamic viscosity is simply dynamic viscosity $\mu_{\text{eff}} = \mu$. If turbulence is modelled using RANS or LES approach, following relation holds $\mu_{\text{eff}} = \mu + \mu_{\text{t}}$, where μ_{t} is turbulent dynamic viscosity. Its calculation is performed by the turbulent model.

Effective thermal diffusivity is defined as sum of laminar and turbulent thermal diffusivities

$$\alpha_{\text{eff}} = \alpha + \alpha_{\text{t}} = \frac{\mu_{\text{t}}}{\text{Pr}_{\text{t}}} + \frac{\mu}{\text{Pr}} = \frac{\mu_{\text{t}}}{\text{Pr}_{\text{t}}} + \frac{\kappa}{c_p},$$

where κ is the termal conductivity and the turbulent Prandtl number is set to one $\mathrm{Pr_t} = 1$. In this case the thermal diffusivity is defined differently from the standard physical definition in multiplication by the density:

$$\alpha = \frac{\kappa}{c_p}, \ \alpha_{\text{standard}} = \frac{\kappa}{\rho c_p}.$$

Specific enthalpy is defined as sum of specific internal energy and kinematic pressure $h = e + p/\rho$. Both enthalpy and internal energy might be computed from temperature: $h = c_p T$, $e = c_v T$, where c_p is specific heat capacity at constant pressure and c_v specific heat capacity at constant volume.

Whole system is closed with perfect gas equation of state

$$p = \rho r T, \tag{14.5}$$

where r is specific gas constant and T thermodynamic temperature.

14.1.3 orangeSolver, orangeDyMSolver

TCFD solver for steady-state [transient] fluid flow with heat transfer is called orangeSolver [orangeDyMSolver]. It was developed from the buoyantSimpleFoam [buoyantPimpleFoam] solver. In any matters the orangeSolver [orangeDyMSolver] behaves the same way as any standard OpenFOAM solver. It is compatible with all OpenFOAM applications and libraries. Solver is modified to be more robust, it can use MRF method, limits the for variables can be specified and many other changes have been done.

Governing equations for orangeSolver are compressible steady-state Navier-Stokes equations

$$\begin{aligned} \nabla\cdot(\rho\mathbf{u}) &= 0 \\ \nabla\cdot(\rho\mathbf{u}\otimes\mathbf{u}) &= -\nabla p_{\text{rgh}} - (\mathbf{r}\cdot\mathbf{g})\nabla\rho + \nabla\cdot(\mu_{\text{eff}}\nabla\mathbf{u}) + \\ &\quad + \nabla\cdot\left[\mu_{\text{eff}}\left((\nabla\mathbf{u})^T - \frac{2}{3}(\nabla\cdot\mathbf{u})\mathbf{I}\right)\right] \\ \nabla\cdot(\rho\mathbf{u}h) + \nabla\cdot(\rho\mathbf{u}K) &= \nabla\cdot(\alpha_{\text{eff}}\nabla h) + \rho\mathbf{u}\cdot\mathbf{g} \end{aligned} \tag{14.6}$$

Governing equations for orangeDyMSolver are compressible transient Navier-Stokes equations

$$\begin{aligned} \frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{u}) &= 0 \\ \frac{\partial\rho\mathbf{u}}{\partial t} + \nabla\cdot(\rho\mathbf{u}\otimes\mathbf{u}) &= -\nabla p_{\text{rgh}} - (\mathbf{r}\cdot\mathbf{g})\nabla\rho + \nabla\cdot(\mu_{eff}\nabla\mathbf{u}) + \\ &\quad + \nabla\cdot\left[\mu_{eff}\left((\nabla\mathbf{u})^T - \frac{2}{3}(\nabla\cdot\mathbf{u})\mathbf{I}\right)\right] \\ \frac{\partial\rho h}{\partial t} + \nabla\cdot(\rho\mathbf{u}h) + \frac{\partial\rho K}{\partial t} + \nabla\cdot(\rho\mathbf{u}K) &= \frac{\partial p}{\partial t} + \nabla\cdot(\alpha_{eff}\nabla h) + \rho\mathbf{u}\cdot\mathbf{g} \end{aligned} \tag{14.7}$$

Both "orange" solvers differs from the "red" ones in using pressure field $p_{\text{rgh}} = p - \rho\mathbf{g}\cdot\mathbf{r} = p - \rho g h$ as primary, where $\mathbf{r}$ is position vector. p_{rgh} might be viewed as pressure without hydrostatic part.

Whole system is closed with equation of state. The two possibilities are perfect gas 14.5, or Boussinesq equation of state:

$$\rho = \rho_0[1 - \beta(T - T_o)], \tag{14.8}$$

where β is thermal expansion coefficient, ρ_0 is reference density and T_0 reference temperature. Boussinesq equation of state is used for buoyancy driven flows.

14.1.4 greenDyMSolver

TCFD solver for transient, cavitating fluid flow is called greenDyMSolver. It was gradually developed during the time from the interPhaseChangeDyMFoam solver. In any matters the greenDyMSolver behaves the same way as any standard OpenFOAM solver. It is compatible with all OpenFOAM applications and libraries. Solver is modified to be more robust, limits the for variables can be specified and many other changes have been done.

The main features are:

- Two-fluid model
- Eulerian-Eulerian approach
- Incompressible, isothermal, immiscible fluids
- Interface capturing approach based on VOF (volume of fluids)
- The momentum and other fluid properties are solver for a mixture (mixture viscosity, density), i.e., a single momentum equation is solved.
- Transport equation of phase-fraction includes the phase change

Governing equations

The physical properties of the fluid for VOF method of liquid-vapour mixture are modeled by a volume fraction α_l, α_v, where $\alpha_l = 1$ indicates a pure liquid (water), $\alpha_v = 1$ a pure gas (vapor) and $\alpha_l + \alpha_v = 1$. We will sometimes refer to α_l as α because it is a major phase in this two-phase system.

Properties of a mixture is given by the mixture density (ρ_m) and the mixture dynamic viscosity (μ_m):

$$\begin{aligned} \mu_m &= (1-\alpha)\mu_v + \alpha\mu_l \\ \rho_m &= (1-\alpha)\rho_v + \alpha\rho_l \,, \end{aligned}$$

where index l indicates property of water (liquid) whereas index v indicates property of vapour.

The system of equations looks then as follows:

$$\frac{\partial \rho_m}{\partial t} + \nabla \cdot (\rho_m \mathbf{u}) = 0 \tag{14.9}$$

$$\frac{\partial \rho_m \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)(\rho_m \mathbf{u}) = -\nabla p + \nabla \cdot (\mu_{\text{eff}}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)) + f_\sigma \tag{14.10}$$

$$\frac{\partial \rho_v \alpha_v}{\partial t} + (\mathbf{u} \cdot \nabla)(\rho_v \alpha_v) + (\mathbf{u}_\alpha \cdot \nabla)(\alpha_v(1-\alpha_v)) = R_c - R_e \,, \tag{14.11}$$

where f_σ is a surface tension force

$$f_\sigma = -\sigma \left[\nabla \cdot \frac{\nabla\alpha}{|\nabla\alpha|}\right] \nabla\alpha$$

and $\mathbf{u}_\alpha$ denotes an interface-compression velocity used for modelling an artificial compression describing the shrinkage of the phase-interface towards a sharper one. It is applied at the interface only and is given as

$$\mathbf{u}_\alpha = \min\left\{C_\alpha|\mathbf{u}|, \max\{\mathbf{u}\}\right\} \frac{\nabla\alpha}{|\nabla\alpha|} .$$

The source terms on the right side of Equation (14.11) R_c and R_e denote the rate of mass transfer of condensation and evaporation. Cavitation models are reffered to the particular representation of the source terms R_c and R_e. Particular models are discussed in Section 14.8.

14.2 Formulas for total quantities

Total (stagnation) quantities are used extensively in physical description of the flow. In CFD, and so in TCFD, are mostly used in boundary conditions. TCFD solvers also automatically computes total quantities fields for further postprocessing. These quantities defined as values at stagnation point in the flow, that is a point, where the fluid velocity is zero and the fluid is brought to rest isentropically (without losses).

- Incompressible cases
  Total pressure (kinematic)

$$p_{\text{k,tot}} = p_k + \frac{|\mathbf{u}|^2}{2} \tag{14.12}$$

  Total pressure

$$p_{\text{tot}} = p + \rho\frac{|\mathbf{u}|^2}{2} \tag{14.13}$$

- Compressible cases
  Total pressure

$$p_{\text{tot}} = p\left(1 + \frac{\gamma-1}{2\gamma}\frac{|\mathbf{u}|^2}{rT}\right)^{\frac{\gamma}{\gamma-1}} \tag{14.14}$$

  Total temperature

$$T_{\text{tot}} = T\left(1 + \frac{\gamma-1}{2\gamma}\frac{|\mathbf{u}|^2}{rT}\right) \tag{14.15}$$

14.3 Formulas for the Efficiency and Machine Charasteristics Evaluation

In the formulas contained in the following subsections, subscript $_{\text{in}}$ means averaged quantities at the inlet, whereas subscript $_{\text{out}}$ denotes averaged quantities at the outlet. $_{tot}$ denotes total quantities, defined by the relations above.

14.3.1 Hydro Turbine Efficiency

The water turbine efficiency can be evaluated using following efficiency formula:

$$\eta^{waterTurbine} = \frac{M\omega}{\dot{V}_{in}h_{in} - \dot{V}_{out}h_{out}}, \tag{14.16}$$

where η denotes the efficiency, M is the torque, ω is the angular velocity, $\dot{V}$ is the volumetric flow rate and h is the specific enthalpy which can be evaluated as follows:

$$h = p + (-\rho\mathbf{g}) \cdot \mathbf{r} + \frac{1}{2}\rho|\mathbf{u}|^2. \tag{14.17}$$

14.3.2 Pump Efficiency

The pump efficiency can be evaluated using following efficiency formula:

$$\eta^{pump} = \frac{\dot{V}_{\mathrm{in}}h_{in} - \dot{V}_{out}h_{out}}{M\omega}, \tag{14.18}$$

where η denotes the efficiency, M is the torque, ω is the angular velocity, $\dot{V}$ is the volumetric flow rate and h is the specific enthalpy which can be evaluated as follows:

$$h = p + (-\rho\mathbf{g}) \cdot \mathbf{r} + \frac{1}{2}\rho|\mathbf{u}|^2. \tag{14.19}$$

The evaluation of pump efficiency for different patches is also possible.

14.3.3 Compressor Efficiency

Adiabatic efficiency. Total to Total. For compressor the following efficiency formula is used:

$$\eta_a^{compressor} = \frac{T_{tot,out,IS} - T_{tot,in}}{T_{tot,out} - T_{tot,in}} = \frac{T_{tot,in}\left(\frac{p_{tot,out}}{p_{tot,in}}\right)^{\frac{\gamma-1}{\gamma}} - T_{tot.in}}{T_{tot,out} - T_{tot,in}} \tag{14.20}$$

14.3.4 Turbine Efficiency

Adiabatic efficiency. Total to static. For steam turbine or centrifugal turbine the following efficiency formula is used:

$$\eta_a^{turbine} = \frac{T_{tot,out} - T_{tot,in}}{T_{tot,in}\left(\frac{p_{out}}{p_{tot,in}}\right)^{\frac{\gamma-1}{\gamma}} - T_{tot,in}} \tag{14.21}$$

14.3.5 Fan Formulas

Fan Efficiency

Fan efficiency is given by the formula

$$\eta^{fan} = \frac{P_t}{P_w}, \tag{14.22}$$

where following quantities are used:

- P_t aerodynamic power, $P_t = Y_t \cdot \dot{m}_2$ $[W]$
- P_w torque power, $P_w = M \cdot \omega$ $[W]$
- Y_t total work, $Y_t = Y_{st} + Y_d$ $[m^2/s^2]$
- Y_{st} static work, $Y_{st} = f \cdot \frac{\Delta p}{\rho_{in}}$ $[m^2/s^2]$
- Y_d dynamic work, $Y_d = \frac{u_2^2 - u_1^2}{2}$ $[m^2/s^2]$
- f compress factor, $f = 1 - 0.36 \cdot \frac{\Delta p}{p_{out}}$ $[-]$
- Δp static pressure difference $\Delta p = p_{in} - p_{out}$ $[Pa]$
- M torque at wheel, $[Nm]$

Fan Parameters

Pressure coefficient

$$\psi = \frac{2\Delta p_{tot}}{\rho_{in} u_c^2} = \frac{729.5\Delta p_{tot}}{\rho_{in} n^2 D^2} \; [-] \tag{14.23}$$

Flow coefficient

$$\phi = \frac{\dot{V}}{A u_c} = \frac{24.3\dot{V}_w}{n D^3} \; [-] \tag{14.24}$$

In these formulas the u_c denotes peripheral speed of the impeller external diameter,

14.3.6 Wind Turbine Efficiency

Wind turbine efficiency $\eta^{windTurbine}$ is given by formula

$$\eta^{windTurbine} = \frac{P}{P_{max}} = \frac{M\omega}{0.593\frac{1}{2}\rho A u_{in}^3}, \tag{14.25}$$

where P is the power of the turbine, P_{max} is the maximal theoretical power given by Betz's law, M is torque of the rotor and ω is angular velocity of the rotor, ρ is density of air, A is area covered by the rotor and u_{in} is the velocity magnitude at inlet.

14.3.7 Propeller Efficiency

There are two quantities of interest for propeller:

- Thrust – T $[N]$
- Torque – M $[Nm]$

In case of propeller, reference pressure is not taken into account when evaluating thrust and torque.

Dimensional analysis leads to a definition of propeller coefficients representing its performance. [1]

- Torque coefficient (k_Q):
$$k_Q = \frac{M}{\rho n^2 D^5}\,, \tag{14.26}$$
- Thrust coefficient (k_T):
$$k_T = \frac{T}{\rho n^2 D^4}\,, \tag{14.27}$$

where n is the propeller speed [rev/s] and D denotes the propeller diamater [m].

The efficiency can be evaluated using the supplied power to the propeller (P_{in}) and the useful power output (P_{out}):

$$\eta^{propeller} = \frac{P_{\text{out}}}{P_{\text{in}}} = \frac{TU_0}{2\pi n M} = \frac{1}{2\pi}\frac{k_T}{k_Q}J\,, \tag{14.28}$$

where J is the advance ratio and is given as the distance advanced by the propeller in one revolution divided by the propeller diameter, i.e., $J = U_0/(Dn)\,.$

14.3.8 Stator Efficiency

Stator efficiency is evaluated as the total pressure drop divided by the total pressure at the inlet:

$$\eta^{stator} = \frac{p_{\text{tot,in}} - p_{\text{tot,out}}}{p_{\text{tot,in}}} \tag{14.29}$$

14.3.9 Virtual Tunnel Efficiency

The simulation type **virutalTunnel** has no efficiency formula to be evaluated. This simulation type is primarily focused on forces and force coefficient evaluation.

14.4 General CFD notes

14.4.1 Unstructured Grid

The computational mesh data is kept in unstructured OpenFOAM format. See e.g. [12].

14.4.2 Finite Volume Method

Solver is based on Finite Volume Method more information can be found e.g. in [2], [10] or [9].

[1] MIT propulsion notes (*http://web.mit.edu/16.unified/www/FALL/thermodynamics/notes/node86.html*)

14.4.3 Three Dimensional

All the models are solved in three dimensions, even 2D-like or 1D-like models are treated as 3D using special boundary conditions. See e.g. [12].

14.4.4 Segregated Solver

Segregated solver is used to compute unknown variables. The Finite Volume Solution Method can either use a segregated or a coupled solution procedure. With segregated methods an equation for a certain variable is solved for all cells, then the equation for the next variable is solved for all cells, etc. For more details see e.g. [1].

14.4.5 Cell Centered Approach

A cell-centered approach stores the variable in all cell centers whereas a node-centered scheme stores it in the points. For more details see e.g. [3].

14.4.6 Under-Relaxation

Under-Relaxation reduces solution oscillations and helps to keep the computation stable. After each iteration, at each cell, a new value for variable U in cell i is then updated using following equation:

$$U_i^{NEW,USED} = U_i^{OLD} + \alpha\Big(U_i^{NEW,PREDICTED} - U_i^{OLD}\Big) \tag{14.30}$$

where α is under-relaxation factor. The choice $\alpha = 1$ corresponds to no under-relaxation. The choice $\alpha < 1$ is under-relaxation. This may slow down speed of convergence but increases the stability of the computation, i.e. it decreases the possibility of divergence or oscillations in the solution. For more details see e.g. [1].

14.4.7 System of Linear Equations

Finite Volume Method converts the system of differential equations to the system of linear equations:

$$\mathbf{A} \cdot x = b \tag{14.31}$$

Such a linear algebra problem can be solved with following methods implemented in OpenFOAM:

14.4.8 SIMPLE Algorithm

For solving pressure - velocity coupling the SIMPLE algorithm is used. For more details see e.g. [1].

14.4.9 Spatial Integration Numerical Scheme

Space discretization scheme is limitedLinear, which is central scheme of second order accuracy. There are 55 default OpenFOAM schemes to select:

For more details see e.g. [9], [1] or [3].

14.4.10 Non-Orthogonal Correctors

Pressure equation is repeated according to number of non-orthogonal correctors. This may reduce the influence of bad computational mesh. For more details see e.g. [9], [1] or [3].

14.4.11 Number of Iterations on Rotor and Stator Part

User can specify how many sub-iterations are spend on rotor part and stator part during single iteration. It is recommended to use default option: one sub-iteration on rotor part and one sub-iteration on stator part. See file fvSolution.

14.4.12 Minimal and Maximal Values Options

During the computation, especially right at its start, some nonphysical oscillations of solution may appear. To make the solver more robust there may be minimal and maximal values specified for selected variables. See file fvSolution.

14.4.13 Turbulent Flow

There are a lot of turbulence models available in OpenFOAM. It is possible to list all of them, that can be used within simpleFoam with following command.

```
# simpleFoam -listTurbulenceModels
```

The same command also works with other solvers.

14.4.14 MRF (Multiple Reference Frame) Method for Rotation of Rotating Parts

For simulating of the rotation it is used Multiple Reference Frame (MRF) method. MRF adds source term (acceleration) to velocity (momentum) equations. Filter term is applied on volume cells cellZone. For more details see e.g. [9].

14.4.15 Message Passing Interface (MPI)

For parallel computations there is Message Passing Interface (MPI) standard used. For more details see e.g. [13].

14.5 Notes on gravitational potential and hydrostatic pressure

14.5.1 Potential of a homogeneous gravitational field

A homogeneous gravitational field is characterized by a constant vector field $\mathbf{g}$, the well known gravitational acceleration. Let us consider some volume or vessel filled with an incompressible fluid of density ρ. The gravitational field exerts a force on the fluid. Its force density $\mathbf{f}$ is given by the following well known formula

$$\mathbf{f} = \rho \mathbf{g} \quad . \tag{14.32}$$

It is a simple task to find a potential to (14.32). Let us remind, a potential (if it exists) of some given force field $\mathbf{F}$ is defined as a certain function ϕ satisfying the following equation

$$-\nabla \phi = \mathbf{F} \quad . \tag{14.33}$$

We can see that to the $\mathbf{f}$ given by (14.32) there exists a potential, let us denote it by φ, stating

$$\varphi\left(\mathbf{r}\right) = -\rho \mathbf{g} \cdot \mathbf{r} + C \quad , \tag{14.34}$$

where $C \in \mathbb{R}$ is a constant of integration[2] and $\mathbf{r}$ is a position vector[3].

14.5.2 Hydrostatic pressure

Let us consider an incompressible fluid at rest in the presence of a homogeneous gravitational field so it is supported by walls of some, possibly open, vessel. Inside a volume of the fluid there is the well known hydrostatic pressure, let us denote it by p. Assume the fluid has one free[4] part of its boundary, i.e. one part of its surface forms a level, i.e. plane or its part. Let us denote by $\mathbf{r}_0$ a position vector[5] of an arbitrary point of this plane. Hence for p holds

$$p\left(\mathbf{r}\right) = \rho \mathbf{g} \cdot \left(\mathbf{r} - \mathbf{r}_0\right) \quad . \tag{14.35}$$

Notice the relation (14.35) is valid inside the volume of the fluid only. If we need to extend its domain, it is necessary to assure zero values for p outside of the volume of the fluid. For instance, this is satisfied naturally, if we consider ρ a constant scalar field but vanishing outside the volume of the fluid.

14.5.3 Center of mass of a surface

Consider a two dimensional surface $\mathfrak{S}$, choose a system of coordinates and define a position vector $\mathbf{r}_{\mathfrak{S}}$ of its center of mass as following

$$\mathbf{r}_{\mathfrak{S}} = \frac{\iint_{\mathfrak{S}} \mathbf{r}\, \mathrm{d}S}{\iint_{\mathfrak{S}} \mathrm{d}S} \quad , \tag{14.36}$$

where $\mathbf{r}$ is a position vector of an element $\mathrm{d}S$.

[2] For it is the change or difference, whether finite or infinitesimal, of the potential that matters, not its actual value, we disregard such a constant unless stated explicitly.

[3] defining the position of a certain point with respect to origin of a system of coordinates

[4] i.e. not in contact with any wall of the vessel

[5] in a given system of coordinates

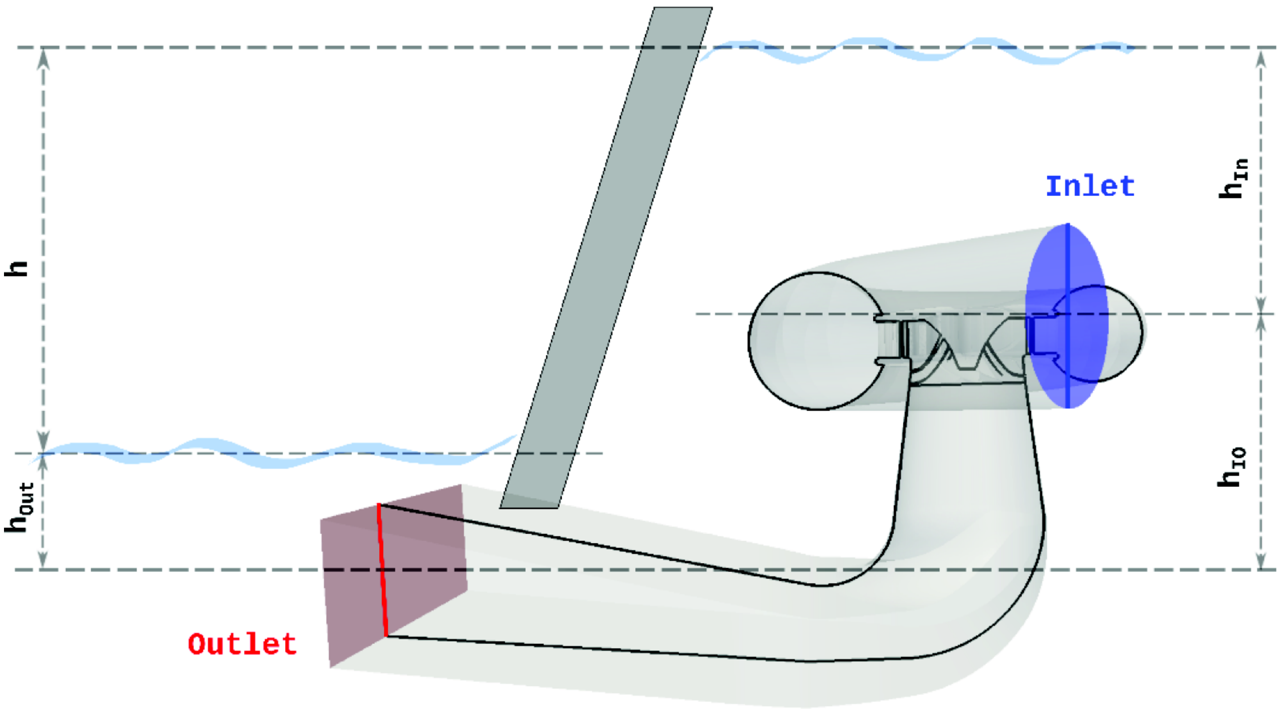


Figure 14.1: General water turbine sketch. Physical setting of a turbine casing with respect to a dam. Water levels indicated.

14.6 Water turbines - notes on calculations

Figure 14.1 represents a physical setting of a turbine casing with respect to a dam. Let us denote by h the head, i.e. the difference between heights of water levels in front of the dam and at the back of it. Vertical distance between the center of mass of the inlet surface of volute and the high water level is denoted by h_{In}. Vertical distance between the center of mass of the outlet surface of draft tube and the low water level is denoted by h_{Out}. Finally, vertical distance between the centres of mass of inlet and outlet surfaces is denoted by h_{IO}.

We can see there are several simple relations among quantities just defined

$$h + h_{Out} = h_{In} + h_{IO} \tag{14.37}$$

$$h = h_{In} + h_{IO} - h_{Out} \tag{14.38}$$

$$h - h_{IO} = h_{In} - h_{Out} \tag{14.39}$$

To simplify calculations we usually considers water an incompressible fluid with no phase changes possible and we solve the well known equations, the conservation laws the fluid obeys.

Conservation of linear momentum is expressed by the Navier-Stokes equation. In this case (incompressible fluid, steady-state) it can take the form

$$\mathbf{u} \cdot \nabla \mathbf{u} - \nu \Delta \mathbf{u} = -\tfrac{1}{\rho} \nabla p + \tfrac{1}{\rho} \mathbf{f} \quad , \tag{14.40}$$

where $\mathbf{u}$ is a velocity field, ν kinematic viscosity, ρ density, p pressure field and $\mathbf{f}$ is a force density of a gravitational field, given by (14.32).

Conservation of mass is expressed by the continuity equation. In this case it takes the simple form

$$\nabla \cdot \mathbf{u} = 0 \quad , \tag{14.41}$$

It remains to discuss boundary conditions for the equations mentioned above. For we usually do not know the inlet velocity field, we are about to prescribe pressure field for both inlet and outlet surface.

At the inlet surface a total pressure is known and it is equal to hydrostatic one. With the use of formula (14.35) we can write

$$p_{tot,In}(\mathbf{r}) = \rho \mathbf{g} \cdot (\mathbf{r} - \mathbf{r}_{0,H}) \quad , \tag{14.42}$$

where $\mathbf{r}_{0,H}$ is a position vector of an arbitrary point of the high water level (plane). Since total pressure is defined as a sum of static and dynamical pressure

$$p_{tot} = p + \tfrac{1}{2}\rho \mathbf{u}^2 \quad , \tag{14.43}$$

we can, by means of this definition (14.43) and formula (14.42), express p at the inlet

$$p_{In}(\mathbf{r}) = \rho \mathbf{g} \cdot (\mathbf{r} - \mathbf{r}_{0,H}) - \tfrac{1}{2}\rho \mathbf{u}^2(\mathbf{r}) \quad . \tag{14.44}$$

At the outlet the situation is a little bit complicated. After the flow (with high total pressure) exits the draft tube it mixes with water (with low total pressure) surrounding the draft tube. We usually take simplifying steps and assume the static pressure of exiting flow equals the hydrostatic pressure of surrounding water, i.e. we write

$$p_{Out}(\mathbf{r}) = \rho \mathbf{g} \cdot (\mathbf{r} - \mathbf{r}_{0,L}) \quad , \tag{14.45}$$

where $\mathbf{r}_{0,L}$ is a position vector of an arbitrary point of the low water level.

Described steps correspond to real physical setting. In a task like this we need to know not only the h, but even h_{In} (optionally h_{Out}), i.e. a position of the turbine with respect to the high or low water level. Quantity h_{IO} can be obtained from the given turbine casing geometry, of course.

14.6.1 Alternative formulation

Model setting

Since equation (14.40) is independent on actual pressure field values, but only on its gradients, we can formally simplify the boundary condition at the outlet by adding a certain constant C to the right hand side of the prescription (14.45) in order to get mean value of the outlet pressure equal to zero and not to hydrostatic pressure at the center of mass of the outlet surface. If we do this, we have to amend the inlet boundary condition (14.42) the same way, of course. We can find easily that the constant C has following value

$$C = \rho \mathbf{g} \cdot (\mathbf{r}_{0,L} - \mathbf{r}_{Outlet}) = -\rho g h_{Out} \quad , \tag{14.46}$$

where $\mathbf{r}_{Outlet}$ is a position vector of the centre of mass of the outlet surface. Hence prescription (14.45) changes to

$$p_{Out}(\mathbf{r}) = \rho \mathbf{g} \cdot (\mathbf{r} - \mathbf{r}_{Outlet}) \quad , \tag{14.47}$$

and prescription (14.42) to

$$p_{tot,In}(\mathbf{r}) = \rho\mathbf{g} \cdot (\mathbf{r} - \mathbf{r}_{0,H} + \mathbf{r}_{0,L} - \mathbf{r}_{Outlet}) \quad . \tag{14.48}$$

A mean value $\overline{p_{tot,In}}$ of the total pressure at the inlet surface now gives

$$\begin{aligned}
\overline{p_{tot,In}} &= \tfrac{1}{S_{Inlet}} \iint\limits_{Inlet} \rho\mathbf{g} \cdot (\mathbf{r} - \mathbf{r}_{0,H} + \mathbf{r}_{0,L} - \mathbf{r}_{Outlet})\,\mathrm{d}S && (14.49)\\
&= \tfrac{1}{S_{Inlet}} \iint\limits_{Inlet} \rho\mathbf{g} \cdot \mathbf{r}\,\mathrm{d}S + \rho\mathbf{g} \cdot (-\mathbf{r}_{0,H} + \mathbf{r}_{0,L} - \mathbf{r}_{Outlet}) && (14.50)\\
&= \rho\mathbf{g} \cdot (\mathbf{r}_{Inlet} - \mathbf{r}_{0,H} + \mathbf{r}_{0,L} - \mathbf{r}_{Outlet}) && (14.51)\\
&= \rho g (h - h_{IO}) \quad , && (14.52)
\end{aligned}$$

where $\mathbf{r}_{Inlet}$ is a position vector of the center of mass of the inlet surface and $g \equiv \|\mathbf{g}\|$.

We can see that in this model setting there is no need to know the position of a turbine to the water level. It suffices to know h and take the measurement of h_{IO}.

Model setting without an explicit use of g in the momentum equation

Let us take a look at the equation (14.40) as if its right hand side were known. Then we can introduce an alternative quantity, denoted by $p_{-\rho gh}$, to the static pressure p

$$p_{-\rho gh} = {}''p - \rho gh'' = p - \rho\mathbf{g} \cdot \mathbf{r} = p + \varphi \quad . \tag{14.53}$$

By means of (14.53) we can formally substitute equation (14.40) by the following one

$$\mathbf{u} \cdot \nabla\mathbf{u} - \nu\Delta\mathbf{u} = -\tfrac{1}{\rho}\nabla p_{-\rho gh} \quad , \tag{14.54}$$

for the right hand side takes the same values. This is the consequence of a fact that field (14.32) has potential and thus a change in potential energy of an arbitrary element of a fluid is not dependent on its path, but on its initial and final position only.

By introducing $p_{-\rho gh}$ we formally drop $\mathbf{g}$ out from the momentum equation. However, in order to obtain the same solution as in the previous settings, it is necessary to alter the prescription (14.48) by adding $\rho\mathbf{g} \cdot (\mathbf{r}_{Outlet} - \mathbf{r}_{Inlet})$ to its right hand side[6]. By doing this we obtain a new prescription, but this time for the quantity $p_{-\rho gh}$

$$p_{-\rho gh,tot,In}(\mathbf{r}) = \rho\mathbf{g} \cdot (\mathbf{r} - \mathbf{r}_{0,H} + \mathbf{r}_{0,L} - \mathbf{r}_{Inlet}) \quad . \tag{14.55}$$

If we calculate the mean value of (14.55), we obtain

$$\begin{aligned}
\overline{p_{-\rho gh,tot,In}} &= \tfrac{1}{S_{Inlet}} \iint\limits_{Inlet} \rho\mathbf{g} \cdot (\mathbf{r} - \mathbf{r}_{0,H} + \mathbf{r}_{0,L} - \mathbf{r}_{Inlet})\,\mathrm{d}S && (14.56)\\
&= \tfrac{1}{S_{Inlet}} \iint\limits_{Inlet} \rho\mathbf{g} \cdot \mathbf{r}\,\mathrm{d}S + \rho\mathbf{g} \cdot (-\mathbf{r}_{0,H} + \mathbf{r}_{0,L} - \mathbf{r}_{Inlet}) && (14.57)\\
&= \rho\mathbf{g} \cdot (\mathbf{r}_{0,L} - \mathbf{r}_{0,H}) && (14.58)\\
&= \rho gh \quad . && (14.59)
\end{aligned}$$

[6] This term is equal to the change of the potential energy density of an element of the fluid by its passage through the turbine (from the volute inlet to the draft tube outlet).

Prescription at the outlet surface remains formally the same as (14.47), but this time for the quantity $p_{-\rho gh}$

$$p_{-\rho gh,Out}(\mathbf{r}) = \rho \mathbf{g} \cdot (\mathbf{r} - \mathbf{r}_{Outlet}) \quad , \tag{14.60}$$

and hence its mean value is zero.

We can see that in this setting there is no need to know a position of a turbine with respect to the water level and there is also no need to even take the measurement of h_{IO}. It only suffice to know h, the head.

14.6.2 Conclusion

We have seen there are two basic approaches to water turbine calculations.

1. With $\mathbf{g}$ in the momentum equation
   (a) physical setting
   (b) model setting
2. Without explicit $\mathbf{g}$ in the momentum equation

Following table shows possible boundary conditions for the pressure variable

	pressure					
setting	inlet			outlet		
$\mathbf{g}$, physical	hTP	$=$	$\rho g h_{In}$	fMV	$=$	$\rho g h_{Out}$
$\mathbf{g}$, model	hTP	$=$	$\rho g(h - h_{IO})$	fMV	$=$	0
$\not{\mathbf{g}}$	hTP	$=$	$\rho g h$	fMV	$=$	0

Table 14.1: Boundary conditions

We note that *hTP* stands for *hydrostaticTotalPressure* boundary condition and values listed in the Table 14.1 for this type represent values of hydrostatic pressure in the centre of mass of the inlet surface. Whereas *fMV* stands for *fixedMeanValue* boundary condition and values listed in the Table 14.1 for this type represent values of static pressure in the centre of mass of the outlet surface.

We add that *pressureInletVelocity* boundary condition is prescribed for the inlet velocity field for all of the above settings, where the velocity magnitude is computed from the difference between total and static pressure and its direction is taken as a local normal to the inlet surface (usually planar). Also *zeroGradient* boundary condition is prescribed for the outlet surface for all of the above settings.

14.7 Interface between rotor and stator part

14.7.1 AMI vs. Mixing Plane

At the interface between stator and rotor part, for each variable one can prescribe either AMI (*Arbitrary Mesh Interface*, sometimes called *Frozen Rotor*) boundary condition or Mixing Plane boundary condition. AMI maps variable directly to the neighbour patch. Mixing Plane computes the variable average first and then maps just the average value to the neighbour patch. Both approaches have benefits and drawbacks to each other.

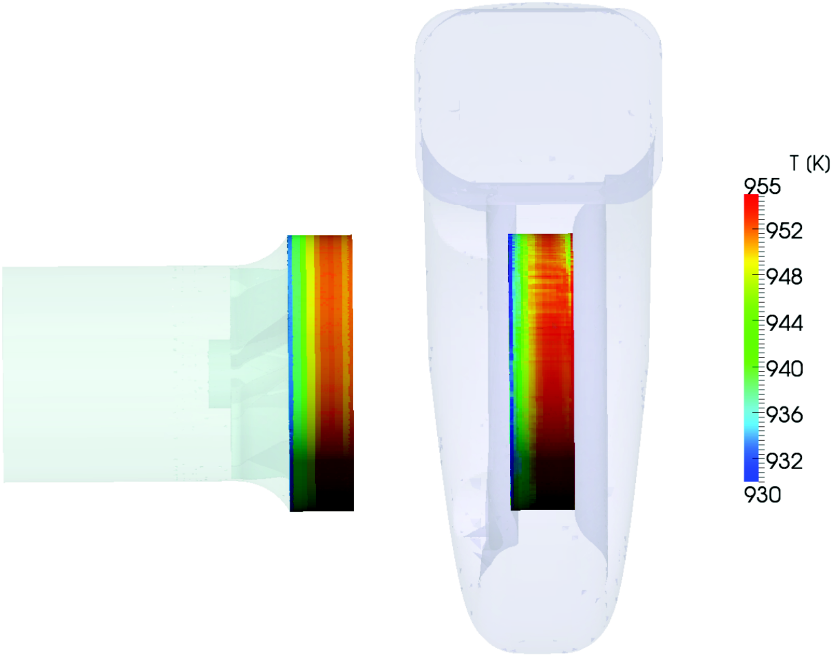


Figure 14.2: Radial turbine. Example of Mixing Plane Averaging from stator region to rotor region.

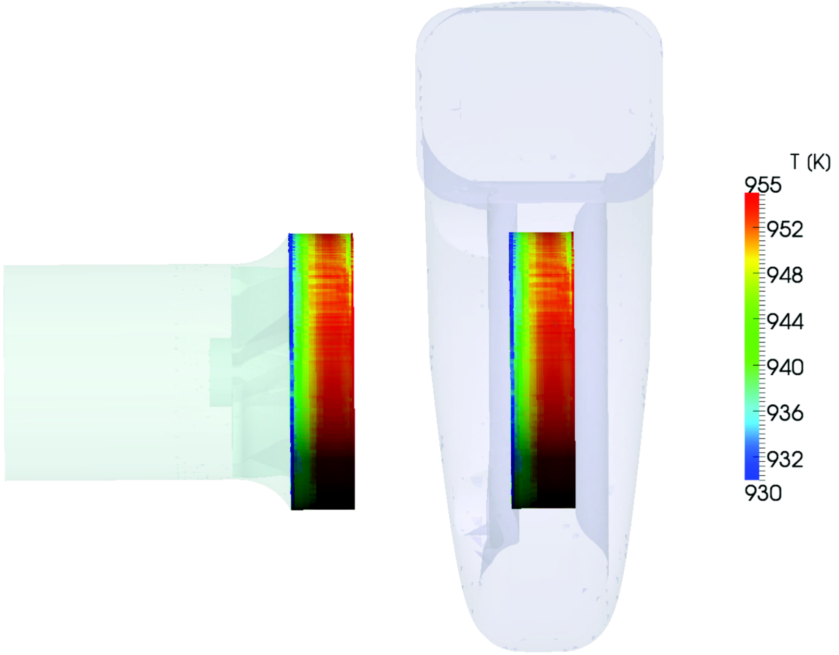


Figure 14.3: Radial turbine. Example of AMI interpolation from stator region to rotor region.

14.8 Cavitation Modeling in TCFD

TCFD in its current form use the following models for handling the cavitation:

- **Postprocessing based methods** provides parameters evaluated from a standard solvers:
  - **Cavitation risk** provides a "warning", when pressure drops too low and thus cavitation inception is likely.
  - **Cavitation index** is standard measure of cavitation.
  - **NPSHa**, **NPSHr** evaluates both net positive suction head available and required
- **Multiphase cavitation** is a full model for cavitation based on greenDyMSolver, see 14.1.4. The available liquid-vapour mass transfer models of the phase change in the transport equation are:
  - **Schnerr-Sauer** model
  - **Merkle** model [7]
  - **Kunz** model [8]

For the GUI setup of these models, see section 11.2.3.

[7] Not directly implemented, available on demand by scripting options.

[8] Not directly implemented, available on demand by scripting options.

14.8.1 Cavitation Risk, Cavitation Index and NPSH

Cavitation risk, Cavitation index, NPSHa (net positive suction head available) and NPSHr (net positive suction head reuiqred) are all triggered simultaneously by checking check box Cavitation risk in the PHYSICS : Multiphyiscs section of the TCFD menu, see figure 11.8. The methods work as a post-processing, so triggering them on/off does not affect the results.

In order for this feature to work properly, user must fill in proper Reference pressure , Reference density and Reference temperature . Reference temperature is used for calculation of the saturated vapor pressure using Antoine equation (see [18]) with coefficients taken from [19]. User can put a custom value of saturated vapour pressure by checking the Custom saturated vapour pressure check box.

Most incompressible calculations are done with zero pressure at the inelt or outlet. Reference pressure is way to offset the pressure to real physical value. Basically, there are two possible setups:

- Pressure at inlet or outlet is set to zero. Reference pressure must be set to the value the flow will experience in reality, e.g. atmospheric pressure.
- Pressure at inlet or outlet is set to the value the flow will actually experience. Reference pressure should be left zero.

Reference density is for proper conversion between dynamic and kinematic pressure.
Additionally, user can set:

- **Cavitating patches**: Patches to be used for evaluation of surface cavitation (default is set to all rotating patches).
- **Low pressure patch**: Typically set to inlet patch for pump simulations and outlet patch for turbine simulations.
- **High pressure patch**: Typically set to outlet patch for pump simulations and ibnlet patch for turbine simulations.

NOTE: Cavitation Index and NPSH evaluation is available for simulation types pump and hydroTurbine only.

Cavitation Risk

Cavitation risk evaluates saturated vapor pressure of water and then marks all cells, where the pressure drops under the saturated vapor pressure level. The result can be visualized by seeing cavitation field. Cells with value 0 are above saturated vapour pressure level, cells below that level have value 1. TCFD also evaluates volume of all cells with cavitation risk and surface of Cavitating patches with cavitation risk. Both those quantities can be seen in report, see example graph in figure 14.4.

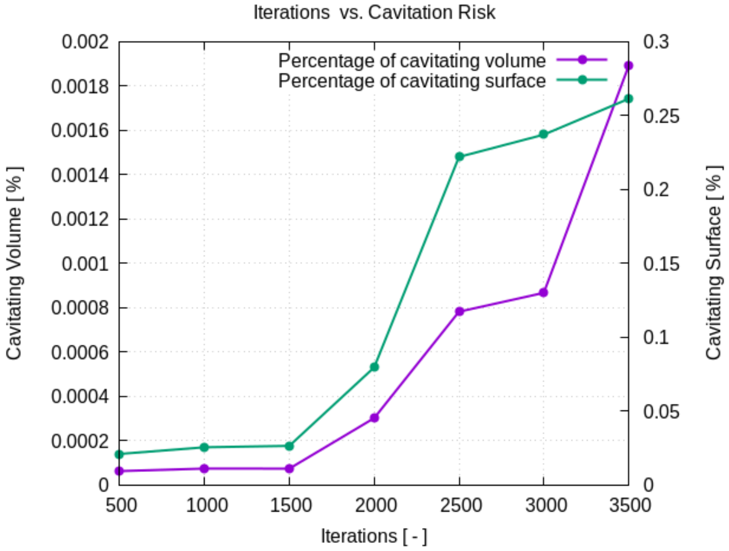


Figure 14.4: Example of Cavitation risk graph from report.

Cavitation Index

Cavitation index is standard measure of cavitation calculated by formula

$$\sigma = \frac{1}{gH}\left(\frac{p_{\text{lowP}} - p_{\text{vapour}}}{\rho} + \frac{\mathbf{u}^2_{\text{lowP}}}{2}\right) \tag{14.61}$$

where H is total head, g gravitational acceleration, p static pressure and $\mathbf{u}$ denotes velocity. The subscript $_{\text{lowP}}$ refers to value evaluated at Low pressure patch.

To get the value out of a numerical simulation the value p_{vapour} is replaced by p_{min} evaluated at Cavitating patches. Nevertheless, this value is highly sensitive to numerical outliers. Therefore, "Histogram Method" is applied. Instead of evaluating p_{min} a new value $p_{\text{histo-X\%}}$ is used. To get this value, X% of area with the lowest pressure values is cut off and the minimum is evaluated on remaining area:

$$\sigma_{0.5\%} = \frac{1}{gH}\left(\frac{p_{\text{lowP}} - p_{\text{histo-0.5\%}}}{\rho} + \frac{\mathbf{u}^2_{\text{lowP}}}{2}\right) \tag{14.62}$$

TCFD automatically evaluates $\sigma_{0.1\%}$, $\sigma_{0.5\%}$ and $\sigma_{1.0\%}$.

For a more detailed explanation of "Histogram Method", refer to [37].

NPSHa, NPSHr

NPSHa - Net Positive Suction Head available (see [19]) is an available head value at the suction side (e.g., on the inlet patch of the pump impeller). NPSHr - Net Positive Suction Head required is a required head at the suction side to avoid cavitation.

It is evaluated by the following formula

$$\text{NPSHa} = \frac{p_{\text{lowP}} + \frac{1}{2}\rho|\mathbf{u}_{\text{lowP}}|^2 - p_{\text{vapour}}}{\rho g}\,. \tag{14.63}$$

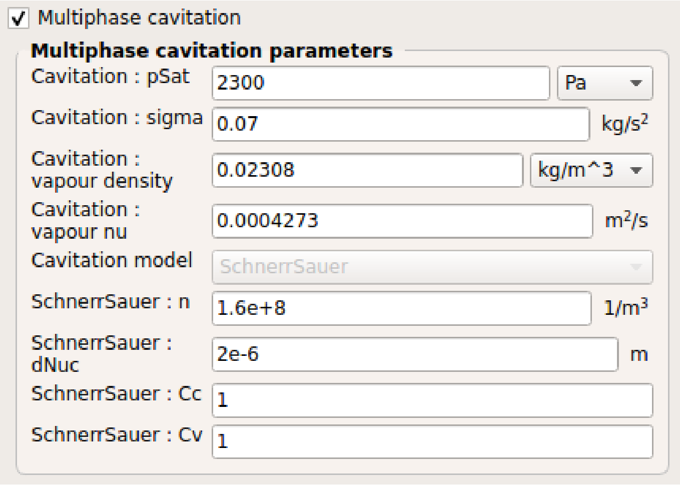
For NPSHr the "Histogram Method" is used in a similar way as for evaluation of cavitation index using the alternative formula to Eq. (14.63):

$$\mathrm{NPSHr}_{0.5\%} = \frac{p_{\mathrm{lowP}} + \frac{1}{2}\rho|\mathbf{u}_{\mathrm{lowP}}|^2 - p_{\mathrm{histo\text{-}0.5\%}}}{\rho g} .$$

TCFD automatically evaluates $\mathrm{NPSHr}_{0.1\%}$, $\mathrm{NPSHr}_{0.5\%}$ and $\mathrm{NPSHr}_{1.0\%}$.

14.8.2 Multiphase cavitation

Multiphase cavitation is triggered by Multiphase cavitation check box in PHYSICS : Multiphysics section of the TCFD menu, see figure 11.8. It is only available for transient calculations. Its options in TCFD GUI are shown again here in the figure 14.5.



☑ Multiphase cavitation

Multiphase cavitation parameters

Cavitation : pSat 2300 Pa

Cavitation : sigma 0.07 kg/s²

Cavitation : vapour density 0.02308 kg/m^3

Cavitation : vapour nu 0.0004273 m²/s

Cavitation model SchnerrSauer

SchnerrSauer : n 1.6e+8 1/m³

SchnerrSauer : dNuc 2e-6 m

SchnerrSauer : Cc 1

SchnerrSauer : Cv 1

Figure 14.5: TCFD – Multiphase cavitation options

As opposed to postprocessing based methods, multiphase cavitation uses different solver, so results may (and probably will) slightly differ, when its triggered on. In this case, only Reference pressure is important, setting is the same as in previous case.

The following cavitation models are available:

Schnerr-Sauer Model

This model (see [5]) is based on the bubble dynamics model using Rayleigh equation. The model assumes the vapour fraction as a function of the constant bubble radius having spherical shape without interactions. The following mass transfer formulas are defined:

- Mass transfer of evaporation ($p < p_v$):

$$R_e = \frac{\rho_l \rho_v}{\rho_m}\alpha_v(1-\alpha_v)\frac{3}{R_B}\sqrt{\frac{2}{3}\left(\frac{p_v - p}{\rho_l}\right)} \tag{14.64}$$

- Mass transfer of condensation ($p > p_v$):

$$R_c = \frac{\rho_l \rho_v}{\rho_m} \alpha_v (1 - \alpha_v) \frac{3}{R_B} \sqrt{\frac{2}{3} \left(\frac{p - p_v}{\rho_l} \right)} \tag{14.65}$$

- The bubble radius R_B is given by:

$$R_B = \sqrt[3]{\left(\frac{\alpha_v}{1 - \alpha_v} \frac{3}{4\pi} \frac{1}{n_0} \right)} \tag{14.66}$$

  where n_0 specifies the number of bubbles per volume of liquid.

The model constants are:

pSat (p_v)	Saturation vapor pressure,
sigma (σ)	Surface (interface) tension,
vapor density (ρ_v)	Density of vapor phase,
vapor ν	Kinematic viscosity of vapor phase,
n (n_0)	Bubble number density,
dNuc (d_{nuc})	Nucleation site diameter,
Cc (C_c)	Vaporisation rate coefficient,
Cv (C_v)	Condensation rate coefficient.

The nucleation site diameter is used to properly model evaporation. The term $\alpha_v(1 - \alpha_v)$ is replaced by the term $\alpha_{nuc}(1 - \alpha_v)$, where α_{nuc} is the nucleation site volume fraction and is evaluated as $\alpha_{nuc} = n_0 \pi d_{nuc}^3 / 6$.

Merkle Model

For the Kunz model the mass transfer is given by:

- Mass transfer of evaporation ($p < p_v$):

$$R_e = C_v \frac{\alpha_v \rho_v}{t_\infty \left(\frac{1}{2} \rho_l |\mathbf{u}_\infty|^2 \right)} (p - p_v) \tag{14.67}$$

- Mass transfer of condensation ($p > p_v$):

$$R_c = C_c \frac{\alpha_v (1 - \alpha_v) \rho_v}{t_\infty \left(\frac{1}{2} \rho_l |\mathbf{u}_\infty|^2 \right)} (p_v - p) \tag{14.68}$$

where $\mathbf{u}_\infty$ denotes the mean stream velocity and t_∞ is the mean flow time sale computed as $l / \mathbf{u}_\infty$, where l is the characteristic length scale.

Kunz Model

For the Kunz model the mass transfer is very similar to the Merkle model. R_e is the same and R_c is simplified so that the mass transfer of condensation is indepent of the magnitude of pressure. The mass transfers are given as:

- Mass transfer of evaporation ($p < p_v$):

$$R_e = C_v \frac{\alpha_v \rho_v}{t_\infty \left(\frac{1}{2}\rho_l |\mathbf{u}_\infty|^2\right)}(p - p_v) \tag{14.69}$$

- Mass transfer of condensation ($p > p_v$):

$$R_c = C_c \frac{(1 - \alpha_v)\alpha_v^2 \rho_v}{t_\infty} \tag{14.70}$$

where the model parameters are similar to the Merkle model.

It is not recommend to use both Multiphase cavitation and Cavitation risk at the same time. Cavitation risk is not suitable for solver used in Multiphase cavitation mode, so its results may not be reliable. However, since Cavitation risk is post-processing, other results than its own will not be affected.

Results are delivered in the form of fields alpha.water and alpha.vapor , that contain volumetric fraction of water and vapor respectively. Also, volume of all vapor is evaluated and shown in report, see example on figure 14.6. Beware, it shows only volume of vapor itself, not volume of all cells containing some vapor, so it is not directly comparable with volume of all cells with risk of cavitation inception from previous section.

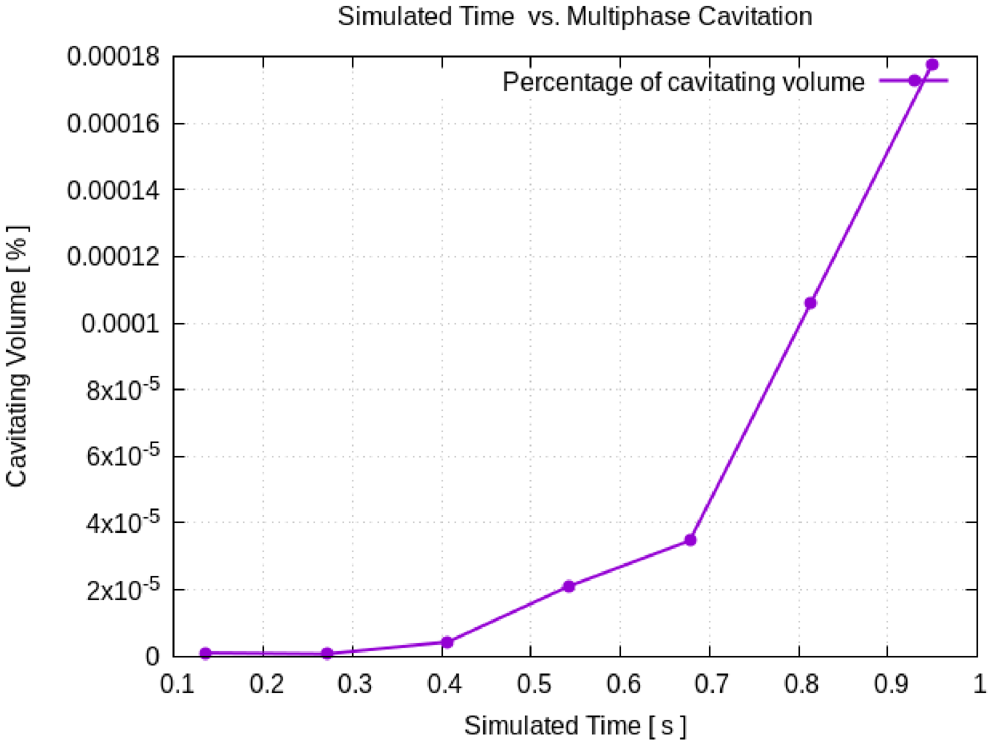


Figure 14.6: Example of Multiphase cavitation graph from report

Chapter 15

TCFD – Notes & Recommendations

15.1 General Notes & Recommendations

1. Circumferential and Meridional angle - tangent is chosen in such a way that axis, radial and tangent (in this order) form a right-handed coordinate system (look at Fig. 15.1). $\mathbf{u}$ is relative velocity, $\mathbf{u}_m$ is projection of $\mathbf{u}$ into meridional plane, $\mathbf{u}_r$ is projection of $\mathbf{u}$ into radial plane, $\mathbf{u}_t$ is projection of $\mathbf{u}$ into tangential plane. Meridional angle α is angle between axis and $\mathbf{u}_m$. It is positive, when $\mathbf{u}_m$ points away from axis and negative when $\mathbf{u}_m$ points toward axis. Axial circumferential angle β_a, is angle between tangent and $\mathbf{u}_t$. Radial circumferential angle β_r is angle between tangent and $\mathbf{u}_r$. It is positive when $\mathbf{u}_r$ points out (of the cylinder in the picture), it is negative when $\mathbf{u}_r$ points in.

2. Formulas for computing the values of turbulent quantities on inlet (see section 11.4.1):

   - Compute Turbulent kinetic energy k from Turbulent energy intensity I and Reference velocity U_{ref}
   $$k = \frac{3}{2}(U_{ref} I)^2, \tag{15.1}$$
   where typically $I = 0.05$ (5%) for internal flows and $I = 0.01$ (1%) for external flows.
   - Compute Turbulent energy dissipation rate ϵ from Turbulent kinetic energy k and Length scale l
   $$\epsilon = C_\mu^{\frac{3}{4}} \frac{k^{\frac{3}{2}}}{l}. \tag{15.2}$$
   - Compute Turbulent energy specific dissipation rate ω from Turbulent kinetic energy k and Length scale l
   $$\omega = \frac{k^{\frac{1}{2}}}{C_\mu^{\frac{3}{4}} l} \tag{15.3}$$
   where C_μ is an empirical constant of a turbulence model (default value $C_\mu = 0.09$).
   - Compute Turbulent energy dissipation rate ϵ from Turbulent kinetic energy k and Hydraulic diameter L:

     For fully developed turbulent pipe flow the turbulence length scale can be estimated from hydraulic diameter as $l = 0.07L$. Turbulent dissipation is then evaluated by Equation (15.2) or (15.3).

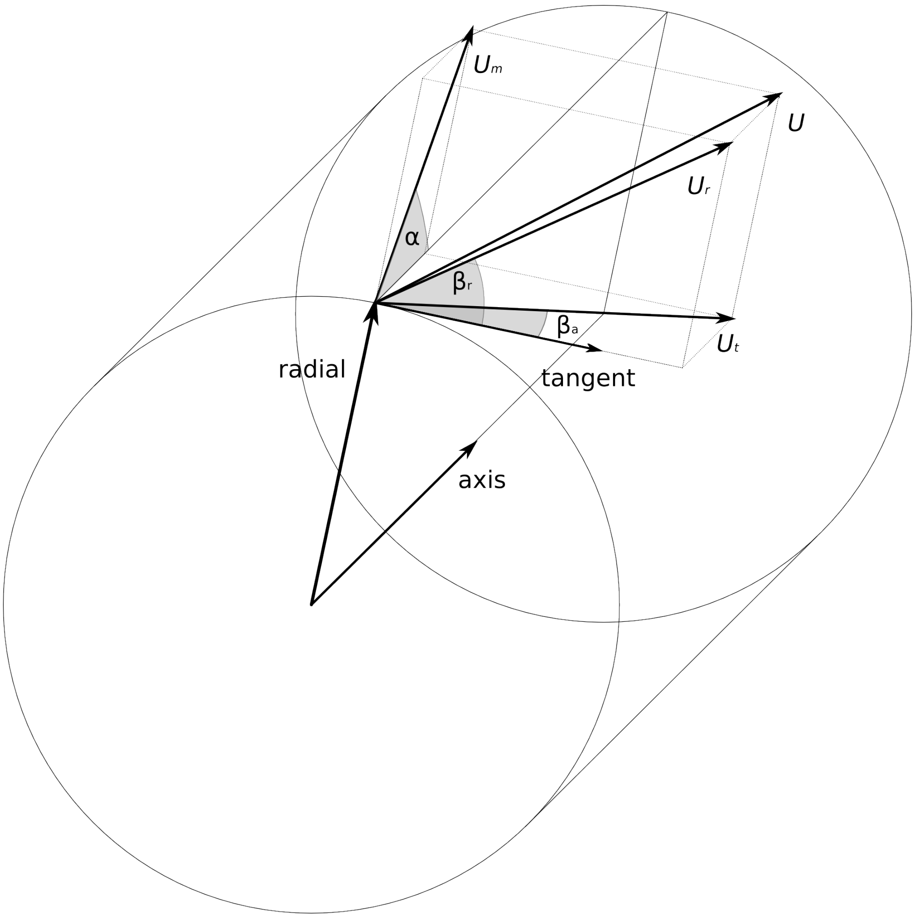


Figure 15.1: TCFD – Graphical representation of circumferential and meridional angle.

- Compute Turbulent energy dissipation rate ϵ from Turbulent kinetic energy k and Turbulent viscosity ratio μ_t/μ:

$$\epsilon = C_\mu \frac{\rho k^2}{\mu} \left(\frac{\mu_t}{\mu}\right)^{-1}. \tag{15.4}$$

- Compute Turbulent energy specific dissipation rate ω from Turbulent kinetic energy k and Turbulent viscosity ratio μ_t/μ:

$$\omega = \frac{\rho k}{\mu} \left(\frac{\mu_t}{\mu}\right)^{-1}. \tag{15.5}$$

- Compute Turbulent eddy viscosity $\tilde{\nu}$ from the Turbulent viscosity ratio

$$\tilde{\nu} = \mu \left(\frac{\mu_t}{\mu}\right). \tag{15.6}$$

- Compute ReThetaT (transition momentum thickness Reylonds number) Re_{θ_t} from Turbulent kinetic energy k and Reference velocity U_{ref}
  First compute turbulent intensity (here in %)

$$I = 100 \frac{\sqrt{2/3k}}{|U_{ref}|}, \tag{15.7}$$

then the $Re_{\theta t}$

$$Re_{\theta t} = \begin{cases} 1173.51 - 589.428I + \frac{0.2196}{I^2} & \text{if } I \leq 1.3, \\ \frac{331.5}{(I-0.5658)^{0.671}} & \text{if } I > 1.3. \end{cases} \tag{15.8}$$

15.2 Rescue list

If anything goes wrong - please go through this list item by item and check your case settings again. By the experience from the technical support, the vast majority of all the issues are caused by one of the following reasons:

1. Check, if the mesh is set up correctly - consult TMESH rescue list (section 9.7).
2. Do you see OpenGL errors after opening GUI in ParaView? Check technical note on page 15 (or 18 for Linux).
3. Is there enough free disk space available?
4. Is there enough memory available?
5. Check if the boundary conditions are reasonable (especially at the inlet and at the outlet).
6. Check of all the units (physical dimensions) of all quantities are correct.

7. All the interfaces between two components must fit perfectly. There is only one exception: in the periodic segment cases when using Mixing Plane interface condition, the interfaces may not fit in the circumferential direction. In all other cases the interfaces must fit perfectly.

8. Check the rotating reference frame(s). Do they have correct axis of rotation?

9. Check the speed of rotation and its units. Check the direction of rotation (right hand rule).

10. If using Mixing Planes interface condition, check the number of mixing planes (averaging intervals).

11. Check if all the components and walls that should rotate have assigned a rotating reference frame.

12. In the report check the y+ values.

13. Is the mesh OK visually?

14. Check if all the initial values for all quantities are reasonable.

15. Check the number of iterations or simulation time is reasonable - simulation has to be converged. Sometimes Convergence check utility may stop the simulation too soon. Check it in the report, the quantities should be steady (converged).

16. If your issue is still unresolved, please contact us at support@cfdsupport.com. The best way to do that is sharing the complete case to our technical support. If your case cannot be shared for any reason - please send us the detailed case description and zipped `TCFD/logRun` folder.

Chapter 16

Turbo Blade Post - graphical postprocessing

Turbo Blade Post is designed for postprocessing of rotating machinery. Both radial and axial machines. Pumps, hydro (water) turbines, compressors, turbochargers, propellers and many more.

It can be applied to postprocess the results of the TCAE calculation, respectively the OpenFOAM calculation in general.

Turbo Blade Post is product of company CFD Support s.r.o. (www.cfdsupport.com). It was especially created to enable an effective postprocessing of rotating machinery. Turbo Blade Post is a set of plugins for ParaView software.

ParaView offers the possibility to extend its functionality in several directions. This encompasses modifications to the GUI, implementation of new sources (i.e. generation of predefined curves and bodies), definition of new selection functions etc. The most useful category of plugins are the *Filters*.

The Turbo Blade Post include two these *Filters* - Meridional Average and Turbo Unwrap, which are described in this chapter. These are available in the *Filters* menu, see Fig. 16.1.

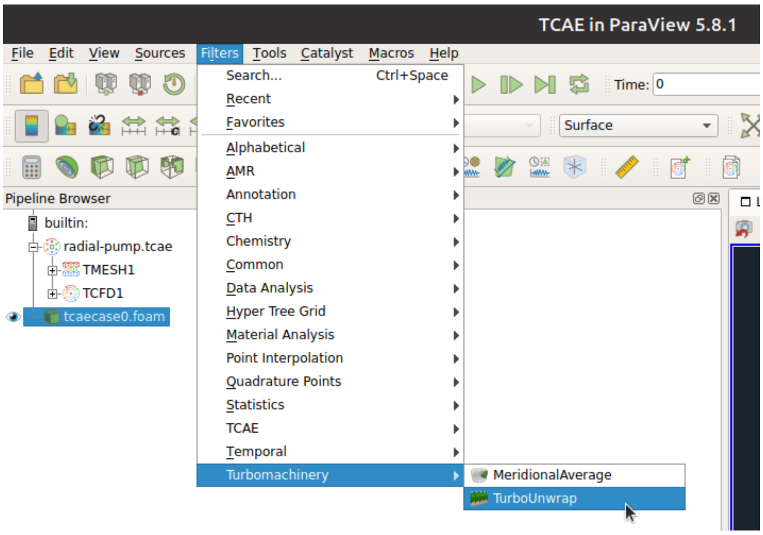


Figure 16.1: ParaView – filters menu

Note — The filters tend to disappear from their categories after use. This is a feature of ParaView, which moves the last used filter menu item into *Filters → Recent*. That list is, however, limited. Nevertheless, all filters are always accessible through the *Filters → Search* option.

16.1 Geometry & Mesh

All following Turbo Blade Post usage examples are presented on a numerical results from a simulation of an incompressible flow in a pump. The boundary geometry of the simulated volume of the pump is shown in the Fig. 16.2. The meshing and calculation has been done by TCAE. The resulting meshes are displayed in the figures 16.3 and 16.4. Numerical results are illustrated in the figures 16.5 and 16.6. These are classical visualisations from ParaView. Turbo Blade Post offers several new ways how to inspect the numerical data, which are presented in the following chapters.

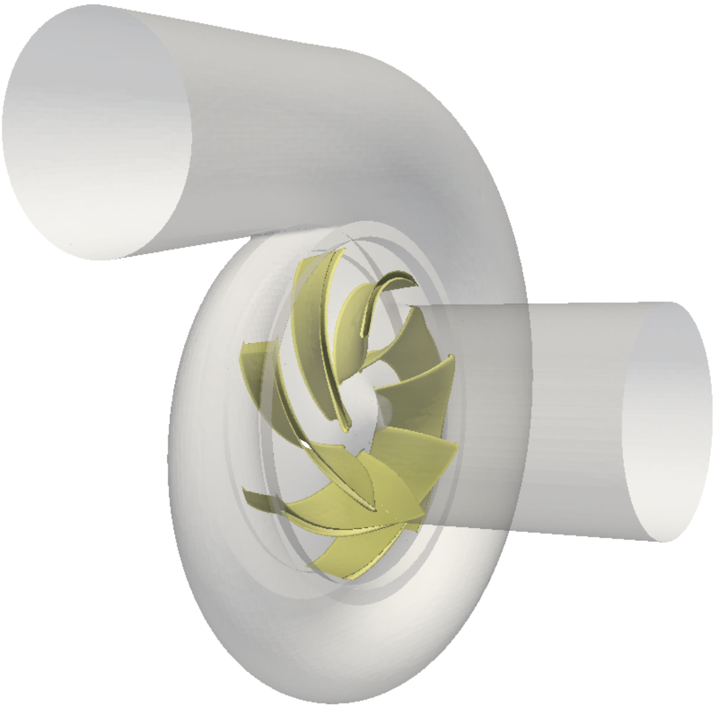


Figure 16.2: Geometry of the TCAE tutorial pump used in this examples.

16.2 Example: Meridional average

The aim of the first example is to meridionally average the scalar quantities in the vicinity of the blades, in the rotating (MRF) section of the fluid. Whereas the simple axial slice very often

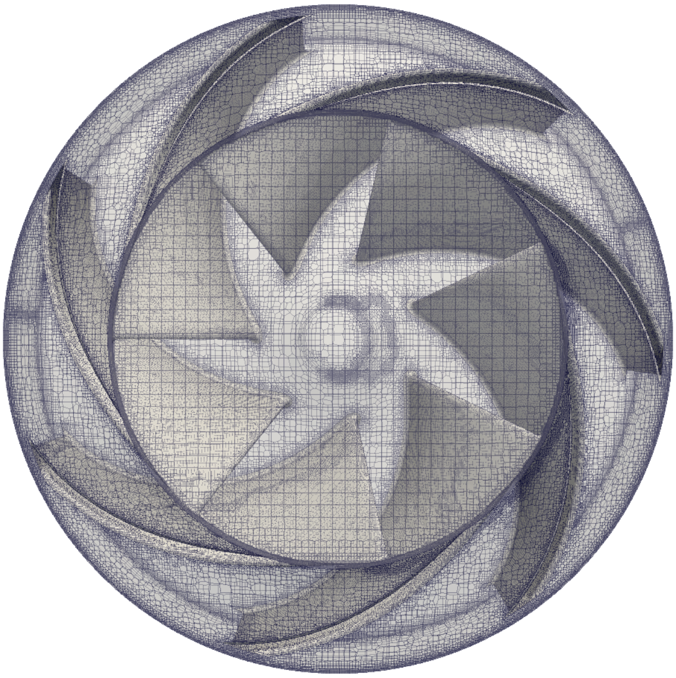


Figure 16.3: Computational mesh in the rotor MRF (rotating) zone as generated by TMESH using the snappyHexMesh mesher.

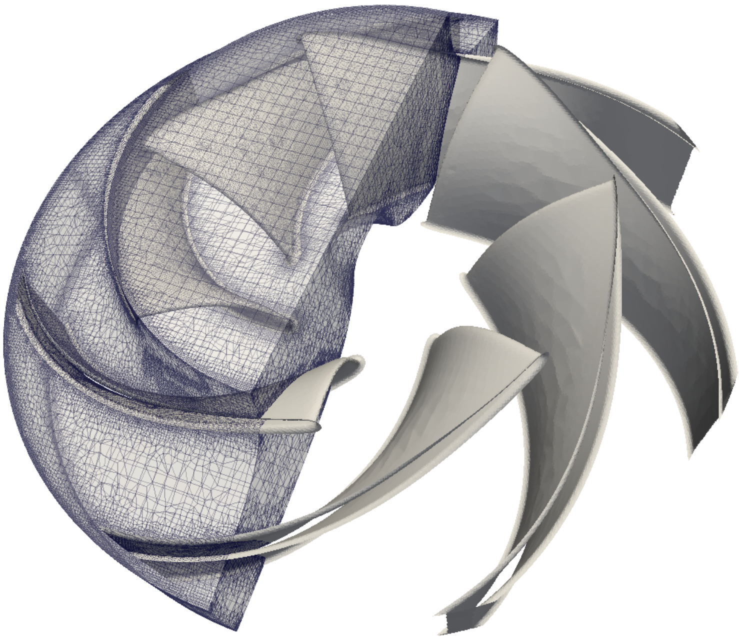


Figure 16.4: Computational mesh in the rotor MRF (rotating) zone as generated by TMESH using the snappyHexMesh mesher.

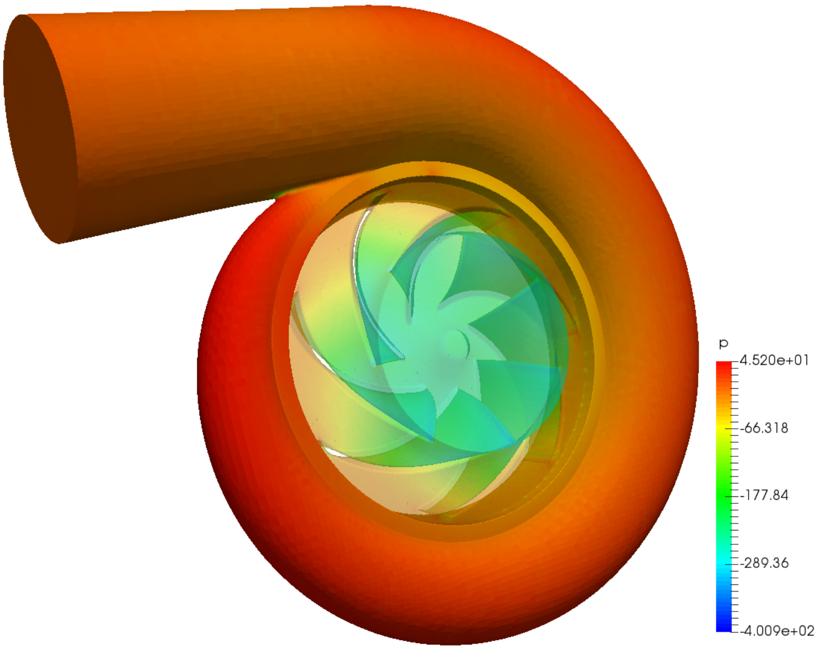


Figure 16.5: Static pressure field in the pump. The incompressible simulation has been done by TCFD.

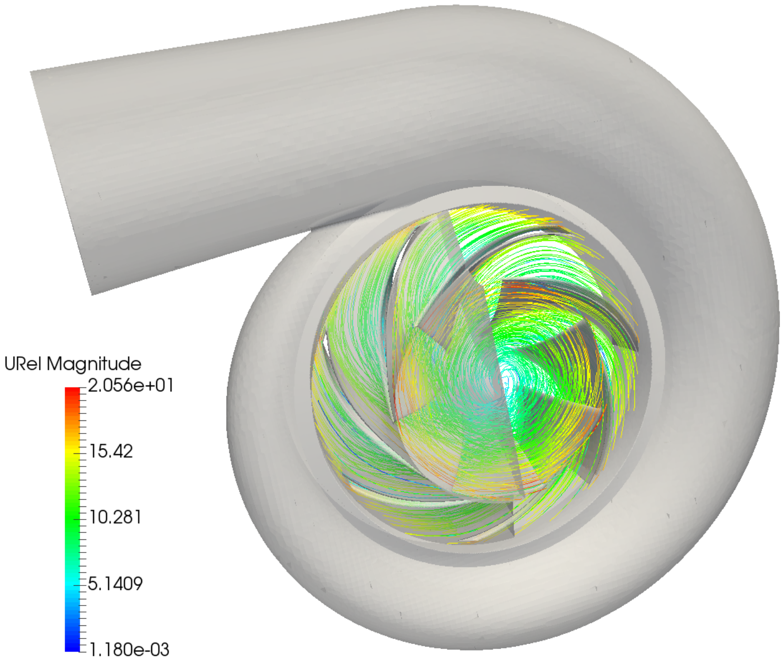


Figure 16.6: Relative velocity streamlines in the rotating part (MRF zone) of the mesh. The incompressible simulation has been done by TCFD.

cuts a blade, the Meridional Average avoids the holes by displaying circumferential average of values around the axis of rotation, see Fig. 16.7.

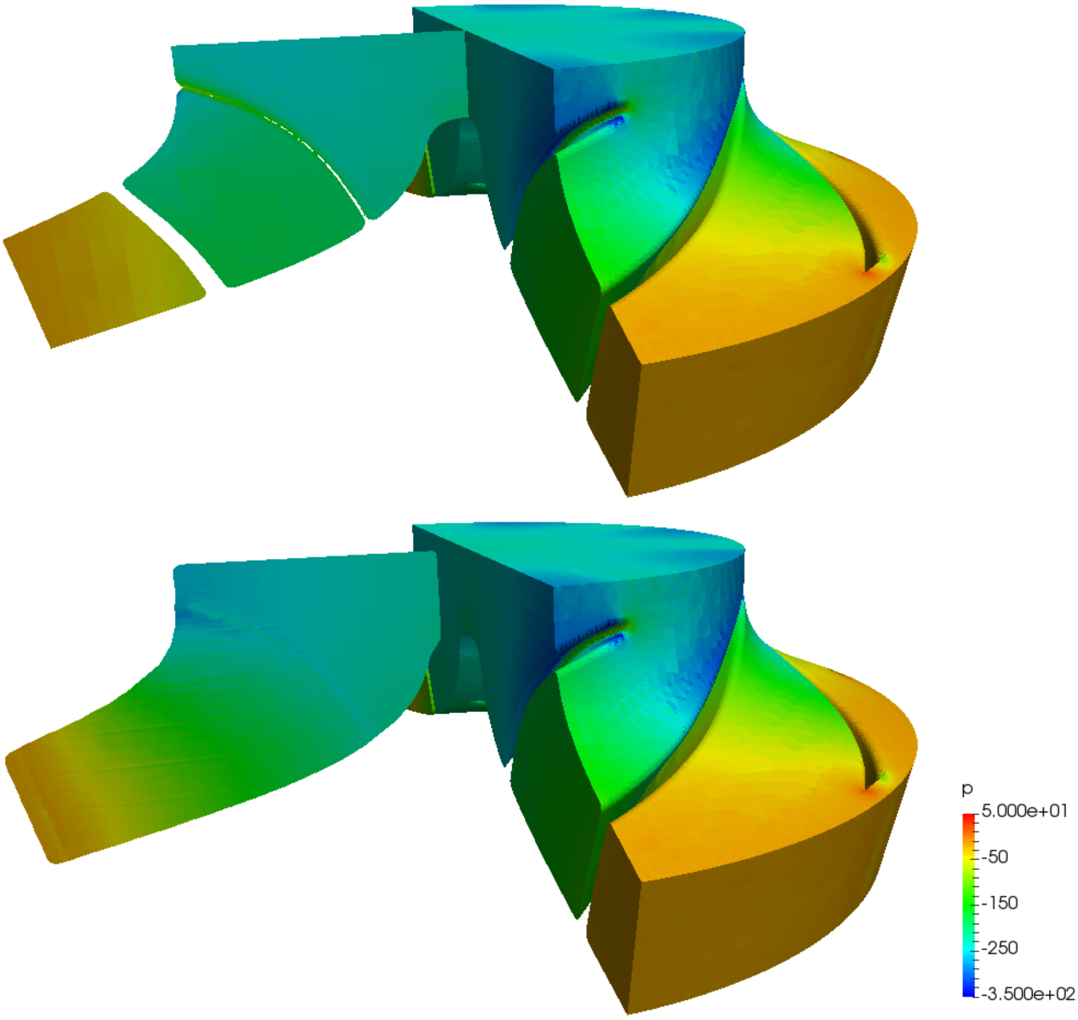
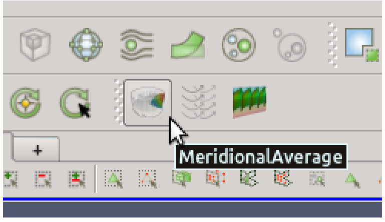


Figure 16.7: Comparison between a common ParaView *Slice* filter (top) and Turbo Blade Post Meridional Average filter (bottom) – application on TCFD calculation of incompressible flow in a pump.

16.2.1 Step by step guide

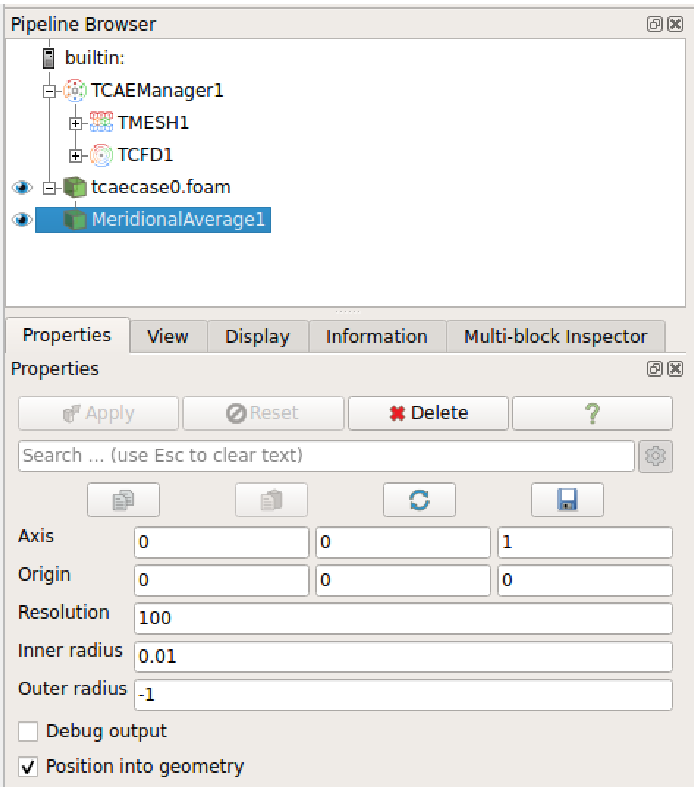
Step 1 — After performing TCAE calculation, load the results as described in 4.4.5. Select *Internal Mesh* in the *Mesh Regions*, the desired physical fields in the *Cell Arrays* and check on *Read zones*.

Step 2 — Meridional average can be calculated by an application of the filter Meridional Average, which is part of Turbo Blade Post. You should see the icon of the filter in the toolbar. First, if not, select the item with results in the *Pipeline Browser* (<case-name>.foam, Fig. 16.8).



Then, either use the button above, or select the filter Meridional Average in the *Filters → Turbomachinery* (or *Filters → Alphabetical*) menu, or use the search box from *Filters → Search*. This will add the filter into the *Pipeline Browser* (see Fig. 16.8).

Step 3 — The properties of the filter Meridional Average are shown in the Fig. 16.8. Some of the options are advanced and can be displayed using the *Toggle advanced properties* button (wheel symbol). The setup consists of a just a few numbers: (a) rotation axis, (b) axis origin, (c) clip out radius and (d) resolution. The "resolution" is the number of points of the resulting projection in radial or axial direction, whichever is larger. A non-zero "Inner radius" is necessary if the rotation axis pierces through the computational mesh, i.e. if there is no hole along the axis. In this tutorial the rotation axis is the axis z, the origin coincides with the coordinate system origin and we choose the clip out radius to be $0.01m$, as shown in the figure. Confirm the settings by pressing the *Apply* button. A non-zero "Outer radius" can be used to clip out some unwanted parts.



Axis 0 0 1
Origin 0 0 0
Resolution 100
Inner radius 0.01
Outer radius -1
□ Debug output
☑ Position into geometry

Figure 16.8: Turbo Blade Post – Properties of the filter Meridional Average, part of the Turbo Blade Post postprocessing toolbox. The parameters have default values, except for the non-zero "Clip out radius".

Step 4 — Once the filter completes, it will produce a projection as in the Fig. 16.9. By default, it shows the distance of individual points to the nearest surface (hub, shroud, inlet or outlet). However, all scalar fields have been averaged by the filter and are available in the field selection drop-down list in the main toolbar. The figures 16.10 and 16.11 show the averaged results for static pressure and relative velocity, respectively.

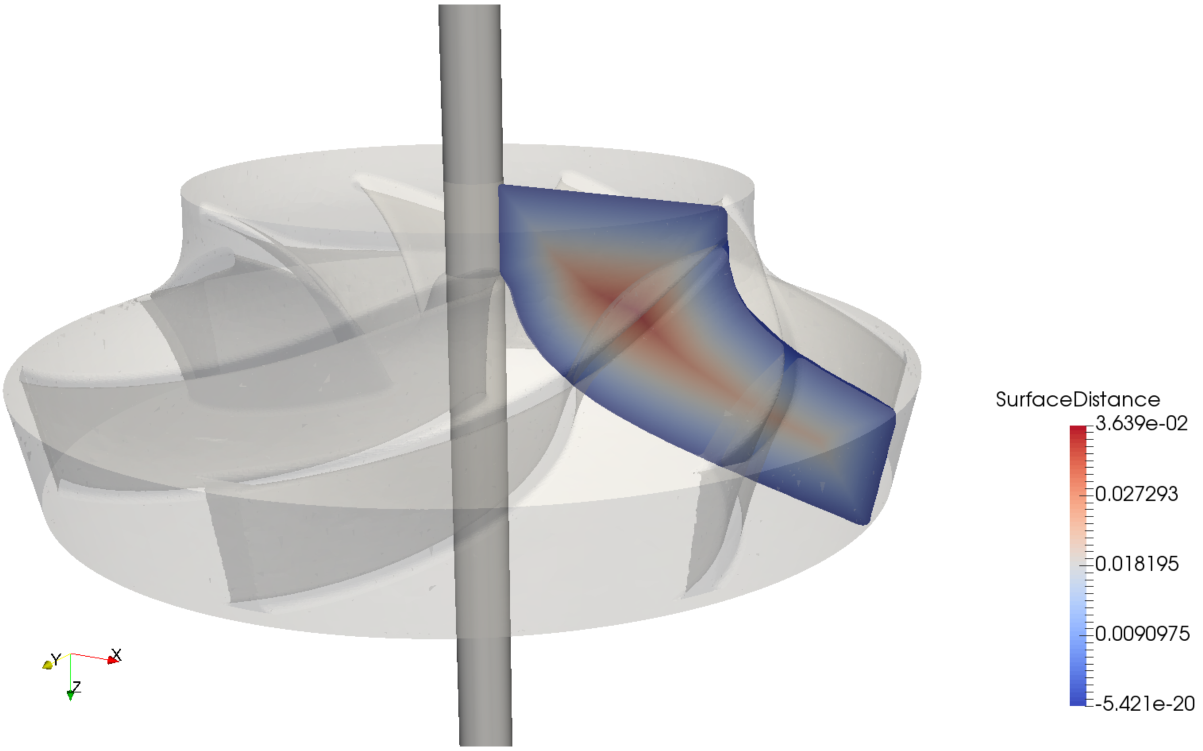


Figure 16.9: Resulting projection of the geometry constructed by the Turbo Blade Post toolset (Meridional Average filter) showing an auxiliary field, together with the original geometry and the cutting cylinder that corresponds to the chosen "Clip out radius".

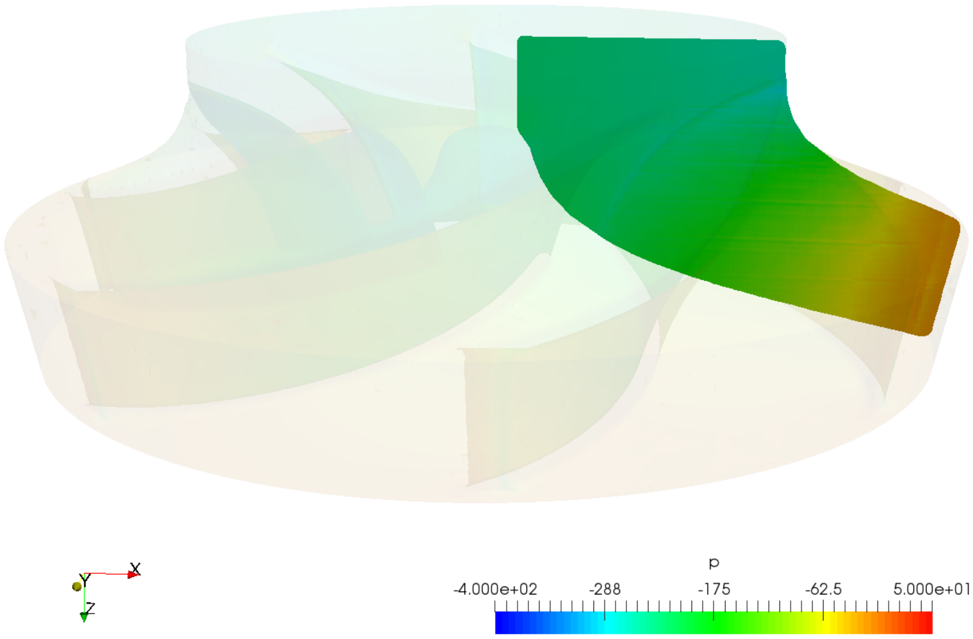


Figure 16.10: Meridional average of the static pressure for an incompressible flow in a pump constructed by Turbo Blade Post / Meridional Average. Results are from a calculation by TCFD.

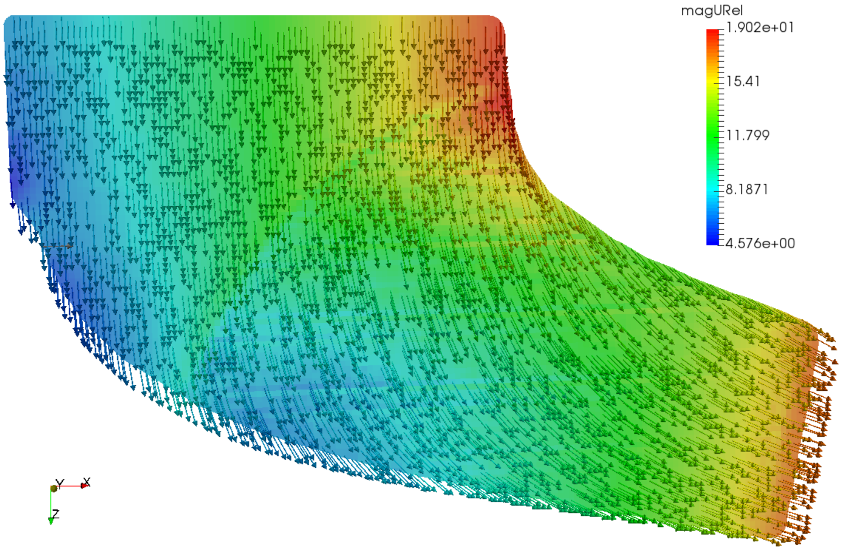


Figure 16.11: Meridional average of relative velocity restricted to the slice for an incompressible flow in a pump. Contructed by Turbo Blade Post / Meridional Average. Results are from a calculation by TCFD.

16.3 Example: Blade-to-blade view

The blade-to-blade view offers a unique perspective for an inspection of the flow between the blades at a fixed relative distance between hub and shroud surfaces. In Turbo Blade Post it can be generated in two steps: First, the cylindrical mesh of the rotating zone needs to be "unwrapped" into a rectangular block. Second, the unwrapped block has to be cut at the requested distance.

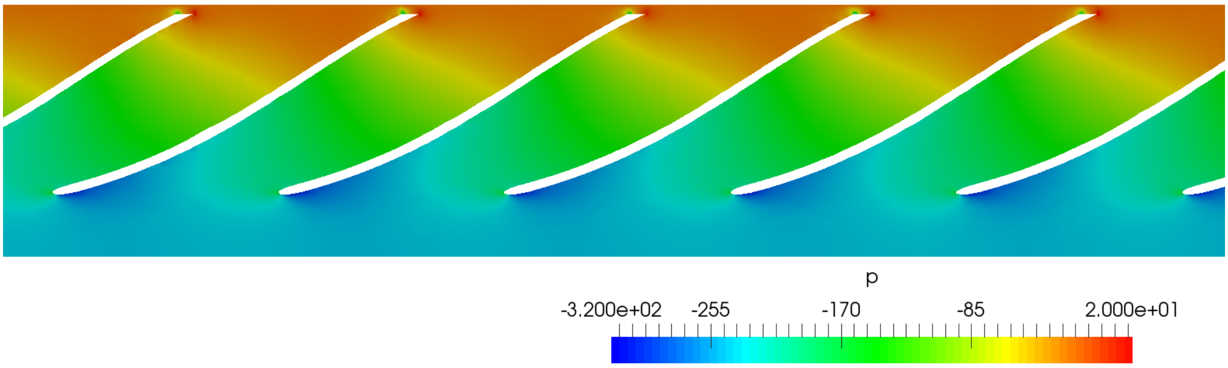
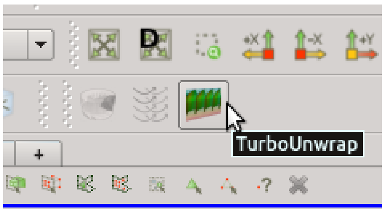


Figure 16.12: Blade-to-blade view constructed by the Turbo Blade Post toolset showing static pressure for an incompressible flow in pump. Results are from a calculation by TCFD.

16.3.1 Step by step guide

Step 1 — After performing TCAE calculation, load the results as described in 4.4.5, where in the *Mesh Regions* select *internalField*, hub and shroud patches, check on *Read zones* and *Copy data to cell zones* and choose the needed physical fields in the *Cell Arrays*.

Step 2 — The transformation from the cylinder - or disk-like rotating area (as shown in the Fig. 16.3 or 16.4) into the normalized rectangular block can be calculated by the filter Turbo Unwrap. You should see the icon of the filter in the toolbar.



Either use this button, or select the filter Turbo Unwrap in the *Filters* → *Turbomachinery* (or *Filters* → *Alphabetical*) menu, or use the search box from *Filters* → *Search*. This will add the filter into the *Pipeline Browser*.

Step 3 — The basic properties of the filter Turbo Unwrap are shown in the Fig. 16.13. When the advanced options are hidden, there are only a few options to define. First of all, it is necessary to choose the internal mesh, that will be transformed, and the hub and shroud patches, which will serve as a leaders to define the transformation. After the transformation is done, hub and shroud will be perfectly flat and parallel to each other, conformly deforming the mesh in between. If multiple mesh parts are selected in the "Unwrap mesh" window or multiple patches are selected in "Hub" or "Shroud" windows then they will be internally merged into a single entity before proceeding. In this example the mesh is well prepared and we can just select the three items that we loaded in the first step.

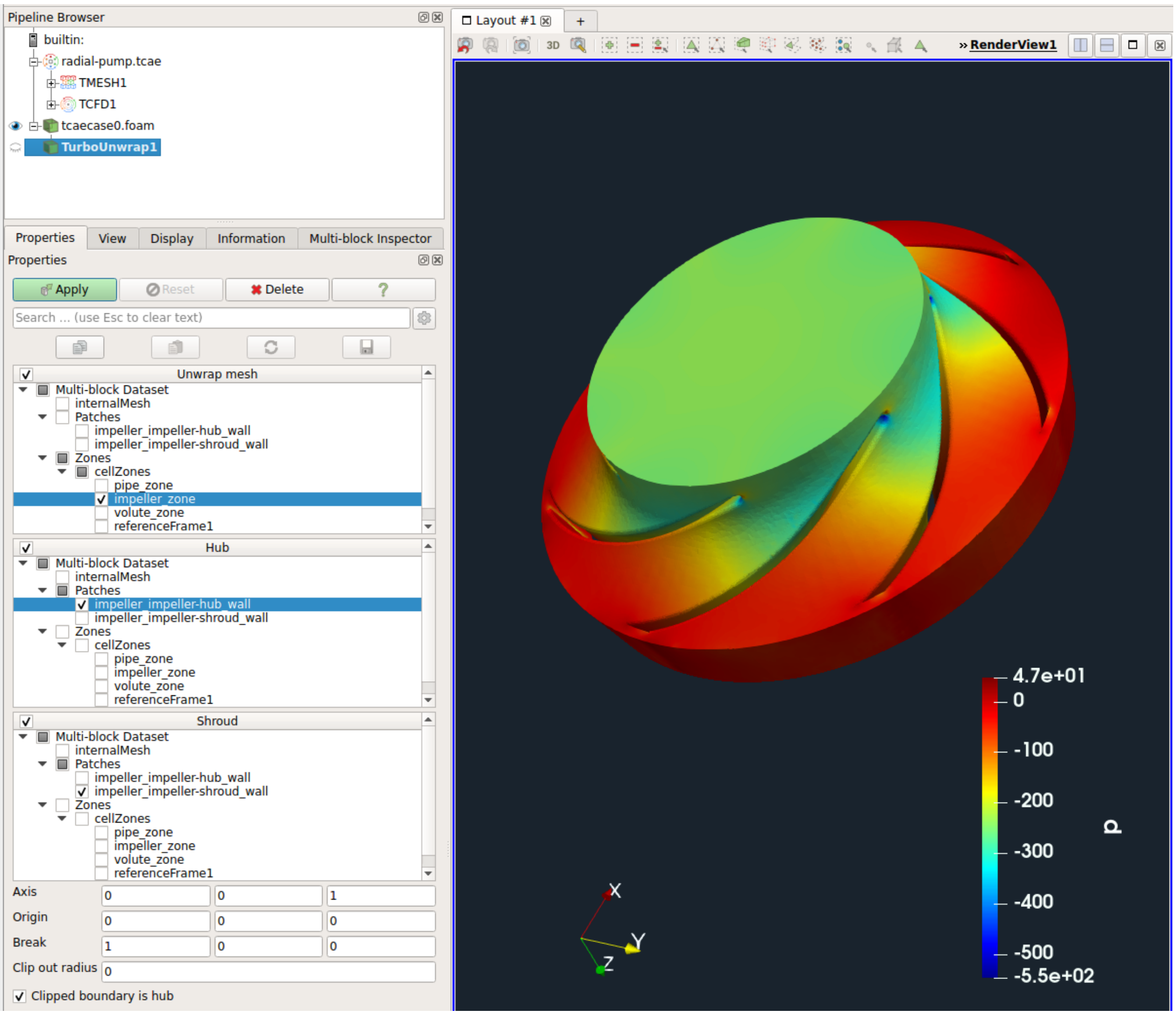
Step 4 — Set the direction and position of the rotating axis using the parameters "Axis" and "Origin". In our case we use z-axis, which is the default option.

Step 5 — Select the position of the breaking semiplane and cutting cylinder, see Fig. 16.14. In this case we have chosen the plane direction along the x-axis and the radius of the cylinder equal to $0.01m$, see Fig. 16.13.

Step 6 — Click on *Apply*. Particularly the cylinder clipping (and to less extent also the transformation of the mesh and of the vector fields) can be quite time consuming for large meshes. Some parts of the algorithm are parallelized and will automatically make use of multi-core machines. When the algorithm finishes, the result will look similarly to Fig. 16.16. The nature of the transformation is illustrated in the figures 16.14–16.17.

Step 7 — Unlike the scalar quantities (like pressure) the vector fields need to be transformed, too, when the mesh transforms. This is done automatically for the *cell* fields U and URel. As a by-product the filter also produces several other *cell* fields that can be used as an input to other filters: The local streamline vectors UStream and URelStream, and the *cell* field URelLIC, which is particularly well suited for uage in Line Integral Convolution (*"SurfaceLIC"*) representation. To use these fields in filters that request *point* fields, it is necessary to interpolate the data from points to cells, which is done as the final step using the filter *Cell data to point data* (can be found in *Filters* → *Alphabetical*). This filter has no important settings.

Step 8 — Now all that is necessary to obtain a specific blade-to-blade view is to use a standard *Slice* filter (z-normal) to cut through the block at the requested height. The coordinate $z = 0$ corresponds to the hub patch, $z = 1$ corresponds to the shroud patch, $z = 0.5$ corresponds to the surface in the middle between hub and shroud etc. This step is shown in the Fig. 16.18, the resulting cut displaying pressure is the Fig. 16.12, or showing the *SurfaceLIC* representation (with settings from 16.20) in the Fig. 16.19.



Pipeline Browser

- builtin:
  - radial-pump.tcae
    - TMESH1
    - TCFD1
  - tcaecase0.foam
    - TurboUnwrap1

Properties | View | Display | Information | Multi-block Inspector

Properties

Apply | Reset | ✖ Delete | ?

Search ... (use Esc to clear text)

☑ Unwrap mesh

- ▣ Multi-block Dataset
  - □ internalMesh
  - □ Patches
    - □ impeller_impeller-hub_wall
    - □ impeller_impeller-shroud_wall
  - ▣ Zones
    - ▣ cellZones
      - □ pipe_zone
      - ☑ impeller_zone
      - □ volute_zone
      - □ referenceFrame1

☑ Hub

- ▣ Multi-block Dataset
  - □ internalMesh
  - ▣ Patches
    - ☑ impeller_impeller-hub_wall
    - □ impeller_impeller-shroud_wall
  - □ Zones
    - □ cellZones
      - □ pipe_zone
      - □ impeller_zone
      - □ volute_zone
      - □ referenceFrame1

☑ Shroud

- ▣ Multi-block Dataset
  - □ internalMesh
  - ▣ Patches
    - □ impeller_impeller-hub_wall
    - ☑ impeller_impeller-shroud_wall
  - □ Zones
    - □ cellZones
      - □ pipe_zone
      - □ impeller_zone
      - □ volute_zone
      - □ referenceFrame1

Axis	0	0	1
Origin	0	0	0
Break	1	0	0
Clip out radius	0		

☑ Clipped boundary is hub

Figure 16.13: Turbo Blade Post – Basic parameters of the filter Turbo Unwrap.

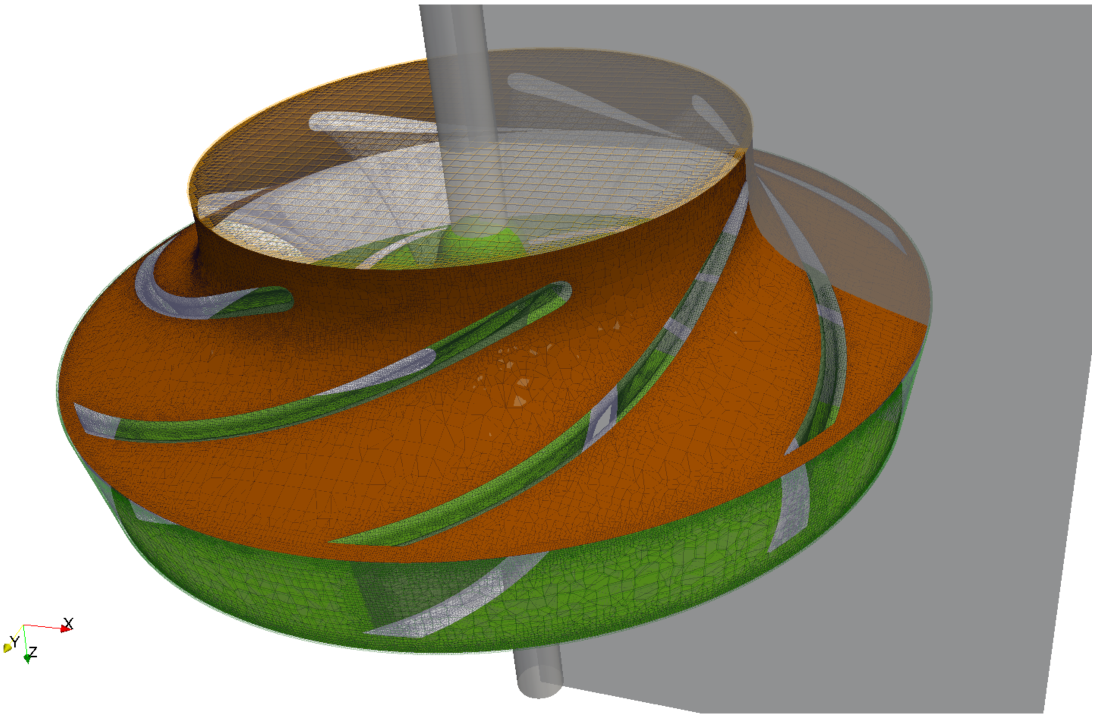


Figure 16.14: Rotating area of the simulated pump's volume before the application of the filter Turbo Unwrap. The patches are coloured here to make them easily differentiable in the following Fig. 16.16. Hub patch (bottom) is in solid green, shroud patch (top) in solid orange, inflow (top) is orange wireframe and outflow (bottom) green wireframe. The figure also contains the clipping cylinder corresponding to the parameter Clip out radius and the cutting semiplane corresponding to the direction specified by the parameter Break.

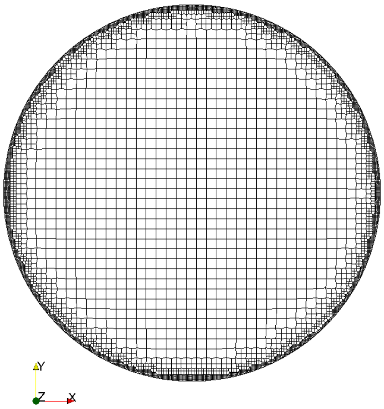


Figure 16.15: Detail of the inflow interface mesh structure of the pump test case before the application of Turbo Blade Post plugin Turbo Unwrap.

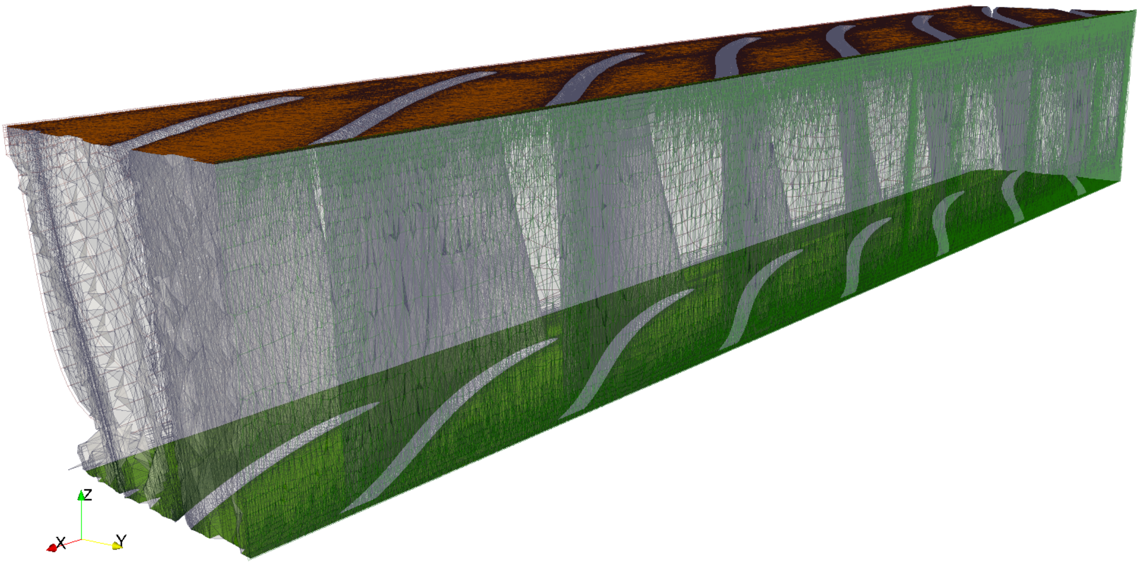


Figure 16.16: Rotating area of the simulated pump's volume after the application of the filter Turbo Unwrap. The hub and shroud patches are now parallel, the new z coordinate runs from hub ($z = 0$) to shroud $z = 1$, the normalized circular coordinate is transformed to $x \in (0, 2\pi)$ and the remaining coordinate that is orthogonal to both, "along the stream", is mapped to $y \in (0, 1)$. The green hub patch is now in the bottom, orange shroud on top, inflow is left and behind (not visible here) and the green outflow is in the front and to the right. The two additional sides of the block arose by cutting the mesh by a semiplane specified by the revolution axis and the parameter Break. It is not a straight cut through the cells, rather the cells that would be split by the semiplane are removed completely.

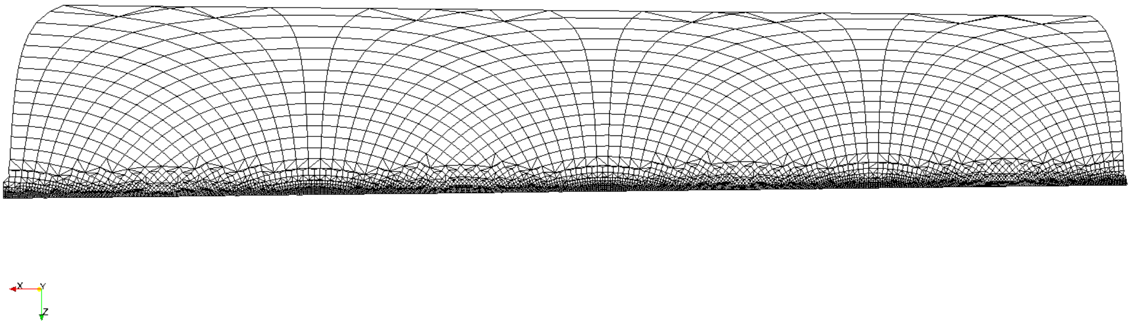
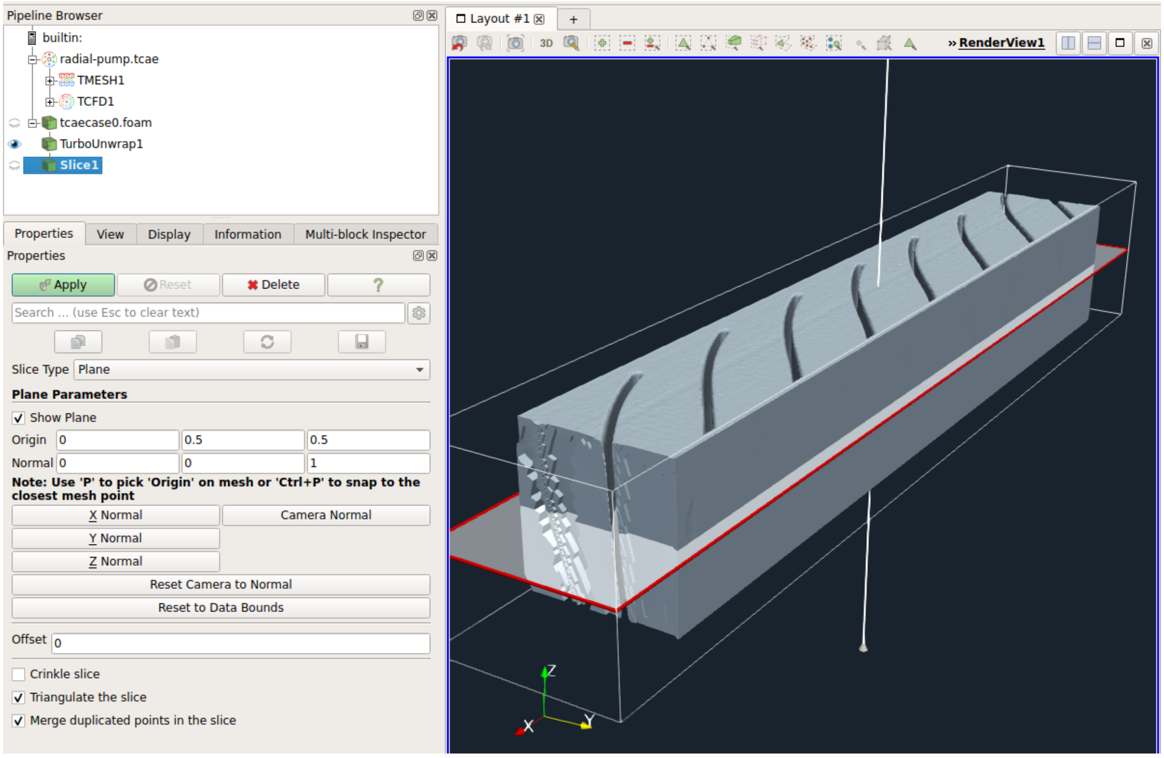


Figure 16.17: Detail of the inflow interface mesh structure of the pump test case after the application of Turbo Blade Post plugin Turbo Unwrap.



Pipeline Browser

- builtin:
  - radial-pump.tcae
    - TMESH1
    - TCFD1
  - tcaecase0.foam
    - TurboUnwrap1
    - Slice1

Properties

Slice Type: Plane

Plane Parameters

☑ Show Plane

Origin	0	0.5	0.5
Normal	0	0	1

Note: Use 'P' to pick 'Origin' on mesh or 'Ctrl+P' to snap to the closest mesh point

Offset: 0

☐ Crinkle slice

☑ Triangulate the slice

☑ Merge duplicated points in the slice

Figure 16.18: Turbo Blade Post – Construction of the blade-to-blade view using *Slice* filter from the rotor block unwrapped by Turbo Unwrap.

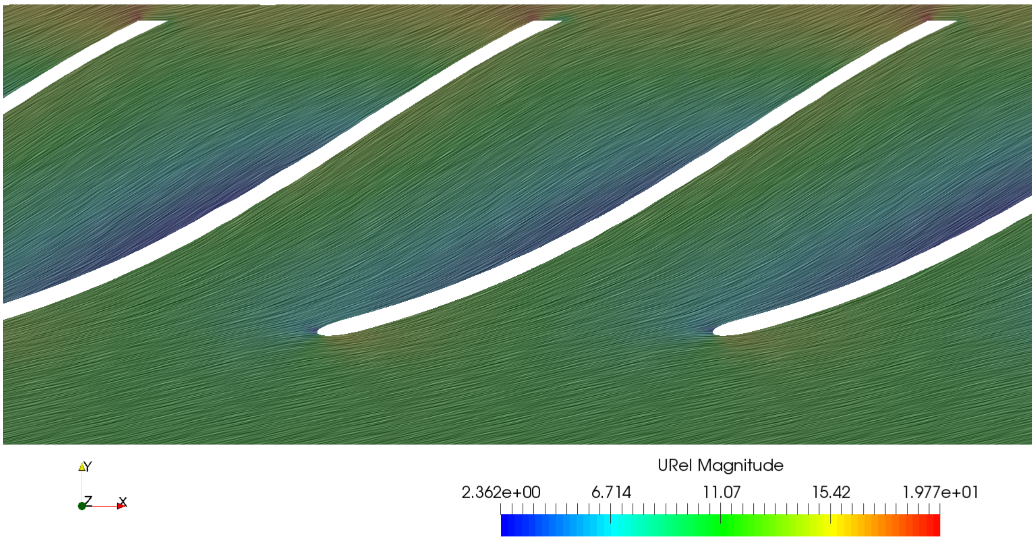
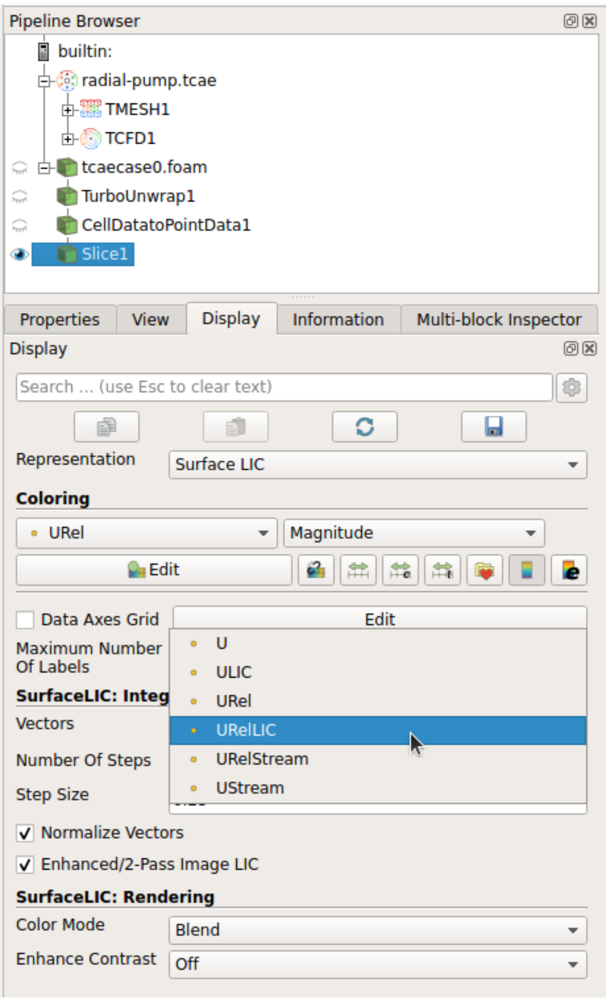


Figure 16.19: Blade-to-blade view from the rotor block unwrapped by Turbo Blade Post plugins showing the Line Integral Convolution (LIC) representation of the flow based on the relative velocity. The results are from a calculation by TCFD.



Pipeline Browser

- builtin:
  - radial-pump.tcae
    - TMESH1
    - TCFD1
  - tcaecase0.foam
  - TurboUnwrap1
  - CellDatatoPointData1
  - Slice1

Properties | View | Display | Information | Multi-block Inspector

Display

Search ... (use Esc to clear text)

Representation: Surface LIC

Coloring

URel | Magnitude

Edit

□ Data Axes Grid | Edit

Maximum Number Of Labels

SurfaceLIC: Integ

Vectors

- U
- ULIC
- URel
- URelLIC
- URelStream
- UStream

Number Of Steps

Step Size

☑ Normalize Vectors

☑ Enhanced/2-Pass Image LIC

SurfaceLIC: Rendering

Color Mode: Blend

Enhance Contrast: Off

Figure 16.20: Surface Line Integral Convolution setup after application of Turbo Unwrap + *Cell Data To Point Data* filters. The coloring is taken from the magnitude of the vector field *URel*, the directions and sizes from *URelLIC*.

16.4 Example: Pressure around the blade

For industrial applications of the CFD simulations it is often necessary to visualize the pressure distribution around the blade, when cut at a specific height. This is a direct analogy of the pressure profiles used in aeronautics when simulating wing profiles etc. In Turbo Blade Post this can be achieved by application of a sequence of ParaView filters on the blade patches.

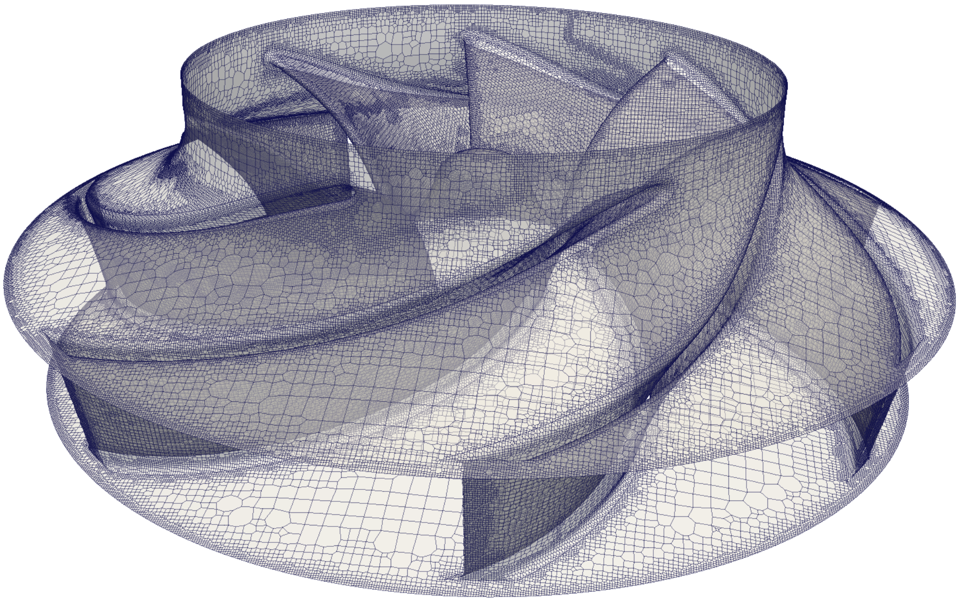
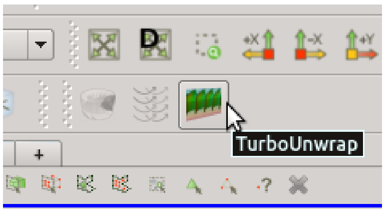


Figure 16.21: Input geometry (hub, shroud and blade patches) for the Turbo Blade Post / Turbo Unwrap filter for construction of the around-the-blade pressure profile.

16.4.1 Step by step guide

Step 1 — After performing TCAE calculation, load the results as described in 4.4.5. It is not necessary to load the whole mesh; choose only the blade walls and the hub and shroud patches (result is displayed in the Fig. 16.21).

Step 2 — First, the blades need to be transformed from the cylinder- or disk-like arrangement to a straight rectangular block. This is done by the filter Turbo Unwrap . You should see the icon of the filter in the toolbar.



Either use this button, or select the filter Turbo Unwrap in the *Filters* → *Turbomachinery* (or *Filters* → *Alphabetical*) menu, or use the search box from *Filters* → *Search*. This will add the filter into the "Pipeline browser".

Step 3 — The basic properties of the filter Turbo Unwrap are shown in the Fig. 16.22. When the advanced options are hidden, there are only a few options to define. First of all, it is necessary to choose the blade wall patch, that will be transformed, and the hub and shroud patches, which will serve as a leaders to define the transformation. After the transformation is done, hub and shroud will be perfectly flat and parallel to each other, conformly deforming the mesh in between. If multiple mesh parts are selected in the "Unwrap mesh" window or multiple patches are selected in "Hub" or "Shroud" windows then they will be internally merged into a single entity before proceeding. In this example the mesh is well prepared and we can just select the three items that we loaded in the first step.

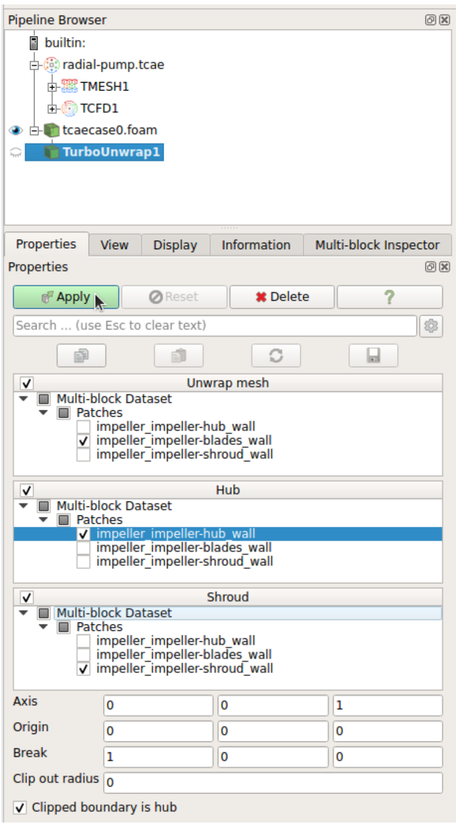
Step 4 — Set the direction and position of the rotating axis using the parameters "Axis" and "Origin". In our case we use z-axis, which is the default option.

Step 5 — Unlike in the case of the blade-to-blade view, when constructing the pressure profile it is mostly not necessary to specify a non-zero "Clip out radius", because the blades do not reach all the way to the axis in the middle. We will leave the parameter having its default zero value.

Step 6 — Click on *Apply*. The transformation should be relatively fast, because the sufrace mesh of the blades is orders of magnitude easier to process than the full volume mesh. It may be necessary to zoom in or out a little (depending on the geometry) to make the result fit to window. Outcome of this step is shown in the Fig. 16.23, where the blades are coloured by pressure.

Step 7 — Having the blades transformed we can now cut them at a specific height (z-axis) using the filter *Slice*. This will result in several two-dimensional intersection contours.

Step 8 — Add the filter *Plot Data* from *Filters* → *Alphabetical* or using *Filters* → *Search*. Unselect all fields but pressure (see Fig. 16.24). Above the field selection box use "Points_Y" as the "X Array Name". This will use points' Y coordinates as the data for the horizontal axis. Below the field selection box use *None* as *Line Style* and *Circle* as *Marker Style*. This will only show one bullet per a mesh point, making the result independent on the order of the projected points. Now press *Apply*. You should obtain a similar figure to 16.24.



Pipeline Browser

- builtin:
  - radial-pump.tcae
    - TMESH1
    - TCFD1
  - tcaecase0.foam
    - TurboUnwrap1

☑ Unwrap mesh
- Multi-block Dataset
  - Patches
    - □ impeller_impeller-hub_wall
    - ☑ impeller_impeller-blades_wall
    - □ impeller_impeller-shroud_wall

☑ Hub
- Multi-block Dataset
  - Patches
    - ☑ impeller_impeller-hub_wall
    - □ impeller_impeller-blades_wall
    - □ impeller_impeller-shroud_wall

☑ Shroud
- Multi-block Dataset
  - Patches
    - □ impeller_impeller-hub_wall
    - □ impeller_impeller-blades_wall
    - ☑ impeller_impeller-shroud_wall

Axis	0	0	1
Origin	0	0	0
Break	1	0	0
Clip out radius	0		

☑ Clipped boundary is hub

Figure 16.22: Turbo Blade Post – Settings of the Turbo Blade Post / Turbo Unwrap filter for transformation of blades of the pump.

Note — The plot will contain data from all blades. As the blades are equivalent, it doesn't hurt the visualization. If just a single blade profile was required, it would be necessary to separate one of the contours using a pair of the *Clip* filters .

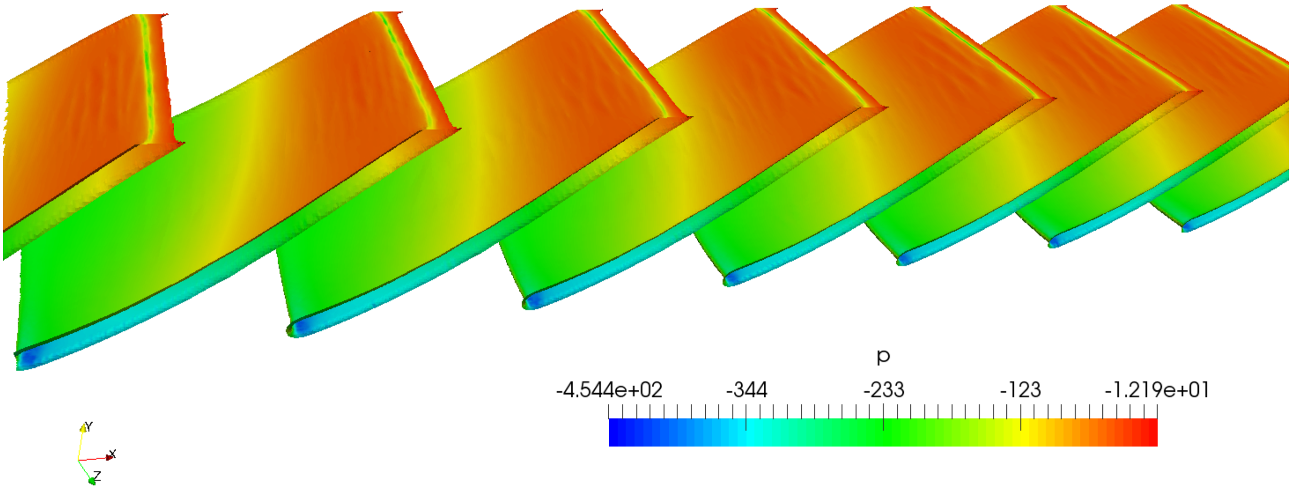
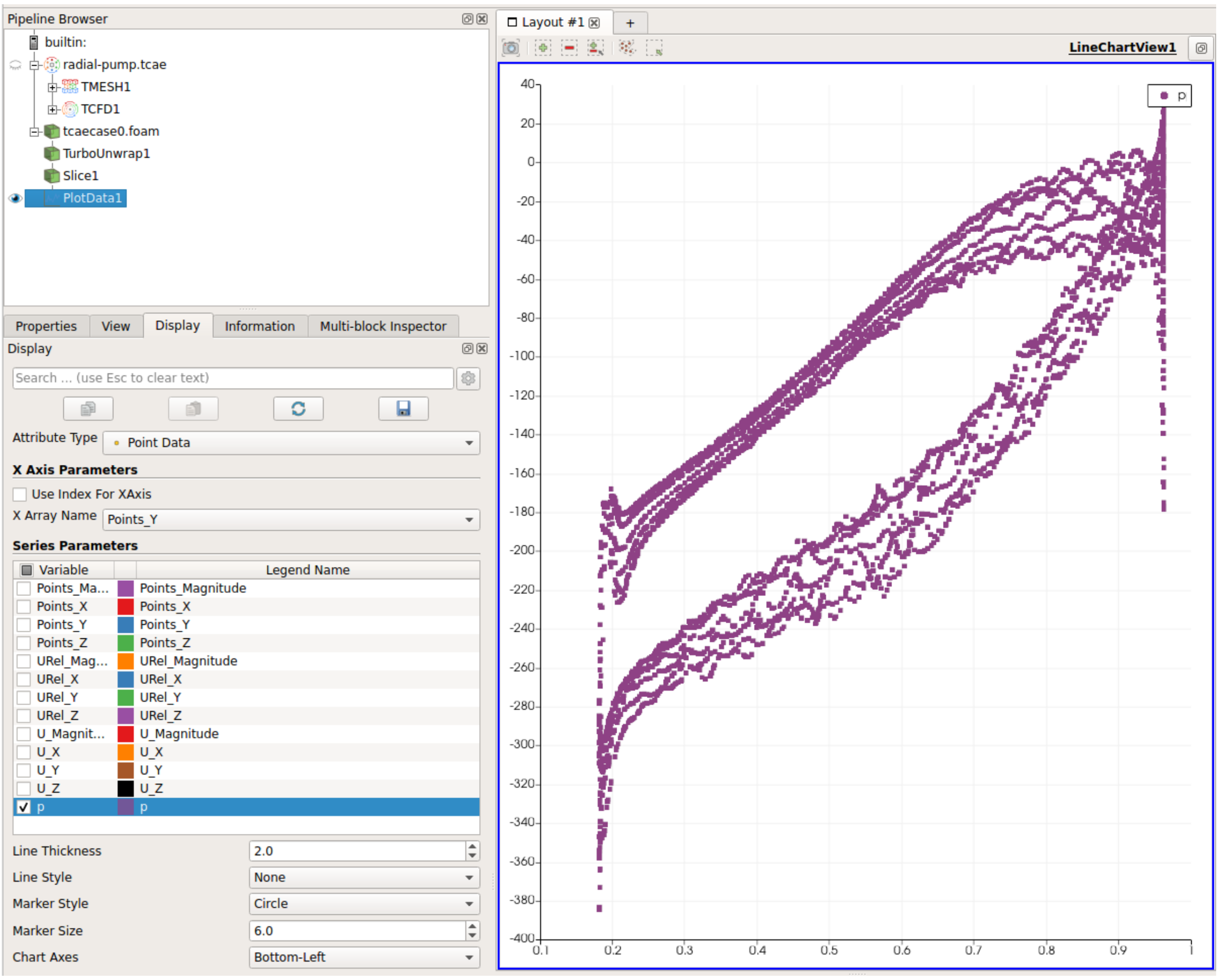


Figure 16.23: Blades of a pump transformed by Turbo Blade Post – Turbo Unwrap to a straight arrangement and displaying pressure field computed by TCFD.



Pipeline Browser

- builtin:
  - radial-pump.tcae
    - TMESH1
    - TCFD1
  - tcaecase0.foam
    - TurboUnwrap1
    - Slice1
    - PlotData1

Properties | View | Display | Information | Multi-block Inspector

Display

Search ... (use Esc to clear text)

Attribute Type: Point Data

X Axis Parameters

☐ Use Index For XAxis

X Array Name: Points_Y

Series Parameters

☐ Variable		Legend Name
☐ Points_Ma...	🟪	Points_Magnitude
☐ Points_X	🟥	Points_X
☐ Points_Y	🟦	Points_Y
☐ Points_Z	🟩	Points_Z
☐ URel_Mag...	🟧	URel_Magnitude
☐ URel_X	🟦	URel_X
☐ URel_Y	🟩	URel_Y
☐ URel_Z	🟪	URel_Z
☐ U_Magnit...	🟥	U_Magnitude
☐ U_X	🟧	U_X
☐ U_Y	🟫	U_Y
☐ U_Z	⬛	U_Z
☑ p	🟪	p

Line Thickness: 2.0

Line Style: None

Marker Style: Circle

Marker Size: 6.0

Chart Axes: Bottom-Left

Figure 16.24: Turbo Blade Post – Pressure distribution around the blade for a given height (i.e. relative distance from the hub to the shroud) constructed using the Turbo Unwrap, *Slice* and *Plot Data* filters. The horizontal axis shows the transformed Y axis, which corresponds to the normalized inlet-to-outlet direction position (for given hub-to-shroud distance and angular position). The vertical axis displays the value of the pressure as computed by TCFD.

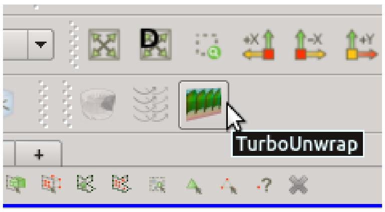
16.5 Example: Blade pressure and suction side contours

Turbo Blade Post allows also an easy selection of a single blade from the full complex of all blades, which might be difficult otherwise due to a strong curvature of the blades. The method works by transforming the blade patches from their original cylinder- or disk-like arrangement to a straight arrangement just like in the previous example and by isolating the chosen blade.

16.5.1 Step by step guide

Step 1 — After performing TCAE calculation, load the results as described in 4.4.5. It is not necessary to load the whole mesh; choose only the blade walls and the hub and shroud patches (result is displayed in Fig. 16.21).

Step 2 — First, the blades need to be transformed from the cylinder- or disk-like arrangement to a straight rectangular block. This is done by the filter Turbo Unwrap. You should see the icon of the filter in the toolbar.



Either use this button, or select the filter Turbo Unwrap in the *Filters* → *Turbomachinery* (or *Filters* → *Alphabetical*) menu, or use the search box from *Filters* → *Search*. This will add the filter into the *Pipeline Browser* .

Step 3 — The basic properties of the filter Turbo Unwrap are shown in the Fig. 16.22. When the advanced options are hidden, there are only a few options to define. First of all, it is necessary to choose the blade wall patch, that will be transformed, and the hub and shroud patches, which will serve as a leaders to define the transformation. After the transformation is done, hub and shroud will be perfectly flat and parallel to each other, conformly deforming the mesh in between. If multiple mesh parts are selected in the "Unwrap mesh" window or multiple patches are selected in "Hub" or "Shroud" windows then they will be internally merged into a single entity before proceeding. In this example the mesh is well prepared and we can just select the three items that we loaded in the first step.

Step 4 — Set the direction and position of the rotating axis using the parameters "Axis" and "Origin". In our case we use z-axis, which is the default option.

Step 5 — Unlike in the case of the blade-to-blade view , when constructing the pressure profile it is mostly not necessary to specify a non-zero "Clip out radius", because the blades do not reach all the way to the axis in the middle. We will leave the parameter having its default zero value.

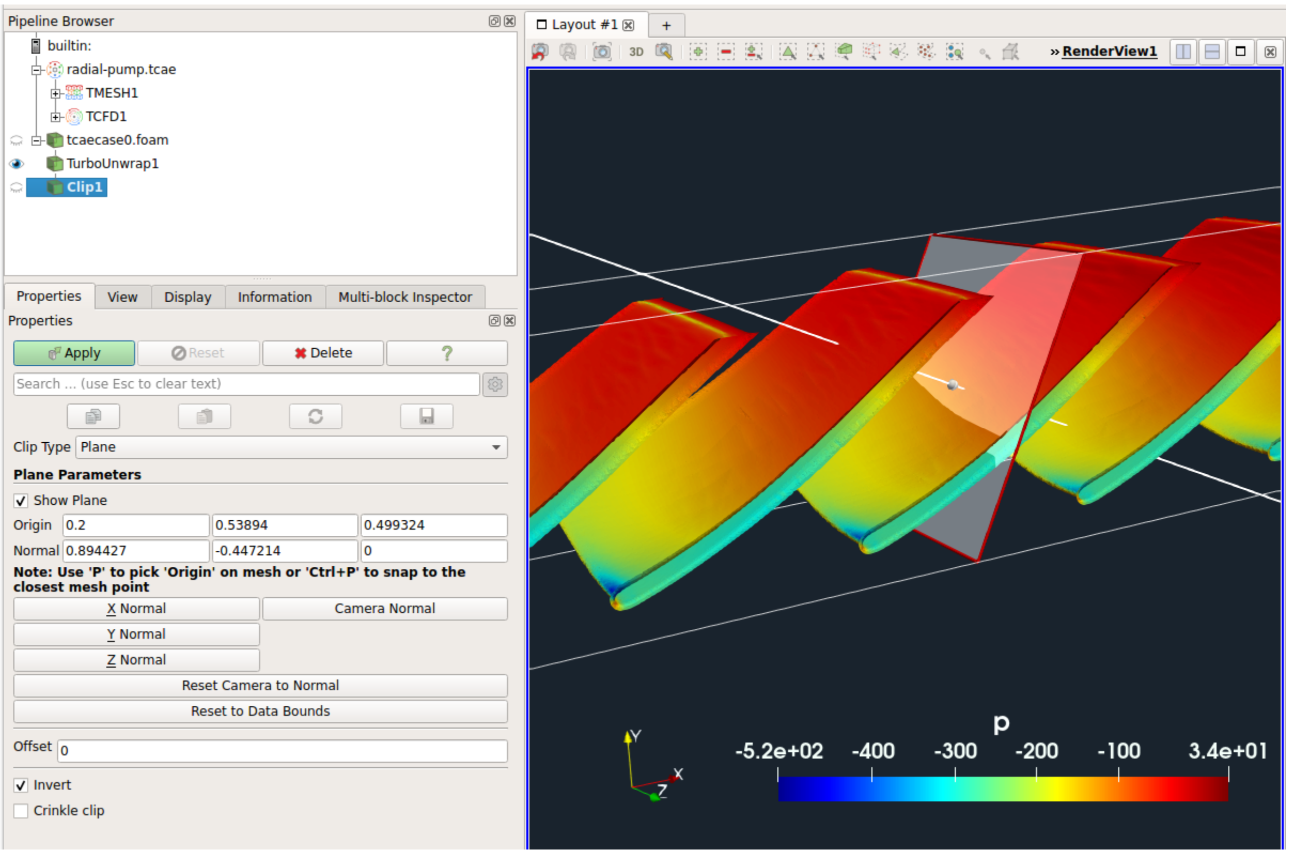
Step 6 — Click on *Apply*. The transformation should be relatively fast, because the sufrace mesh of the blades is orders of magnitude easier to process than the full volume mesh. It may be necessary to zoom in or out a little (depending on the geometry) to make the result fit to window. Outcome of this step is shown in the Fig. 16.23, where the blades are coloured by pressure.

Step 7 — Having the blades transformed we can now separate the chosen blade from the rest. This is done by the filter *Clip*. Adding the *Clip* filter will provide the user with an interactive positioning tool consisting of a ball in the origin, clipping plane and its normal. Use the ball to position the plane. Use the axis to orient the plane; see Fig. 16.25. Once you have placed the clipping plane to one side of the chosen blade so that it doesn't intersect any blade, click *Apply*.

Step 8 — Repeat Step 7 appending another *Clip* filter to isolate the chosen blade also from the other side. You may need to check the *Invert* option in the settings of the filter *Clip*.

Step 9 — Append the filter *Contour*. In the drop-down list *Contour By* select the pressure field. Use the red cross button to erase suggested contour values (right bottom of the box *Value Range*) and then press grid button (right top) to populate the list of contour values by equidistant values covering the whole pressure range. Do not modify the suggested minimal (*From*) and maximal (*To*) values, and use e.g. 20 samples (*Steps*). See the Fig. 16.26.

Step 10 — Pressing *Apply* will calculate the contour data and display the contours, hiding the blade geometry. Click on the eye symbol in *Pipeline Browser* to show the blade again. The results are in the Fig. 16.27.



Clip Type: Plane

Plane Parameters

☑ Show Plane

Origin	0.2	0.53894	0.499324
Normal	0.894427	-0.447214	0

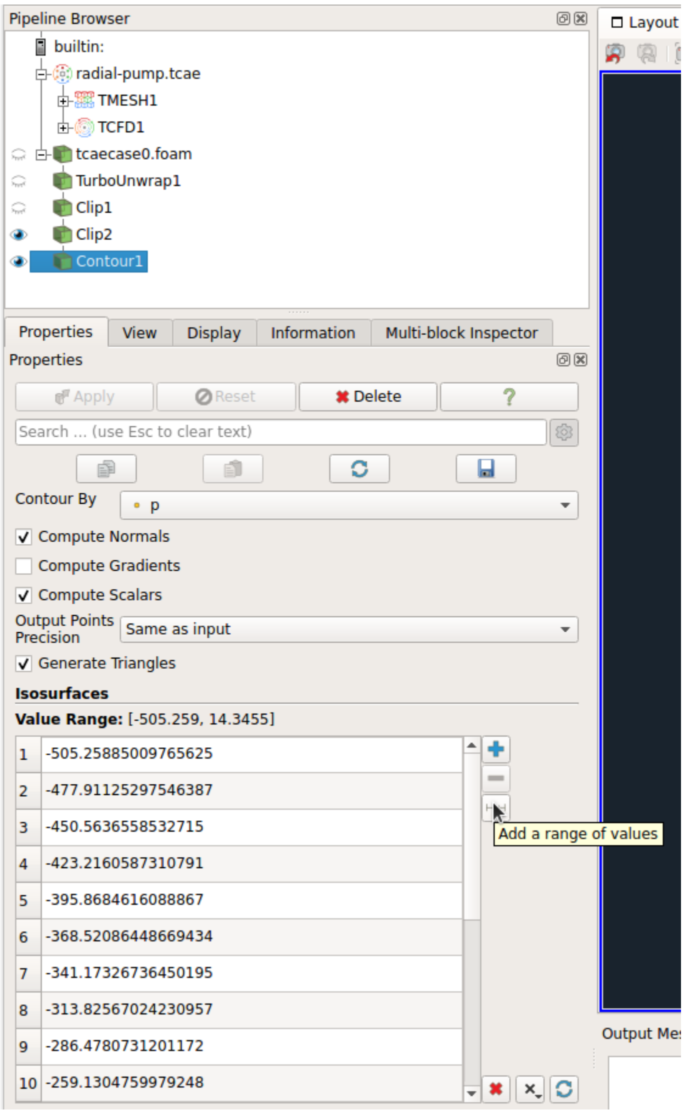
Note: Use 'P' to pick 'Origin' on mesh or 'Ctrl+P' to snap to the closest mesh point

Offset: 0

☑ Invert

□ Crinkle clip

Figure 16.25: Turbo Blade Post – Manipulating with the interactive *Clip* filter when separating a single blade of a pump transformed by Turbo Unwrap filter. The blades are colored by static pressure calculated by TCFD.



Pipeline Browser

builtin:
- radial-pump.tcae
  - TMESH1
  - TCFD1
- tcaecase0.foam
  - TurboUnwrap1
  - Clip1
  - Clip2
  - Contour1

Properties | View | Display | Information | Multi-block Inspector

Properties

Apply | Reset | Delete | ?

Search ... (use Esc to clear text)

Contour By: p

☑ Compute Normals

☐ Compute Gradients

☑ Compute Scalars

Output Points Precision: Same as input

☑ Generate Triangles

Isosurfaces

Value Range: [-505.259, 14.3455]

1	-505.25885009765625
2	-477.91125297546387
3	-450.5636558532715
4	-423.2160587310791
5	-395.8684616088867
6	-368.52086448669434
7	-341.17326736450195
8	-313.82567024230957
9	-286.4780731201172
10	-259.1304759979248

Add a range of values

Figure 16.26: Settings of the filter *Contour* applied on a isolated pump blade transformed by Turbo Unwrap filter.

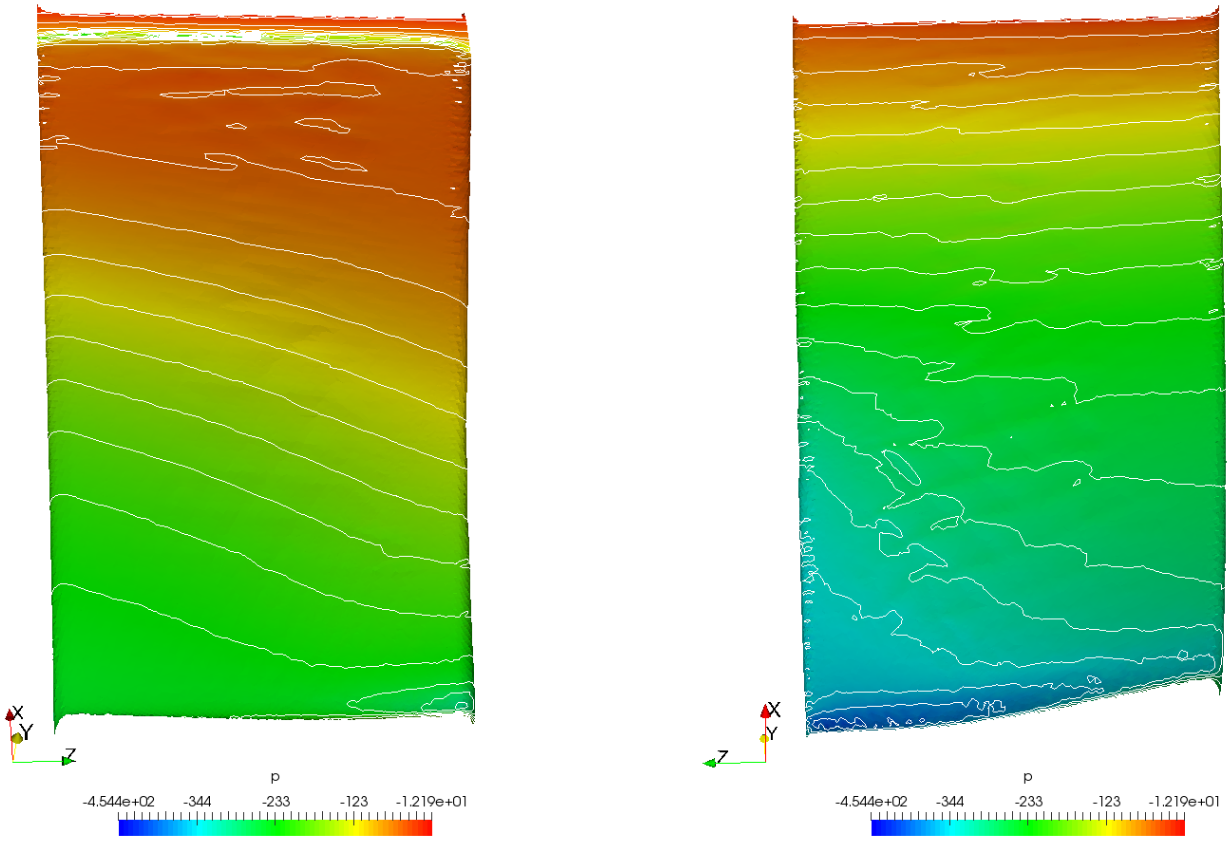


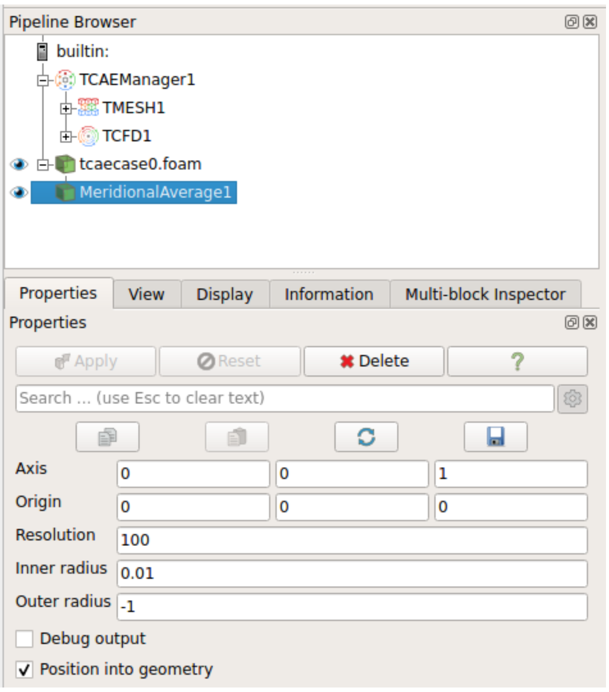
Figure 16.27: Static pressure field and contours on an isolated blade transformed by Turbo Blade Post / Turbo Unwrap filter. The pressure data have been calculated by TCFD. Left is the pressure side, right the suction side of the blade.

16.6 Meridional Average filter – details

The filter Meridional Average creates a geometrical slice by a plane containing the rotation axis and circumferential averages all the field data onto this slice. The slice ignores blades; there are never holes in the slice.

16.6.1 Input parameters

Input parameters of the Meridional Average in the *Properties Panel* are shown in the Fig. 16.28.



Axis 0 0 1

Origin 0 0 0

Resolution 100

Inner radius 0.01

Outer radius -1

□ Debug output

☑ Position into geometry

Figure 16.28: Turbo Blade Post – Meridional Average properties.

Axis and its origin

This input specifies the position and direction of the rotation axis around which the rotor region is placed.

Inner radius

For best results, the rotor region should be donut-shaped, i.e., there should be a hole running through its centre. Only for such shapes the slice contour is well defined. If there is no hole in the middle, this option allows specifying a positive radius of a hole to drill before applying the filter.

Outer radius

This option allows specifying a positive radius to clip out some unwanted parts at the periphery.

Resolution

This parameter controls the number of sampling faces in the slice. In the slice there will be approximately N faces in the axial direction. The faces are squares and their number in the radial direction is calculated automatically.

Debug output

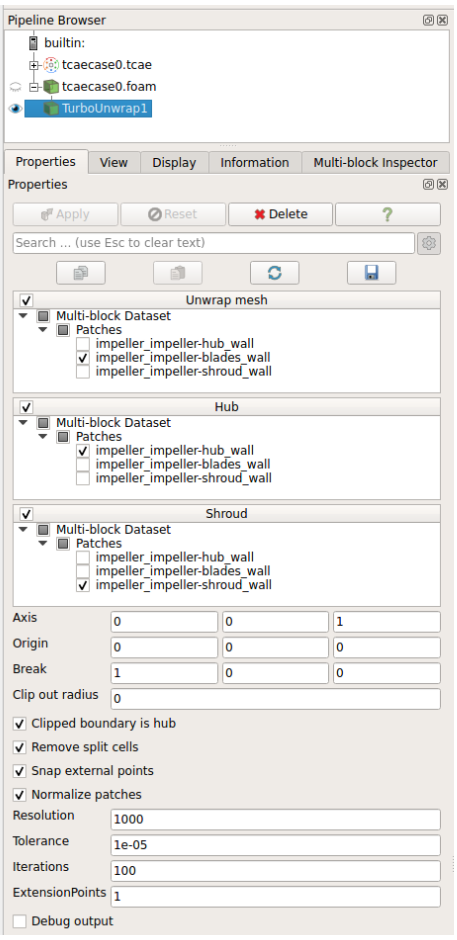
When "Debug output" is checked, then the filter produces some additional output in the form of **.txt* files in the working directory. These files can be used to explore the internal mechanisms particularly when contructing the projected boundary contour.

16.6.2 Averaging

At the moment only the following quantities are sampled: the pressure p, the absolute velocity U (no circumferential component), the relative velocity *URel* (no circumferential component), the absolute velocity magnitude *magU* and the relative velocity magnitude *magURel*.

16.7 Turbo Unwrap filter – details

The filter Turbo Unwrap transforms the rotor region into a box according to a rather complicated set of rules. The purpose is to allow slicing the computational mesh and blades in the fixed relative distance between the hub and shroud patches. The filter can be applied to the whole computational mesh or to its individual parts (clips of e.g. only the blades). The unwrapping transformation is guided by hub and shroud patches. The Turbo Unwrap settings are displayed in Fig. 16.29.


Pipeline Browser
builtin:
tcaecase0.tcae
tcaecase0.foam
TurboUnwrap1

Properties

Search ... (use Esc to clear text)

☑ Unwrap mesh
Multi-block Dataset
Patches
□ impeller_impeller-hub_wall
☑ impeller_impeller-blades_wall
□ impeller_impeller-shroud_wall

☑ Hub
Multi-block Dataset
Patches
☑ impeller_impeller-hub_wall
□ impeller_impeller-blades_wall
□ impeller_impeller-shroud_wall

☑ Shroud
Multi-block Dataset
Patches
□ impeller_impeller-hub_wall
□ impeller_impeller-blades_wall
☑ impeller_impeller-shroud_wall

Axis	0	0	1
Origin	0	0	0
Break	1	0	0
Clip out radius	0		

☑ Clipped boundary is hub
☑ Remove split cells
☑ Snap external points
☑ Normalize patches

Resolution	1000
Tolerance	1e-05
Iterations	100
ExtensionPoints	1

□ Debug output

Figure 16.29: Turbo Blade Post – Advanced Turbo Unwrap filter properties.

16.7.1 Input parameters

Unwrap mesh

The input mesh is one or more mesh parts that are to be "unwrapped" by the filter. There are two possibilities, how to apply Turbo Unwrap to the results - either unwrap volumetric internal mesh of some component, or unwrap just the blade patch(es) (see section 16.7.2).

Hub / Shroud patches

Knowledge of hub and shroud profile is crucial for the transformation algorithm, because the aim of the transformation is to flatten both these patches. This selection box enables user to select both patches.

Axis and its origin

This input specifies the position and direction of the rotation axis around which the rotor region is placed.

Break

To unwrap the mesh, it is necessary to break it somewhere. This option allows specification of a direction, where the cut will occur.

Clip out radius

For best results, the rotor region should be donut-shaped, i.e., there should be a hole running through its centre. Only for such shapes the unwrapping into a box is well defined. If there is no hole in the middle, this option allows specifying a positive radius of a hole to drill before unwrapping.

Clipped boundary is hub

The boundary of the mesh that is newly created by the drilling (specifying non-zero "Clip out radius") can be considered to belong either to the hub or to the inlet/outlet interface. The default is to assign the new boundary to the hub patch. If unselected, the new boundary will be assumed to belong to the inlet/outlet interface.

Advanced parameters

The remaining parameters are in the advanced section and can be shown by clicking on the *Toggle advanced properties* button.

The number parameters "Resolution", "Tolerance" and "Iterations" are internal control parameters that influence precision and speed of the filter. "Resolution" is a approximately the number of control points along the hub and shroud patches. Raising the number of control points can increase the resolution of the approximated patches. "Tolerance" is proportional to the allowed absolute error in determination of point's transformed coordinates (m, t, ζ). The number should generally be smaller than the typical distance between two points in the original mesh. "Iterations" is the iteration limit set for determination of the transformed coordinates. It should not be necessary to raise the default value, unless an extremely fine mesh is being transformed.

The field "Extension Points" is only used when "Snap internal points" is unchecked. It determines which sampled points of the boundary patches are used to extrapolate the patches. The number must be and integer greater than zero. If "1" is given, then only the tail of each patch is used to extrapolate their direction. If larger number is given, then the extrapolation runs in a more averaged direction. If "Snap internal points" is checked, no extrapolation is done and points outside the area bounded by hub and shroud are simply left with some extremal m-coordinate.

The check box "Remove split cells" (default: on) determines whether the cells that are split by unwrapping are to be removed or kept. Removing these cells allows ParaView to display the unwrapped mesh correctly; otherwise it looks as if there was no internal mesh. However, for special meshes this leads to a crash. In such cases, unchecking is an option.

The option "Normalize patches" makes sure that the transformed hub and shroud patches will be of unit size in the ξ (i.e. z) direction. This allows easy construction of the blade-to-blade view , but it may not be desired for the overall view. If this option is unchecked, the hub will still be normalized to unit size, but the transformed shroud will keep its ratio with respect to the hub.

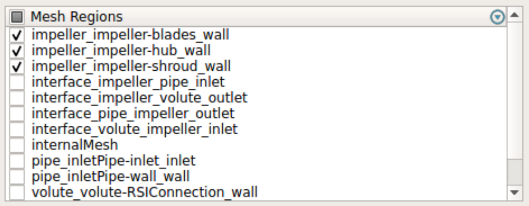
Finally, the "Debug output" option produces additional text information as a **.txt* files in the working directory, which can be used to explore internal mechanisms of the filter, particularly the construction of the m-ξ (y-z) contour which is then used to transform the whole mesh.

16.7.2 Usage

First of all, the results of the calculation has to be properly loaded, which is described in the Sec. 4.4.5. There are two possibilities, how to apply Turbo Unwrap.

If one wants to unwrap the volumetric mesh of some component (stator or rotor) (as demonstrated in 16.3), he/she has to select *internalMesh*, along with the hub and shroud patches in the *Mesh Regions*, and check on *Read zones* and *Copy data to cell zones* in the process of loading results.

On the other hand, if user wants to unwrap the "2D" results, e. g. for the purpose of displaying the pressure on the blade surfaces (Sec. 16.4, it is needed to select just the blade and hub and shroud patches in the *Mesh Regions* (shown in Fig. 16.30).



■ Mesh Regions
☑ impeller_impeller-blades_wall
☑ impeller_impeller-hub_wall
☑ impeller_impeller-shroud_wall
□ interface_impeller_pipe_inlet
□ interface_impeller_volute_outlet
□ interface_pipe_impeller_outlet
□ interface_volute_impeller_inlet
□ internalMesh
□ pipe_inletPipe-inlet_inlet
□ pipe_inletPipe-wall_wall
□ volute_volute-RSIConnection_wall

Figure 16.30: Turbo Blade Post – Selection of multiple parts of the mesh.

The filter will transform the cylinder-like mesh into a box. The meaning of its new dimensions is described in the table 16.1. In brief it can be said that the boundary $z' \equiv \zeta = 0$ is the hub patch, the boundary $z' \equiv \zeta = 1$ is the shroud patch, the boundary $x' \equiv m = 0$ is the one of inlet and outlet interfaces that is lower on the rotation axis and the boundary $x' \equiv m = 1$ is the other one (higher on rotation axis). The boundaries on minimal and maximal $y' \equiv t$ coordinates are only artificial and were originally connected to each other.

The filter transforms only the cell data, not point data, so it may be necessary to apply the filter Cell data to point data afterwards to regain access to the point fields (which are necessary for usage of e.g. the Glyph filter).

Symbol	Mapped to	Range	Meaning
m	x	$(0, 1)$	Distance along the hub/shroud/streamline.
t	y	$(0, 2\pi)$	Circumferential angle.
ζ	z	$(0, 1)$	Distance from the hub; the "span".

Table 16.1: Coordinates of the "unwrapped" mesh. The coordinate m is relative to the full length of the hub/shroud/streamline. The coordinate ζ is relative to the full local distance between hub and shroud.

The filter passes all cell data without change except for the vector fields U and URel, which are transformed into the new coordinate system $(x', y', z') \equiv (m, t, \zeta)$. Besides these two vector fields Turbo Unwrap also creates several additional vector fields, namely UStream and URelStream which should be used to construct streamlines in the transformed mesh and URelLIC, which should be used as an input for the Surface Line Integral Convolution (*SurfaceLIC*) integrator when displaying realtive velocity using *SurfaceLIC* representation.

Part IV

TFEA

Chapter 17

TFEA – Introduction

What is TFEA?

TFEA is an engineering software for Finite Element Analysis on solid geometries. TFEA was designed to calculate the Deformation, Stress, and Modal analysis. It can be used standalone for calculating deformations caused by rotational movement of the solid or together with TCFD. In the later case, the forces from results of CFD analysis are used as boundary conditions for solid.

In the case of coupling with TCFD TFEA is computed for every speedpoint, the results are then store as a time series. The time steps of TFEA correspond to time steps of TCFD and can be visualised together.

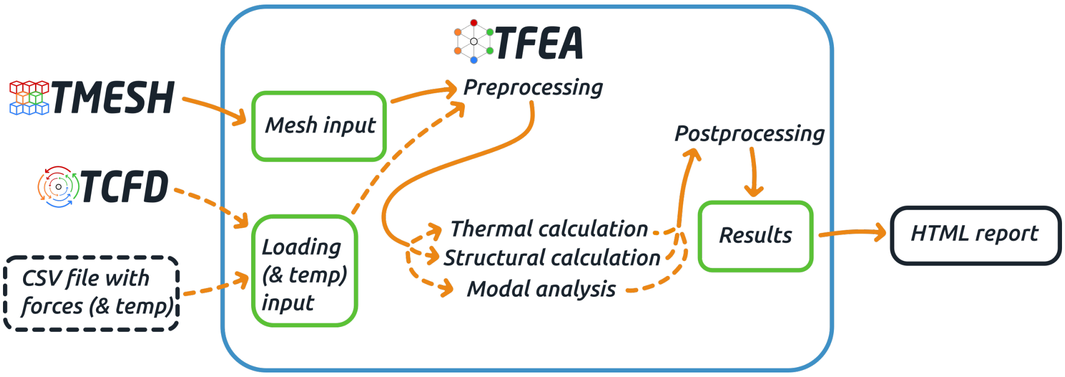


Figure 17.1: TFEA – data workflow

17.1 Governing Equations for Solid

In this section we provide a short discussion on basics of solid mechanics. More information can be found on the internet or in some book, e.g. [23]. In the absence of volume forces (such as gravity or magnetical force field) the governing equations (not taking into account the boundary conditions) for solid read

$$\rho_s \partial_{tt} \mathbf{u}_s = \operatorname{div} \mathbb{T}_s^{(1)}, \tag{17.1}$$

where ρ_s is the solid density, $\mathbf{u}_s$ is the displacement of the solid and $\mathbb{T}_s^{(1)}$ is the 1st Piola-Kirchhoff stress tensor.

For the stress tensor a constitutive relation needs to be chosen. For solid mechanics we benefit from open-source software CalculiX where Saint Vennant-Kirchhoff model is used. For this constitutive relation the 1st Piola-Kirchhoff stress tensor is given by

$$\mathbb{T}_s^{(1)} = \mathbb{F}\left[\lambda_s \mathrm{Tr}(\mathbb{E})\mathbb{I} + 2\mu_s \mathbb{E}\right], \tag{17.2a}$$

$$\mathbb{E} = \frac{1}{2}\left(\mathbb{F}^T\mathbb{F} - \mathbb{I}\right), \tag{17.2b}$$

$$\mathbb{F} = \mathbb{I} + \nabla \mathbf{u}_s, \tag{17.2c}$$

where λ_s and μ_s are *Lamé constants*. Let us note that the transformation between the Cauchy stress tensor $\mathbb{T}$ and 1st Piola-Kirchhoff stress tensor $\mathbb{T}^{(1)}$ is

$$\mathbb{T}^{(1)} = (\det \mathbb{F})\mathbb{T}\mathbb{F}^{-T}.$$

An alternative pair of constants is the *Young modulus* E_s which describes the response of the material to the stress ("the measure of stiffness") and the *Poisson ratio* ν_s which describes the response to deformation of a material in a perpendicular direction ($\nu_s = -\frac{\text{transverse strain}}{\text{axial strain}}$), $\nu_s = 0.5$ for an incompressible material. E_s and ν_s are easier to measure than Lamé constants λ_s and μ_s, but the later ones are more convenient from the point of view of the constitutive equations. Luckily, there is a simple relation between these two sets, it reads

$$\nu_s = \frac{\lambda_s}{2(\lambda_s + \mu_s)} \qquad E_s = \frac{\mu_s(3\lambda_s + 2\mu_s)}{(\lambda_s + \mu_s)}$$

$$\mu_s = \frac{E_s}{2(1 + \nu_s)} \qquad \lambda_s = \frac{\nu_s E_s}{(1 + \nu_s)(1 - 2\nu_s)}.$$

The system of equations (17.1) and (17.2) is apparently non-linear. For small displacement, this can be linearized to

$$\mathbb{T}_s^{(1)} = \lambda_s \mathrm{div}\, \mathbf{u}_s \mathbb{I} + 2\mu_s \left(\nabla \mathbf{u}_s + \nabla \mathbf{u}_s^T\right). \tag{17.3}$$

The system of equations (17.1), (17.3) is what we undestand under the term *linearized equations for solid.*

Let us now say a few words on how the systems of equations are solved within CalculiX. For the linearized case the process is clear, the matrix-vector system is solved only once and that's all The non-linear equations are solved as follows. System of linear equations is determined and solved, the solution and the system is updated. These steps are iterated until convergence.

Clearly, the non-linear case is much more expensive, moreover, for small deformations the difference in results is very often negligible. When the small deformation is expected (which we assume is usually the case in turbomachinery applications), we therefore recomend to use the linearized syste.

17.1.1 Boundary Conditions

To close the systems of equation (17.1), (17.2) and (17.1), (17.3). We need to provide boundary conditions.

We surely want to fix to solid on the part of the boundary (prescribe the homogeneous Dirichlet boundary condition for the solid displacement $\mathbf{u}_s$). Otherwise, as a consequence of the exertion of forces, the solid would have a tendency to move far far away or do something even more strange. In TFEA this is done by specifying so called "Fixed regions".

We are further able to prescribe some quantities on a part of boundary. Currently, this can be done for temperature.

Fluid-Structure Intraction (FSI) - exerting fluid forces computed with TCFD on the boundary corresponding to the interface between fluid and solid is a boundary condition for stress. We require balance of forces (normal stresses) over the interface (in case of correctly set simulation)

$$\mathbb{T}_f \mathbf{n}_f = -\mathbb{T}_s \mathbf{n}_s,$$

where the quantities with subscript f correspond to fluid quantities and the quantities with subscript s correspond to solid quantities.

17.2 Static Analysis

Typically, the results of FEA analysis is displacement vector field and stress tensor field. For stress we also provide *von Mises stress* and *principal stresses*. The extremal values are extracted in report.

Principal stresses are the extreme values of normal stresses possible in the material. More precisely, at every point in a stressed body there are at least three planes, called principal planes, with normal vectors $\mathbf{n}$, called principal directions, where the corresponding stress vector is perpendicular to the plane, i.e., parallel to the normal vector $\mathbf{n}$, and where there are no normal shear stresses. The magnitudes of the three stresses normal to these principal planes are called principal stresses. In other words, they are the largest and smallest eigenvalues of Cauchy stress tensor at a given point, respectively. The maximal principal stress is also called *major principle stress* and the minimal principal stress is also called *minor principal stress*. Another informatino can be found e.g. in Wikipedia [20].

Von Mises stress σ_v can be used for predicting yielding, as, according to theory, material begins yield after σ_v crosses certain critical value. It is defined as

$$\sigma_v = \sqrt{\frac{1}{2}[(\sigma_{xx} - \sigma_{yy})^2 + (\sigma_{yy} - \sigma_{zz})^2 + (\sigma_{zz} - \sigma_{xx})^2 + 6(\sigma_{xz}^2 + \sigma_{xz}^2 + \sigma_{yz}^2)]},$$

where $\sigma_{..}$ are components of Cauchy stress tensor. The determining property for yielding is only applicable for isotropic and ductile material. For more detailed explanation see, e.g. Wikipedia [21].

17.3 Frequency Analysis

The frequency analysis can be performed both, on unloaded structure or on loaded structure, when the loading is known from previous steps. This is controled by the "Compute for loaded solid" checkobox. Thus, the effect of the centrifugal force on the eigeinfrequencies in a turbine blade can be analyzed by first performing a static calculation with these loads, and selecting the preload in the frequency step.

The results of frequency analysis is number of eigenfrequency-eigenmode pairs. This number is set by user under "Number of eigenfrequencies" entry, the ordering is from lowest eigenfrequency.

Eigenfrequency is the frequency in which the solid has a tendency to oscillate and *eigenmode* describes the shape of the oscillation (as displacement and corresponding stress). The frequency is a complex number, where the imaginary part describes buckling. This can be caused by the preload, when computing the unloaded frequency analysis, the eigenfrequencies should be real.

We also report *eigenvalues*, these are the eigenvalues of stiffness matrix. Solving this eigenvalue problem is the way how the eigenfrequencies and eigenmodes are computed. The eigenvalue is the second power of the corresponding eigenfrequency.

Moreover, we report to user *Participation Factor*, *Effective Modal Mass* and *Total Effective Modal Mass*. These quantites are described on the the following lines.

Participation Factor characterizes how much a certain mode will be excited by a rigid body acceleration in a certain direction. We list these values in directions of coordinate axis and also in the directions of rotations around the axis.

Effective Modal Mass is defined as participation factor squared (component-wise). For an acceleration in a specific direction, it shows how much of the total inertial force can be attributed to this mode. This value can be used to estimate how many modes are needed for a good representation in a subsequent response analysis based on mode superposition. The last row in the corresponding table in report, named *total*, is the sum of the effective modal masses corresponding to all computed eigenvalues. This means that this row is a component-wise sum of all previous rows.

Total Effective Modal Mass is the size of rigid motion. If one would calculate infinitely many modes the total effective modal mass should be equal to the total effective mass. Since only a finite number of modes are calculated the total effective modal mass will be smaller. By comparing the total effective modal mass with the total effective mass one gains an impression whether enough modes were calculated to perform good modal dynamics calculation (at least for the rigid motions).

For more details we redirect you to CalculiX manual [22] or to some solid mechanics book.

17.3.1 Frequency Analysis for Periodic Segment

For periodic structure TFEA module is able to perform Finite Element Analysis given only a periodic segment. The frequency analysis in this case requires special treatment, the approach used here we simply take from Calculix, [22]. The frequency analysis is performed for each possible *Nodal Diameter*. Nodal diameter (also known as *Cyclic Symmetry Mode Number*), simply put, is the number of diameters in the structur along which the displacement (of the eigenmodes) is zero. TCAE computes eigenmodes for all possible nodal diameters which is $\frac{N}{2}$ for N even or $\frac{N+1}{2}$ for N odd, N being the number of periodic segments. We thus have $\frac{N}{2}$ (or $\frac{N+1}{2}$ modal analysis simulations when working with periodic segments.

The periodicity is forced by complex constraints tying together left and right periodic boundary regions in a phase shift corresponding to used nodal diameter. The eigenmodes are generated in pairs phase-shifted by $\frac{\pi}{2}$. We save only the first one with its complex and imaginary part.

17.4 Fluid-Structure Interaction

Let us now discuss the Fluid-Structure Interaction phenomenon as it is understood within the TCAE framework. We assume this will be the main usage of TFEA, more precisely, we assume that TFEA will be usually used together with TCFD. Meaning, that we extract the fluid forces exarted on the Fluid-Solid interface and use them as a boundary condition for solid.

Note that we take into account only one-sided coupling, we neglect the effects of geometry changes on the fluid flow. This neglection is reasonable only if the solid deformation is "small".

Typically, the user has one TCAE case within which CFD and subsequently FEA simulation is performed. In TCFD setup, the user chooses number of speedlines and selects between stationary and (stationary +) transient case. The stationary simulation also asks for speedpoints within each speedline. Based on this setting the TFEA analysis is performed. The results of each speedpoint are used as a boundary condition for single FEA calculation, in transient case we take the averaged quantities over the last transient window of each speeline and use them for another FEA calculations. For each stationary speedline the results are saved as a time series with speedpoint being a time instant (so it can be visualize together with TCFD results). For transient calculation TFEA creates a single file per speedline (it corresponds to the averaged stress over the transient window) In the report each speedpoint is reported in its own section, in transient case new report is created that summarizes the results from transient calculation.

The other way to perform Fluid-Structure interaction is to have the forces stored in a directory with CSV files. The structure of the CSV files is assumed to be x, y, z coordinates of the point where the forces are evaluated and then three components of the force field, in case of temperature mapping, follows a column with temperature value. The CSV files obtained with TCAE function "calcSurfaceQuantities" has one last column with the area of the corresponding CFD face. The values on a row are separated by a comma.

17.5 TFEA directory structure

TFEA directory is subdirectory of TCAE case (see 3.2).

The whole directory should be read-only, and its basic structure is shown in the table 17.1, because user might want to extract some useful data for postprocessing or to look at the logs.

🗀 project_folder	Project directory
└ 🗀 simulationRun	
└ 🗀 TFEA	all files, that belong to TFEA module
└ 🗀 mesh	directory with mesh related files
└ 🗋 CCXSettings.inp	CalculiX confirguration file
└ 🗋 TFEAResults.pvd	file for restuls visualization in ParaView
└ 🗀 RESULTS	directory with results for a single time instant
└ 🗀 report	HTML report with its images
└ 🗀 transient	directory containing subdirectories with data of all of the transient points (in FSI transient case)

Table 17.1: TFEA – Directory structure

Chapter 18

TFEA – GUI Setup & Options

Module TFEA in the GUI can be added / removed by using the modules buttons (4.1) and is selected by the clicking on the *TFEA* item in the *Pipeline Browser*, or on one of its *Output Ports*, which are:

- Settings
  Using this port the keywords and their values of TFEA module are displayed as a table in *SpreadSheet View*.

- Report
  If the FEA calculation if finished and the HTML report is done, it will be displayed in *HTML View* through this port.

After selection the TFEA, user can see what is depicted in the Fig. 18.1. In the *Properties Panel*, there are menus, their contents are thoroughly described in following sections: "SIMULATION", "MATERIAL PROPERTIES", "BOUNDARY CONDITIONS" and "POST-PROCESSING".

18.1 SIMULATION

Under this section the general FEA settings as well as material properties and type of solver are to be set (Fig.18.2).

18.1.1 FEA Analysis

The layout of the FEA Analysis card is shown in Figure 18.3. In the beginning the four main checkbox are to be specified, their combination then determines the structure of the rest of this card. We believe the names nicely describes what they do

- If the checkbox "Heat transfer only" is selected, no displacement or stresses are computed. Only temperature distribution. This means that thermal expansion is neglected in thi type of simulation. The results is temperature field.

- "Add thermal effects" means that the thermal effects, like thermal expansion, are taken into account as well as the loading forces. In this type of analysis displacement, stress

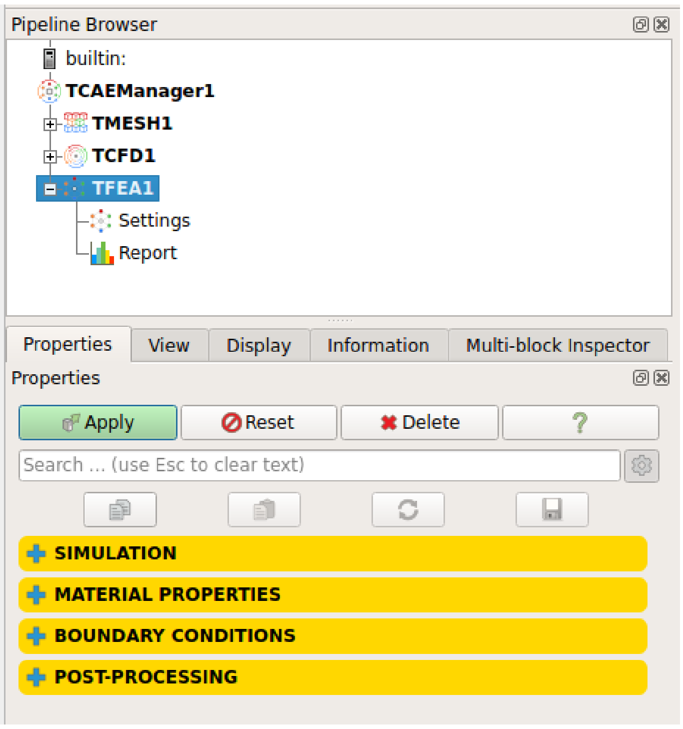


Figure 18.1: TFEA – *Pipeline Browser* and *Properties Panel*

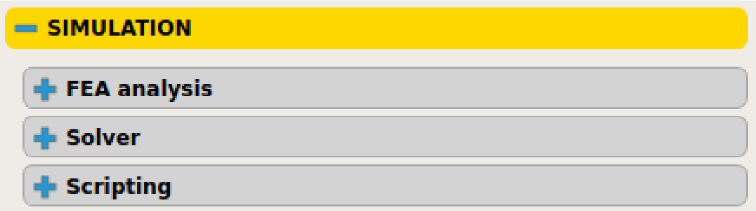
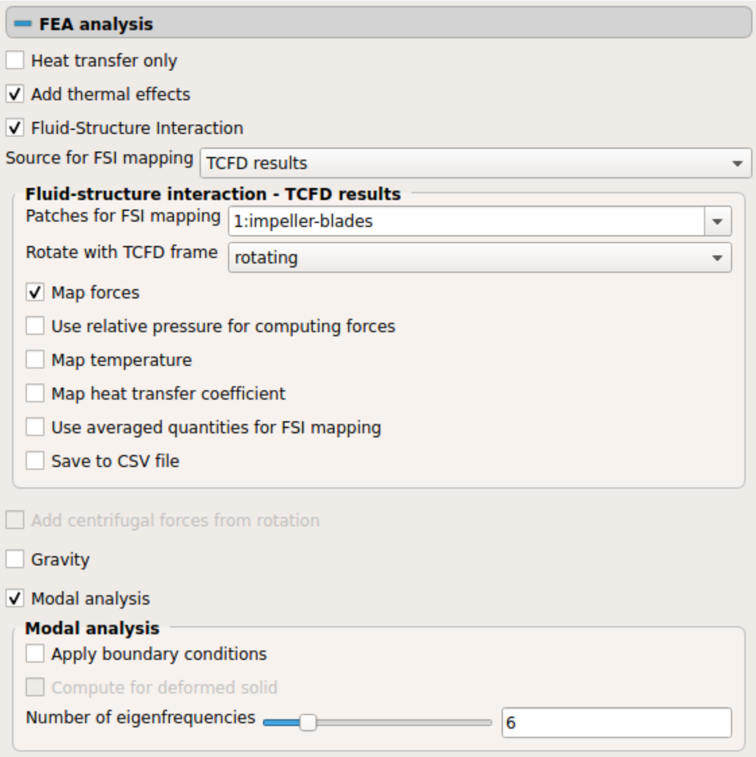


Figure 18.2: TFEA – Simulation.

and temperature fields are computed as a result. If not selected (and "Heat transfer only" is uncheked too), only stress and displacement fields are evaluated.

- "Fulid-Structure Interaction" selects whether the Fluid-Structure Interacrtion (FSI) is performed (more specifically, whether we map forces obatined from CFD analysis are used as a boundary condition).
- "Add centrifugal forces from rotation" selects whether the solid rotates, this option is selectable only if the checkbox "Fulid-Structure Interaction" is unchecked or "Source for FSI mapping" is set to "External directory with CSV files". In the other case the rotation is linked with TCFD settings).
- "Gravity" selects whether the gravity effects are considered.
- "Modal analysis" selects whether the eigenmodes for the solids will be computed.



FEA analysis

□ Heat transfer only

☑ Add thermal effects

☑ Fluid-Structure Interaction

Source for FSI mapping: TCFD results

Fluid-structure interaction - TCFD results

Patches for FSI mapping: 1:impeller-blades

Rotate with TCFD frame: rotating

☑ Map forces

□ Use relative pressure for computing forces

□ Map temperature

□ Map heat transfer coefficient

□ Use averaged quantities for FSI mapping

□ Save to CSV file

□ Add centrifugal forces from rotation

□ Gravity

☑ Modal analysis

Modal analysis

□ Apply boundary conditions

□ Compute for deformed solid

Number of eigenfrequencies: 6

Figure 18.3: TFEA – FEA analysis.

Fulid-structure interaction

The section "Fulid-structure interaction" appears only if the corresponding checkbox is selected.

- "Source for FSI mapping" sets from where the forces exerted by fluid on the solids should be taken.

  - "TCFD results" Selects the results of the TCFD case belonging to this TCAE case. This means that this option is valid only if both, TCFD and TFEA are performed within this TCAE case. In case you already have computed TCFD the results are read without the need of recalculation (i.e., only the TFEA is calculated).
  - "CSV files" Selects that the fluid forces are loaded from CSV files.

- "Patches for FSI mapping" selection box is present only if "Source for FSI mapping" is set to "TCFD results". Here you choose the boundary pathes of TCFD that are used for the force transmission to TFEA. (In order to have reasonable results, this should be the patches that correspond to the fluid-solid interface.)

- "Rotate with TCFD frame" selection box is present only if "Source for FSI mapping" is set to "TCFD results". This will create a link between the selected TCFD frame and the solid. The rotation values of TCFD frame are prescribed for the solid too and the solid rotates with the same angular velocity around the same axis as this frame.

- "CSV directory" entry is present only if "Source for FSI mapping" is set to "External directory with CSV files" (not visible in Figure 18.3). This specifies the path to the directory containing CSV files.

- "Files to map" entry is present only if "Source for FSI mapping" is set to "External directory with CSV files" (not visible in Figure 18.3). Here you select CSV files present in the directory that are used as a source of FSI forces.

- "Map forces" checkbox controls whether we are interested in mapping the forces from TCFD to TFEA.

- "Use relative pressure for computing forces". If his checkbox is selected relative pressure instead of total pressure is used. This option is selectable only in case of having "TCFD results" as "Source for FSI mapping".

- "Map temperature" checkbox is present only if one of the checkboxes "Add thermal effects" or "Heat transfer only" is selected. Here you control whether we are interested in mapping the temperature from TCFD or external file to TFEA.

- "Map heat transfer coefficient" controls mapping of heat transfer coefficient and ambient temperature. This option is available only if one of the checkboxes "Add thermal effects" or "Heat transfer only" is selected.

- "Use averaged quantities for FSI mapping". If his checkbox is selected the averaged values of quantities used for FSI mapping are used. The averaging window is the one given in TCFD. This option is selectable only in case of having "TCFD results" as "Source for FSI mapping".

- "Save to CSV file". If selected, the mapped quantities are saved into an CSV file. This file is located on the path `<project>/simulationRunX/TFEA/RESULTS/mappedQuantities.csv`.

Rotation

There is another checkbox "Add centrifugal forces from rotation", by means of which the rotation of the solid is prescribed. Note, that this option is visible, only if

- Fluid-Structure Interaction is disabled or
- Fluid-Structure Interaction is enabled, and Source for FSI mapping is set to External directory with CSV files.

On the other hand, if user enables Fluid-Structury Interaction and switches Source for FSI mapping to TCFD results, the influence of rotation is implicitly included in the forces, that are mapped from TCFD to TFEA.

Then, if the Add centrifugal forces from rotation is selected, a new submenu with following options appears:

- "Rotation speed" determines the angular velocity with which the solid rotates. The unit of the angular velocity is to be selected in the selection box on the right on the same row.
- "Axis origin" Selects the origin of the axis of rotation.
- "Axis direction" Selects the direction of the axis of rotation.

Gravity

The section "Gravity" appears only if the corresponding checkbox is selected.

- "Gravitational acceleration direction" sets the direction of the gravitational acceleration vector.
- "Gravitational acceleration magnitude" sets the magnitude of the gravitational acceleration vector.

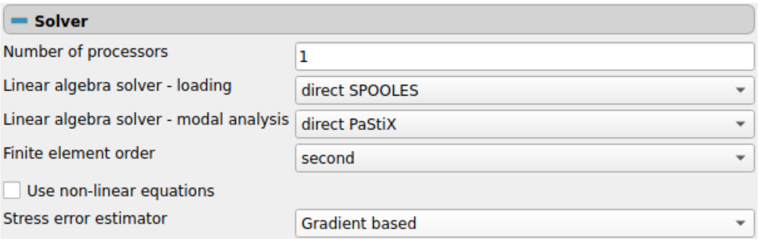
Modal analysis

The section "Modal analysis" appears only if the corresponding checkbox is selected.

- "Apply boundary conditions" let's the user decide if the boundary conditions (the zero displacement conditions that fix the node positions) are applied to the eigenanalysis calculation.
- "Compute for loaded solid" selects between performing modal analysis for undeformed solid (unchecked) or for deformed one (be FSI, rotation or thermal expansion). This checkbox is therefore visible only when at least one of the checkboxes "Fulid-Structure Interaction", "Rotation" or "Add thermal effects" is selected.
- "Number of eigenfrequencies" specifies how many (lowest) eigenfrequencies are computed. Consult section 17.3 of this manual for theoretical details.

18.1.2 Solver

This is the palce where you specify the equations for solids and choose mathematical tools to solve them. In GUI the user will see a section similar to Figure 18.4.



Solver

Number of processors: 1

Linear algebra solver - loading: direct SPOOLES

Linear algebra solver - modal analysis: direct PaStiX

Finite element order: second

□ Use non-linear equations

Stress error estimator: Gradient based

Figure 18.4: TFEA – Solver.

These options are available:

- "Processors" entry specifies how many Open MP threads will be used during the CalculiX run in the FEA calculation.

- "Linear algebra solver - loading" entry provides the user the choice between "direct SPOOLES "direct PaStiX and "iterative" solver for the system of (linear) equations coming from Finite Element discretization of the loading simulation. Note that the iterative solvers consume less memory and can therefore handle larger systems. More specifically, with 32GB of RAM you can solve up to 1 mil. equations using the direct solver. On the other hand the direct solvers tend to be faster for most of the applications. In general, "direct PaStiX" is more efficient (in terms of boith, speed and memory consumption) then "direct SPOOLES" and it is the recommended option.

- "Linear algebra solver - modal analysis" entry provides the user the choice between "direct SPOOLES and "direct PaStiX" solver for the system of (linear) equations coming from Finite Element discretization of the eigenvalue problem. The iterative solver is not available here. Otherwise, the comments from the point above apply here too.

- "Preconditioner - loading" entry is available for "iterative" "Linear algebra solver - loading". This lets the user choose between two preconditioners "diagonal" and "cholesky" (incomplete cholesky). When "Linear algebra solver" is set to "direct" this option is not present. "Cholesky" preconditioner tends to have better convergence than "diagonal" but requires more memory.

- "Finite element order" entry lets the user choose the order of the finite element base functions. The supported options are "first" and "second". The second order elements creates more degrees of freedom (and therefore leads to larger system to solve) and usually provides more acurate results.

- "Use non-linear eqations" option selects the governing equations for solid, whether the full non-linear equations are used (selected) or the linearized equations (unchecked). For theoretical details consult the section 17.1 of this manual.

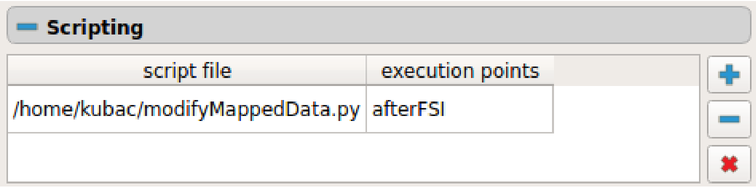
- "Stress error estimate" gives estimate on the error of stress field. The choice is between "Gradient based" and "Zienkiewicz-Zhu". The error estimate field is then possible to visualize in Render View. To read more about the estimates please consult the CalculiX manual [22].
- "Heat flux error estimate" gives possibility to calculate an error estimate of heat flux field. The choice is between "Gradient based" and "None"(no estimate is calculated). The error estimate field is then possible to visualize in Render View. To read more about the estimates please consult the CalculiX manual [22].

18.1.3 Scripting

Similarly as in the TCFD module, TFEA offers usage of external (user-defined) Python scripts (see figure 18.5). They are expected to be written in basic Python 3.7 and the user choose one or more "Execution points". The script is executed in all chosen execution points, these are:

- "afterWrite"
- "beforeCalculation"
- "afterCalculation"
- "beforeFSI"
- "afterFSI"
- "beforeReport"
- "afterReport"

We believe the names of evaluation points are descriptive enough and their meanings are clear.



Scripting

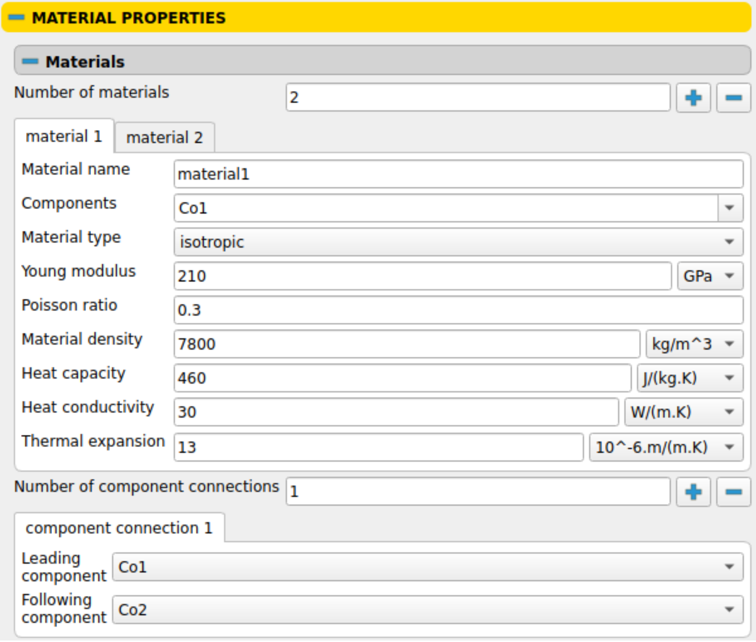
script file	execution points
/home/kubac/modifyMappedData.py	afterFSI

Figure 18.5: TFEA – SIMULATION: Scripting.

18.2 MATERIAL PROPERTIES

The solids in TFEA compose of components. Each component can have different material properties that are to be set in this section. The components can stay independent on each other or can be glued together. Staying independent is meant for the applications where more solids that does not interact with each other are simulated whereas being glued together is meant for applications where one solid is composed of multiple materials. In complex simulations it is even possible to combine both and let's sya have two components glued and third one let independent.

The component connection is also controled inside this section. Take a look on Fig. 18.6 to see how it looks like.



MATERIAL PROPERTIES

Materials

Number of materials: 2

material 1 | material 2

Material name: material1

Components: Co1

Material type: isotropic

Young modulus: 210 GPa

Poisson ratio: 0.3

Material density: 7800 kg/m^3

Heat capacity: 460 J/(kg.K)

Heat conductivity: 30 W/(m.K)

Thermal expansion: 13 10^-6.m/(m.K)

Number of component connections: 1

component connection 1

Leading component: Co1

Following component: Co2

Figure 18.6: TFEA – Material settings.

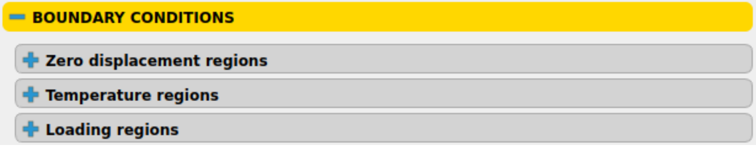
- "Number of materials" specifies the number of materials that will be used in the simulations. The materials are assigned to components.
- "Material name" entry lets the user to name the material.
- "Components" entry holds list of components to which these material properties are prescribed.
- "Material type" entry chooses between "isotropic" and "orthotropic" material.
- "Orientation system" selection box determines the shape of principal directions of orthotropic material. It can be either "cylindrical" or "rectangular".
- "1. point on rotational axes" entry together with "1. point on rotational axes" determines the principal directions of orthotropic material with cylindrical orientation. The principal directions are then axial, radial and tangental.
- "1. point on rotational axes", see "1. point on rotational axes".
- "Point on x axis" entry together with "Point in xy plane" determines the principal directions of orthotropic material with rectangular orientation. The principal directions then coincide with the orientation axes.
- "Point in xy plane", see "Point on x axis".
- "Young modulus" entry controls the value of the Young modulus of the solid material. This option is only for "isotropic" material
- "Poisson ratio" entry controls the value of the Poisson ratio for "isotropic" material.

- "Young moduli" entry controls the value of the (three) Young moduli for "orthotropic" material.
- "Poisson ratios" entry controls the value of the (three) Poisson ratios for "orthotropic" material.
- "Shear moduli" entry controls the value of the (three) Shear moduli for "orthotropic" material.
- "Material density" sets the denstity of the material.
- "Heat capacity" sets the heat capacity of the material. Only visible if one of the checkboxes "Add thermal effects" or "Heat transfer only" is on.
- "Heat conductivity" sets the heat conductivity of the material. Only visible if one of the checkboxes "Add thermal effects" or "Heat transfer only" is on.
- "Thermal expansion" sets the thermal expansion of the material. Only visible if "Add thermal effects" checkbox is on.
- "Number of component connections" controls the number of connections between components. By "Component connection" we mean gluing these components together. In the sense that common interface is found and if the components are not touching on the interface one of the components (so called "following" component) is moved to touch the first ("leading") component. All simulated fields, such as displacement, stress, temperature, etc. are kept continuous over the interface during the simulation. Components that are not connected are simulated completely independently (can even intersect each other).
- "Leading component" specifies the "leading" (see the point above) component of the component connection pair.
- "Following component" specifies the "following" (see the point "Number of component connections") component of the component connection pair.

18.3 BOUNDARY CONDITIONS

Here you specify the boundary conditions (Fig. 18.7. The zero Dirichlet boundary condition for displacement (as "Zero displacement region"), fixed value for temperature or heat flux (as "Temperature region") or constant loading (as "Loading region"). Strictly speaking, these are not a boundary conditions as "Zero displacement region" and "Temperature region"(in case of fixed temperature), can be applied to degrees of freedom inside the solid.

The boundary condition regions and (in the case of FSI) the FSI interface are written to an .vtu file `TFEA/mesh/mesh.vtu`. This is executed in the beginning of the TFEA simulation. Once the simulation is finished, you can see it after activating "Show TFEA results" button. They are stored as a volumetric field for time 0, each condition has its own field with values 0 (not part of the boundary region) and 1 (part of the boundary region).



− BOUNDARY CONDITIONS

✚ Zero displacement regions

✚ Temperature regions

✚ Loading regions

Figure 18.7: TFEA – Boundary conditions.

18.3.1 Zero displacement regions

Number of zero displacement regions (i.e. regions fixed in space) is controlled by the item "Number of zero displacement regions". Based on its value, the appropriate number of panels appear, each of them with these options:

- "Shape of region" Here you specify the geomtrical shape of the region. This can be the whole boundary, Fluid-Structure Interaction interface, a simple geometrical object or more complicated shape described by an "STL file" or "CSV file". The last option is to define the shape as a complement to remaining boundary regions. By simple geometrical object we mean "ball", "box", "cylinder" or "cone". The fix options using a file are implemented as a set of balls of a defined radius around the given points. In the CSV case the points are assumed to be directly given by coordinates (one line corresponds to one point wtih comma separated coordinates). In the STL case the point coordinates are extracted from the file. This means that if the STL contains triangles with long edges some coordinates might not be caught. Please avoid this kind of STLs or use an extra fix option to catch the points in the middle of these faces.

- "Radius" This option applies for shapes "ball", "cylinder" and "cone, it sets the radius of the object.

- "Distance" This option applies for shapes "STL File and "CSV File", it sets the distance to the surface described by STL file or the points stored in the CSV file.

- "Height" This option applies only for shape "cylinder" or "cone" and sets its height. For a negative value, the cylinder is assumed to have infinite length along its axis in both directions, for "cone" option the negative value is illegal.

- "Axis" This option applies only for shape "cylinder" and "cone", it just sets the direction of its axis.

- "Origin" This option applies only for shapes "ball", "cylinder" and "cone", this sets the origin of the shape's axis.

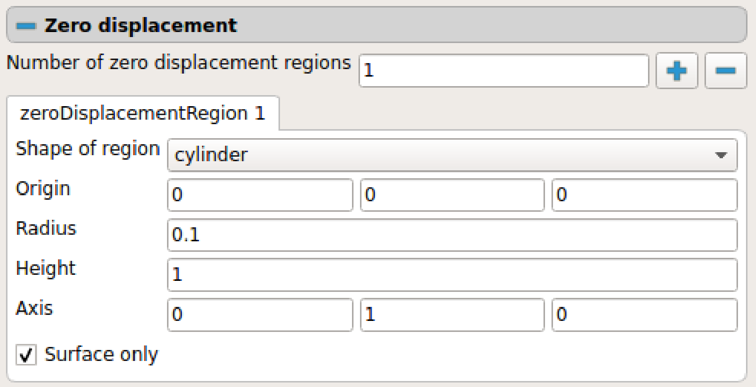
- "Rectangle min. point" This option applies only for shape "rectangle" and sets the lower left corner of the rectangular region.

- "Rectangle max. point" This option applies only for shape "rectangle" and sets the upper right corner of the rectangular region.

- "File name" This option aplies for shapes defined by an "STL File or "CSV File", this only sets the name of the file.

- "Surface only" When this checkbox selected, only degrees of freedom lying on the boundary are effected.
- "Use scale factor from TMESH" option applies to the choice "STL file" in "Shape of region" entry. If this option is on, the geometry is first scaled by the "Scale factor" from TMESH module.
- "Complement to FSI interface" option is available for the choice "Complement" as "Shape of region" and if Fluid-Structure Interaction is being simulated. The boundary region is then the surface without the FSI interface. If moreover "Complement to remaining zero displacement regions" is also selected, the region is further restricted.
- "Complement to remaining zero displacement regions" option is available for the choice "Complement" as "Shape of region". It will subtract from the boundary boundary points already contained in other zero displacement regions.
- "Assign common points with fluid-structure interface to" checkbox appears if FSI analysis is chosen. It controls what to do with points lying on the intersection of FSI interface and this region. You can assign this value to "Both", "FSI" or "This boundary". "Both" means, the points on the intersection will be part of this boundary and, at the same time, the quantities from fluid results will be mapped on this point (i.e., the fluid forces are mapped but point is fixed anyway). "FSI" means the points will be part of FSI inteface but not will be contained in this boundary (i.e., forces are mapped and the points are free to move). "This boundary" means the points are only part of this boundary but not part of the FSI interface (i.e., points are fixed and nothing is mapped to them, the quantities are mapped on closest possible nodes on the FSI interface).

The example of the settings of the fixed regions is shown in the figure 18.8.


Zero displacement

Number of zero displacement regions: 1

zeroDisplacementRegion 1

Shape of region: cylinder

Origin: 0 0 0

Radius: 0.1

Height: 1

Axis: 0 1 0

☑ Surface only

Figure 18.8: TFEA – Zero displacement regions

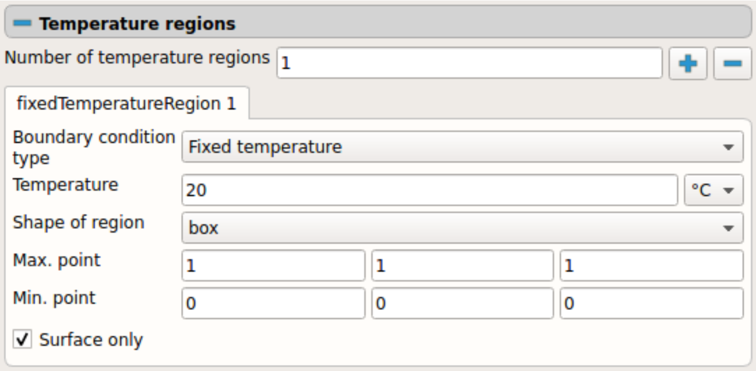
18.3.2 Temperature regions

Temperature regions offers a posibility to set boundary condition for the temperature field. Therefore, temperature regions can be set only if "Heat transfer only" or "Add thermal effects" is on. The boundary condition can be a Dirichlet boundary condition - directly setting temperature value at the point or Neumann type - heat flux is prescribed.

Number of temperature regions is controlled by the item "Number of temperature regions". Based on its value, the appropriate number of panels appear, each of them with these options:

- "Number of temperature regions" This sets how many temperature regions you want to apply.
- "Boundary condition type" is a selection tool for the type of temperature boundary condition. The options are "Fixed temperature", "Fixed heat flux" "Fixed heat transfer" and "Adiabatic". According to this selection the options for needed quantities will show up. Note that, if there is a part of boundary with unspecified temperature boundary condition adiabatic boundary is used.
- "Temperature" Specify the value of temperature in this region that will be fixed through the whole simulation.
- "Heat flux" is where you select the heat flux over the given boundary region.
- "Ambient temperature" sets the ambient temperature of surrounding of this surface from the outer side.
- "Heat transfer coefficient" sets the heat transfer coefficient, that describes the willingnes of the solid to accept the ambient temperature.
- The remaining entries have, mutatis mutandis, the same meaning as described in the previous section 18.3.1.

The example of the settings of the fixed quantity regions is shown in the figure 18.9.


Temperature regions

Number of temperature regions: 1

fixedTemperatureRegion 1

Boundary condition type: Fixed temperature

Temperature: 20 °C

Shape of region: box

Max. point: 1 1 1

Min. point: 0 0 0

☑ Surface only

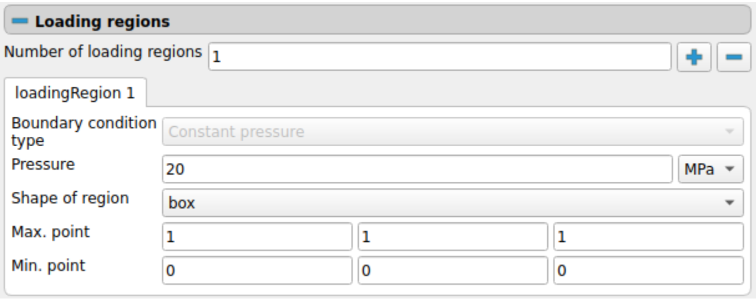
Figure 18.9: TFEA – Temperature regions

18.3.3 Loading regions

In loading regions the user can prescribe a constant loading with pressure on selected part of the solid boundary. A possible setting of the loading boundary condition is demonstrated on Fig. 18.10.

How many of these loading reginos are used in the simulations controls the entry "Number of loading regions". For each boundary region the following is to be set.

- "Boundary condition type" so far the only optoin is "Constant pressure" which performs a constant loading with prescribed pressure on the chosen boundary region.
- "Pressure" controls the pressure value of the loading pressure.
- The remaining entries controls the shape of the region where the BC is to be applied and their meaning is the same as for Temperature regions" or "Zero displacement reginos".



Loading regions

Number of loading regions: 1

loadingRegion 1

Boundary condition type: Constant pressure

Pressure: 20 MPa

Shape of region: box

Max. point: 1 1 1

Min. point: 0 0 0

Figure 18.10: TFEA – Loading regions

18.4 POST-PROCESSING

This section allows user to define how the post-processing (generation of a report) will be done.

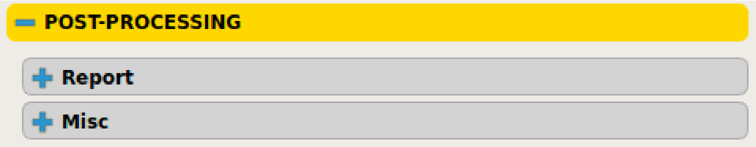
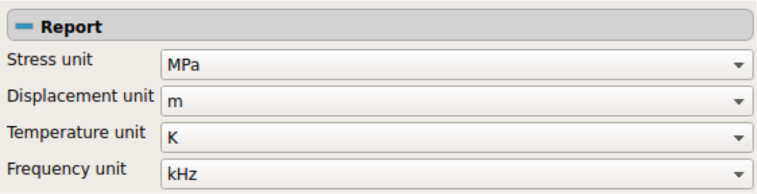


Figure 18.11: TFEA – Postprocessing

18.4.1 Report



Report

Stress unit: MPa

Displacement unit: m

Temperature unit: K

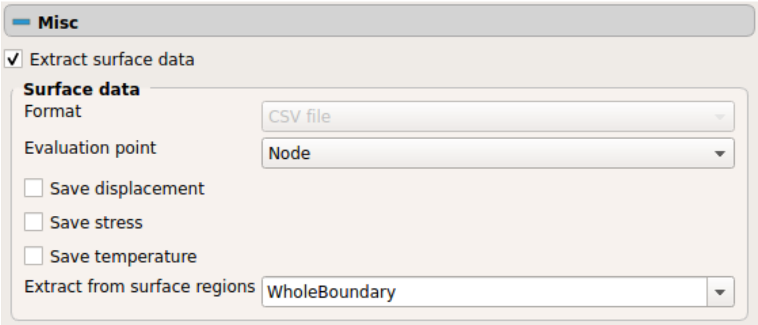
Frequency unit: kHz

Figure 18.12: TFEA – Report

TFEA report does not offer so many possibilities as TCFD report, it just lets you choose the units of reported quantities. The structure of the report section is iliuminated the figure 18.12.

- The selection "Stress unit" defines the stress unit that will be used in the simulation report.
- The selection "Displacemenet unit" defines the displacement unit that will be used in the simulation report.
- The selection "Temperature unit" defines the unit of the temperature (if temperature is computed) that will be used in the simulation report.
- The selection "Frequency unit" defines the unit of the Frequency (if modal analysis is performed) that will be used in the simulation report.

18.4.2 Misc



Misc

☑ Extract surface data

Surface data

Format: CSV file

Evaluation point: Node

□ Save displacement

□ Save stress

□ Save temperature

Extract from surface regions: WholeBoundary

Figure 18.13: TFEA – Misc

The "Misc" section currently contains possibility to write surace data in a form of a CSV file. An example of the section is depicted in the figure 18.13.

- The checkbox "Extract surface data" allows the user to create a CSV file constaining the values of chosen fields on selected boundary.
- In the entry "Format" the choice of the output fomat is to be made. The only currently available format is *CSV file*. The CSV file has in the first three columns the point coordinates, then the columns with field values start. The displacement columns are marked as *Ux*, *Uy* and *Uz* meaning displacements in the directions of coordinate axes. Similarly, the stress columns are *Sxx*, *Syy*, *Szz*, *Sxy*, *Syz* and *Szx*. The temperature column is labeled with *T*.
- The selection box "Evaluation point" makes the decision where the extracted data are located. The user can choose from the node coordinates or center of surface faces.
- In the checkbox ""Save displacement" you select whether you want the displacement field to be saved.
- In the checkbox ""Save stress" you select whether you want the stress tensor field to be saved.

- In the checkbox ""Save temperature" you select whether you want the temperature field to be saved.
- In the selection list "Extract from surface region" you select the surface regions where you are interested in the chosen quantities. You can select *WholeBoundary*, *FSIInterface* (in case of FSI simulation) or any *zero displacement region* or *temperature region*. You can select ultiple items here. The resulting file is stored in the result directory (i.e., `TFEA/RESULTS`) under the name `BoundaryData.csv`.

Chapter 19

TFEA – Configuration File Options

This section contains the list of all keywords of the TFEA module, that can appear in the *.tcae file. General format of the *.tcae file is described in section 3.3.

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
heatTransferOnly	yes	–	yes
▶ *Perform heat transfer simulation without thermal expansion effects or force loading.*			
thermalEffects	yes	–	yes
▶ *Selects whether thermal effects are taken into account.*			
doFSI	yes	–	yes
▶ *Selects whether to perform FSI.*			
doModalAnalysis	yes	–	yes
▶ *Selects whether to perform Modal analysis.*			
rotation	yes	–	yes
▶ *Selects whether the solids rotates (if not determined by FSI).*			
TFEA-gravity	yes	–	yes
▶ *Selects whether the gravity forces are included in the simulation.*			
fsi-source	TCFDResults	–	yes*
▶ *Where to look for FSI quantities.*			
fsi-force	yes	–	yes*
▶ *Whether to map forces for FSI.*			
fsi-temperature	yes	–	yes*
▶ *Whether to map temperature for FSI.*			
fsi-htc	yes	–	yes*
▶ *Whether to map heat transfer coefficient for FSI.*			
fsi-patches	yes	–	yes*
▶ *If TCFDResults is chosen as FSI source, you need to provide a list of CFD patches from which you want to map the FSI quantities.*			
fsi-files	yes	–	yes*
▶ *If CSV files are chosen as FSI source here you list them. The files are imported into the project*			
fsi-referenceFrame	2	–	yes*
▶ *Selects TCFD reference frame that corresponds to the solid, i.e., the rotation is taken from this TCFD frame.*			
fsi-useRelativePressure	yes	–	yes*

Keyword ▶ ***Description***	Allowed / sample values	Units	Mandatory
▶ *Controls whether we map relative or total presssure from TCFD case.*			
fsi-useAveragedQuantities	yes	—	yes*
▶ *Controls whether we use time-averaged quantities for mapping from TCFD case.*			
fsi-saveCSVfile	yes	—	yes*
▶ *Save mapped quantities to an CSV file.*			
modalPreload	yes	—	yes*
▶ *Decides whether the frequency analysis os computed for unloaded or loaded body, only for frequency step.*			
modalAnalysis-useFixes	yes	—	yes*
▶ *Decides whether the boundary conditions (namely the zero displacements) are considered in eigenanalysis calculation.*			
numberOfEigenfrequencies	6	—	yes*
▶ *How many eigenfrequencies are computed, only for frequency step.*			
TFEA-gravitationalAcceleration-magnitude	10	—	yes*
▶ *Magnitude of the gravitational acceleration.*			
TFEA-gravitationalAcceleration-direction	-1 0 0	—	yes*
▶ *Direction of the gravitational acceleration.*			
angularVelocity	1_RPM	—	yes*
▶ *Angular velocity of rotation of the solid body, for loading or thermal-loading step, either rad/s or RPM.*			
axisOrigin	1 0 0	—	yes*
▶ *Axis around whih the body rotates, for loading or thermal-loading step.*			
axisDirection	1 0 0	—	yes*
▶ *Directioin of axis of rotation, for loading or thermal-loading step.*			
TFEA-numberOfProcessors	6		yes
▶ *Number of Open MP threads used.*			
fea-solver-loading	iterative		yes
▶ *Selects between direct SPOOLES, direct PaStiX and iterative linear algebra solver for TFEA loading simulation.*			
fea-preconditioner-loading	cholesky		yes*
▶ *Selects between cholesky and diagonal preconditioner for iterative solver to linear algebra solver of TFEA.*			
fea-solver-modalAnalysis	PaStiX		yes

Keyword ▶ ***Description***	**Allowed / sample values**	**Units**	**Mandatory**
▶ *Selects between direct SPOOLES and direct PaStiX linear algebra solver for TFEA modal analysis simulation.*			
finiteElementOrder	2	–	yes
▶ *Choose order of finite elements used for structural computations.*			
nonlinearEquations	yes	–	yes
▶ *Controler whether to use full governing equations (yes) or linearized (no).*			
stressErrorEstimate	Zienkiewicz-Zhu	–	no
▶ *Error estimate method to stress field.*			
heatFluxErrorEstimate	gradient	–	no
▶ *Error estimate method to heat flux field.*			
numberOfSolidMaterials	2	–	no
▶ *Number of materials used in the simulation.*			
1_material-name	Steel	–	yes
▶ *Choice of the name of material used in FEA computations.*			
1_material-type	isotropic	–	yes
▶ *Choice whether the material is isotropic (default) or orthotropic.*			
1_material-density	4800	kg/m^3	yes
▶ *Choose material density.*			
1_material-youngModulus	4.1E11	Pa	yes*
▶ *Choose Young modulus for isotropic material.*			
1_material-youngModuli	4.1E5 4.1E5 4.1E5	MPa	yes*
▶ *Choose Young moduli for orthootropic material.*			
1_material-poissonRatio	0.4	–	yes*
▶ *Choose Poisson ratio for isotropic material.*			
1_material-poissonRatios	0.4 0.4 0.4	–	yes*
▶ *Choose Poissons ratio for orthotropic material.*			
1_material-shearModuli	4.1E5 4.1E5 4.1E5	MPa	yes*
▶ *Choose shear moduli for orthootropic material.*			
1_material-heatCapacity	446	J/(kg.K)	yes*

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
▶ *Choose heat capacity, only relevant for heat structural computations.*			
1_material-heatConductivity	46	W/(m.K)	yes*
▶ *Choose heat conductivity, only relevant for heat structural computations.*			
1_material-thermalExpansion	25E-6	$\mathrm{K}^{-}1$	yes*
▶ *Choose thermal expansion coefficient, only relevant for heat structural computations.*			
numberOfInterfaces	2	–	no
▶ *Number of interfaces connecting two components.*			
1_interface-leadingComponent	Co1	–	no
▶ *Name of the leading component in the interface.*			
1_interface-followingComponent	Co2	–	no
▶ *Name of the following component in the interface.*			
orientation-system	cylindrical	–	yes*
▶ *Choose between cylindrical and rectangular orientation system.*			
orientation-pointAxis1	1 0 0	–	yes*
▶ *Only for cylindrical orientation system, one point on the rotation axis.*			
orientation-pointAxis2	1 0 0	–	yes*
▶ *Only for cylindrical orientation system, second point on the rotation axis.*			
orientation-pointX	1 0 0	–	yes*
▶ *Only for rectangularal orientation system, point on the x-axis.*			
orientation-pointXY	1 1 0	–	yes*
▶ *Only for rectangularal orientation system, point in the xy-plane but not on x-axis.*			
numberOfZeroDisplacementRegions	1	–	yes
▶ *Choose number of regions with zero Dirichlet BC for displacement.*			
1_zeroDisplacementRegion-shape	rectangle	–	yes*
▶ *Choose shape of the region - rectangle or cylinder.*			
1_zeroDisplacementRegion-radius	0.4	m	yes*
▶ *Choose radius of the region.*			
1_zeroDisplacementRegion-distance	0.4	m	yes*

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
▶ *Choose distance to the structure in which the points are selected.*			
1_zeroDisplacementRegion-axis	1 0 0	—	yes*
▶ *Choose direction of axis along which we are fixing the structure, for cylindrical region.*			
1_zeroDisplacementRegion-origin	1 0 0	—	yes*
▶ *Choose origin of axis along which we are fixing the structure, for cylindrical region.*			
1_zeroDisplacementRegion-height	0.4	m	yes*
▶ *Choose height of the cylinder in which we are fixing coordinates measured from origin, only for cylindrical region. If not given, or non-positive value submited, the cylinder is infinite.*			
1_zeroDisplacementRegion-minVal	-1 -1 -1	—	yes*
▶ *The lower left corner of the rectangle.*			
1_zeroDisplacementRegion-maxVal	1 1 1	—	yes*
▶ *The upper right corner of the rectangle.*			
1_zeroDisplacementRegion-useScaleFactorFromTMESH	yes	—	no
▶ *The STL file specifying this boundary condition will be scaled with the same constant as geometry files used in TMESH.*			
1_zeroDisplacementRegion-onlySurfaceNodes	yes	—	yes
▶ *Take only surface nodes within the region.*			
1_zeroDisplacementRegion-fsiComplement	yes	—	yes
▶ *The surface region is (part of) complement to the interface between fluid and solid.*			
1_zeroDisplacementRegion-regionComplement	yes	—	yes
▶ *The surface region is (part of) complement to the remaining regions.*			
1_zeroDisplacementRegion-fsiIntersection	both	—	yes
▶ *Where to assgin nodes on common to this region and Fluid-Structure Interaction interface.*			
numberOfTemperaureRegions	1	—	yes
▶ *Choose number of regions with temperature-type boundary condition.*			
1_temperatureRegion-name	reg_1	—	yes*
▶ *Choose name of th region.*			
1_temperatureRegion-type	fixedTemperature	—	yes*
▶ *Select the temperature condition type.*			

Keyword ▶ ***Description***	**Allowed / sample values**	**Units**	**Mandatory**
1_temperatureRegion-shape	rectangle	–	yes*
▶ *Choose shape of the region - rectangle or cylinder.*			
1_temperatureRegion-radius	0.4	m	yes*
▶ *Choose radius of the region.*			
1_temperatureRegion-distance	0.4	m	yes*
▶ *Choose distance to the structure in which the points are selected.*			
1_temperatureRegion-axis	1 0 0	–	yes*
▶ *Choose direction of axis along which we are fixing the structure, for cylindrical region.*			
1_temperatureRegion-origin	1 0 0	–	yes*
▶ *Choose origin of axis along which we are fixing the structure, for cylindrical region.*			
1_temperatureRegion-height	0.4	m	yes*
▶ *Choose height of the cylinder in which we are fixing coordinates measured from origin, only for cylindrical region. If not given, or non-positive value submited, the cylinder is infinite.*			
1_temperatureRegion-minVal	-1 -1 -1	–	yes*
▶ *The lower left corner of rectangle.*			
1_temperatureRegion-maxVal	1 1 1	–	yes*
▶ *The upper right corner of rectangle.*			
1_temperatureRegion-onlySurfaceNodes	yes	–	yes*
▶ *Take only surface nodes within the region.*			
1_temperatureRegion-temperature	300_K	–	yes*
▶ *Set the temperature value in this region.*			
1_temperatureRegion-heatFlux	300_K	–	yes*
▶ *Set the heat flux value in this region.*			
1_temperatureRegion-ambientTemperature	300_K	–	yes*
▶ *Set the ambient temperature value used in heat transfer condition in this region.*			
1_temperatureRegion-heatTransferCoefficient	300_K	–	yes*
▶ *Set the heat transfer coefficient value on this surface.*			
1_temperatureRegion-useScaleFactorFromTMESH	yes	–	no

Keyword ▶ *Description*	Allowed / sample values	Units	Mandatory
▶ *The STL file specifying this boundary condition will be scaled with the same constant as geometry files used in TMESH.*			
1_temperatureRegion-fsiComplement	yes	–	yes
▶ *The surface region is (part of) complement to the interface between fluid and solid.*			
1_temperatureRegion-regionComplement	yes	–	yes
▶ *The surface region is (part of) complement to the remaining regions.*			
1_temperatureRegion-fsiIntersection	both	–	yes
▶ *Where to assgin nodes on common to this region and Fluid-Structure Interaction interface.*			
1_loadingRegion-name	reg_1	–	yes*
▶ *Choose name of th region.*			
1_loadingRegion-type	constantPressure	–	yes*
▶ *Select the loading condition type.*			
1_loadingRegion-shape	rectangle	–	yes*
▶ *Choose shape of the region - rectangle or cylinder.*			
1_loadingRegion-radius	0.4	m	yes*
▶ *Choose radius of the region.*			
1_loadingRegion-distance	0.4	m	yes*
▶ *Choose distance to the structure in which the points are selected.*			
1_loadingRegion-axis	1 0 0	–	yes*
▶ *Choose direction of axis along which we are fixing the structure, for cylindrical region.*			
1_loadingRegion-origin	1 0 0	–	yes*
▶ *Choose origin of axis along which we are fixing the structure, for cylindrical region.*			
1_loadingRegion-height	0.4	m	yes*
▶ *Choose height of the cylinder in which we are fixing coordinates measured from origin, only for cylindrical region. If not given, or non-positive value submited, the cylinder is infinite.*			
1_loadingRegion-minVal	-1 -1 -1	–	yes*
▶ *The lower left corner of rectangle.*			
1_loadingRegion-maxVal	1 1 1	–	yes*
▶ *The upper right corner of rectangle.*			

Keyword ▶ ***Description***	**Allowed / sample values**	**Units**	**Mandatory**
1_loadingRegion-onlySurfaceNodes	yes	–	yes*
▶ *Take only surface nodes within the region.*			
1_loadingRegion-pressure	300_MPa	–	yes*
▶ *Set the loading pressure value in this region.*			
1_loadingRegion-useScaleFactorFromTMESH	yes	–	no
▶ *The STL file specifying this boundary condition will be scaled with the same constant as geometry files used in TMESH.*			
1_loadingRegion-fsiComplement	yes	–	yes
▶ *The surface region is (part of) complement to the interface between fluid and solid.*			
1_loadingRegion-regionComplement	yes	–	yes
▶ *The surface region is (part of) complement to the remaining regions.*			
1_loadingRegion-fsiIntersection	both	–	yes
▶ *Where to assgin nodes on common to this region and Fluid-Structure Interaction interface.*			
reportTFEAUnit-stress	1_MPa	–	yes*
▶ *Choose the stress unit for report.*			
reportTFEAUnit-displacement	1_m	–	yes*
▶ *Choose the displacement unit for report.*			
reportTFEAUnit-temperature	1_K	–	yes*
▶ *Choose the temperature unit for report.*			
extractBoundaryData	yes	–	no
▶ *Save data from selected boundary.*			
extractBoundaryData-format	csv	–	no
▶ *Format of extracted data file.*			
extractBoundaryData-evaluationPoint	csv	–	node
▶ *Position where the surface data are evaluated, in the node coordinates or face centres.*			
extractBoundaryData-displacement	yes	–	no
▶ *Save displacement values on the surface.*			
extractBoundaryData-stress	yes	–	no
▶ *Save stress values on the surface.*			

Keyword	Allowed / sample values	Units	Mandatory
▶ ***Description***			
extractBoundaryData-temperature	yes	–	no
▶ *Save temperature values on the surface.*			
extractBoundaryData-surfaceRegions	FSIInterface	–	no
▶ *Vector of strings specifying boundary region of interest.*			
periodicSegment	yes	–	no
▶ *Whether periodic boundary conditions are applied.*			
periodicSegment-numberOfRegions	5	–	no
▶ *How many segments the whole structre is composed of.*			
periodicSegment-origin	1 0 0	–	no
▶ *Origin of axis of cyclic symmetry.*			
periodicSegment-direction	0 1 0	–	no
▶ *Direction of axis of cyclic symmetry.*			

Part V

TCAA

Chapter 20

TCAA – Introduction

What is TCAA?

TCAA is an engineering software for Computational Acoustic Analysis. TCAA was designed to calculate acoustic noise from transient CFD results, both in time and frequency domain. The approach used within TCAA is the so called "Acoustic Analogy". The implementation used in TCAA is based on the *libAcoustics* project[25]. The input to TCAA is a set of CFD quantities stored on faces of source surface. The quantitites are velocity, density and pressure, on top of that also the face normal and area are needed. This input can be read from disk (previously saved with TCFD) or taken directly from TCFD run in the same project.

20.1 Acoustic Analogy

20.1.1 Theory

Acoustic analogies provide a way to calculate pressure fluctuations in a cheaper way than solving Navier-Stokes equations in very high resolution (the spatial resolution is so high the usage in real-world is almost impossible).

There are more types of analogies, the most common one is *Ffowcs Williams-Hawkings*[30] with its *Farassat1A* formulation[31].

In short, the steps to get Ffowcs Williams-Hawkings equations are

1. Introduce a sound-producing surface. This can be a surface surrounding the noise source or surface of the noise source directly (blades of a propeller, for example). In turbomachinery applications our approach usually is to create a cylinder surrounding the rotation zone. Both the volume mesh and the surface mesh need to be fine enough, the rule of thumb is 25 cells on a wave-length to catch the corresponding frequency (frequency 1000 Hz requires cell size at least 0.01372 m).

2. Rearrange the fluid dynamics equations in such a way that we have a wave operator applied on pressure on the left-hand side and other quantities grouped in source terms on the right-hand side. The right-hand side now contains monopole (thickness) dipole (loading) surface sources and quadrupole volume sources. In our implementation we

neglect the quadrupole sources, those are the non-linear sources from turbulence. The monopole sources describe the noise generated by the displacement of fluid as the body passes. The dipole sources model the noise that results from the unsteady motion of the force distribution on the body surface.

3. Now we simplify the remaining terms. We use Green functions and rewrite time derivatives as space derivatives which give us the final equation

$$p^{'}(\mathbf{x},\, t) = p^{'}_{T}(\mathbf{x},\, t) + p^{'}_{L}(\mathbf{x},\, t), \tag{20.1}$$

where $p^{'}_{T}$ is thickness noise term

$$4\pi p^{'}_{T}(\mathbf{x},\, t) = \int_{f=0} \left[\frac{\rho_0(\dot{v}_n + v_{\dot{n}})}{r|1-M_r|^2}\right]_{\mathrm{ret}} \mathrm{d}S + \int_{f=0} \left[\frac{\rho_0 v_n(r\dot{M}_r + cM_r - cM^2)}{r^2|1-M_r|^3}\right]_{\mathrm{ret}} \mathrm{d}S \tag{20.2}$$

and $p^{'}_{L}(\mathbf{x},\, t)$ is the loading noise term

$$\begin{aligned} 4\pi p^{'}_{L}(\mathbf{x},\, t) = \frac{1}{c}\int_{f=0} \left[\frac{\dot{l}_r}{r|1-M_r|^2}\right]_{\mathrm{ret}} \mathrm{d}S + \int_{f=0} \left[\frac{l_r - l_M}{r|1-M_r|^2}\right]_{\mathrm{ret}} \mathrm{d}S \\ + \frac{1}{c}\int_{f=0} \left[\frac{l_r(r\dot{M}_r + cM_r - cM^2)}{r^2|1-M_r|^3}\right]_{\mathrm{ret}} \mathrm{d}S. \end{aligned} \tag{20.3}$$

The symbols in equations (20.2) and (20.3) have the following meaning,

- f is the surface describing function.
- ρ_0 is the reference density.
- r is the distance to the observer.
- M_r is the Mach number in radiation direction.
- M is magnitude of local Mach number.
- c is the speed of sound.
- $\dot{v}_n = \dot{v}_i n_i$, $v_{\dot{n}} = v_i \dot{n}_i$, where v_i and n_i are the $i-$th components of velocity and surface normal vector, respectively.
- $l_r = l_i r_i$, $\dot{l}_r = \dot{l}_i r_i$, $l_M = l_i M_i$, where r_i are components of the radiation direction vector and $l_i = \mathbb{P}_{ij} n_j$ are the components of local force intensity with $\mathbb{P}_{ij} = (p - p_0)\mathbb{I} + \rho\mathbf{u}(\mathbf{u} - \mathbf{u}_f)$ being the components of Lighthill compress tensor and $\mathbf{u}_f$ surface velocity.
- The subscript *ret* signalizes quantities evaluated in retarded time $\tau = t - \frac{r}{c}$.

This list of steps is just to sketch where the equations came from and how they look. For more detailed explanation see for example [26], [27] and references therein.

Taking a look on the equation (20.1) we see the acoustic analogies give us the acoustic pressure value in a chosen location and no volumetric field. We call these locations *observer location* or just *observer*. For each observer the analogy calculation needs to be run separately. Don't worry, this is way faster then TCFD run so no need to restrict on one or two observers.

20.1.2 Implementation

As already stated, the starting point of development of TCAA is the library *libAcoustics* [25]. We made extensive changes to fit our needs the best. First, we converted the original implementation to be compatible with *OpenFOAM* version we use. The biggest difference is TCAA works as post-processing to CFD simulation. Opposed to *libAcoustics* which runs simultaneously with *OpenFOAM*, TCAA reads quantities listed above from disk. This way we find more flexible and better fitting our framework. The main advantege is that the user does not have to rerun CFD simulation to (re-)do CAA analysis. This is in particular useful when an observer is to be added or the user wants to try another CAA setting.

The data coming to TCAA are required to be obtained with fixed time step (this is because the conversion to frequency domain (Fourier analysis) requires fixed time stepping). There are of course drawbacks too, our method obviously consumes more memory and the reading/writing take some time. Nonetheless, we favor post-processing approach as we find these drawbacks subtle. Another improvement w.r.t. *libAcoustics* is the support for incompressible CFD simulations. The last step of acoustic analysis, the conversion from time domain to frequency domain, is implemented completely differently, independent on *libAcoustics* and the following section discusses it.

20.2 Signal Processing

20.2.1 Theory

Once the time series of acoustic pressure values are computed for each observer (the acoustic analogy simulation is finished), transformation to the frequency domain follows. As a result we obtain the *Sound Pressure Level* and *Power Spectral Density* quantities for each observer. These quantities can be interpreted as a measure of loudness of each tone (pressure fluctuation frequency) and overall loudness in the observer location.

This process we call *Signal Processing* and consists of following steps. At first, we clean the time signal from unwanted frequencies with the chosen signal filter (can be *none*, *Butterworth*, *Chebyshev1* or *Chebyshev2*). These are standard types of signal filters and their implementation from the Python SciPy[29] library is used in TCAE. For more details consult the internet[33] or follow the references in Scipy[29] manual.

The cleaned signal is then transformed to frequency domain with *Fourier Transform*. In order to get reasonable results from acoustics the CFD results which play the role of input needs to be good enough. If the CFD is of questionable quality, acoustic results cannot be good. Moreover, we want to use only the CFD data from times when the simulated flow is already stabilized (i.e., the first few rotations of a propeller simulation should be skipped). Then there is the time-step size Δt. We mentioned already that we require CFD to be run with a fixed time step size. This is simply because the input to Fourier Transform is correct and the *sampling frequency* $f_s = \frac{1}{\Delta t}$ is well-defined only for constant fixed Δt. The sampling frequency gives us a limitation for the accuracy of the signal processing results. The frequencies above *Nyquist frequency* $f_N = \frac{f_s}{2}$ cannot be trusted as they are affected by the aliasing phenomenon.

The length of the signal is another aspect that influences the quality of TCAA results. The resolution in frequency domain Δf is proportional to the length of the signal T. The relation is roughly

$$\Delta f = \frac{1}{2T}.$$

The exact number depends on the signal filter used. Let's say the user is interested in frequencies from $50Hz$, this equation gives us the minimum length of the signal coming to Fourier Transform is $0.01s$.

If the signal is long enough more sophisticated treatment is possible. The so-called *Welch method* [28] splits the signal on a chosen number of (possibly overlapping) segments. On each segment *window function* is applied and the result is Fourier-transformed. Finally, the results of the segments are averaged which gives the final result.

The role of the *window functions* is to force the Fourier Transform to focus more on the data from the middle of the interval than on the corner values. The most important reason is the fact that the Fourier Transform silently assumes the last value matches the first value which is true for a perfectly periodic signal of the length of a whole number of periods but not true for real-life applications. This means not using any filter potentially leads to ill input to Fourier Transform. The window function properties and more information about them can be found for instance on Wikipedia [34].

Further reading on signal processing can be found in example in [32].

20.2.2 Implementation

We implement the signal processing procedure as a Python script where we make use of signal processing tools from the standard package *SciPy*[29].

The script computes and saves the following

- **Filtered temporal signal.** Taking the direct result of Acoustic Analogies simulation we apply signal filteres to suppress unwanted frequencies.

- **Sound Pressure Level (SPL).** The filtered signal is transfered to frequency domain with Fourier Transform. let $\hat{f}_\omega$ be the value corresponding to frequency ω of the transformed temporal signal p

  $$\hat{f}_\omega = \left[\mathcal{F}(p)\right]_\omega,$$

  where $\mathcal{F}$ is the Fourier Transform operator. The sound pressure level *spl* for frequency ω is then evaluated as

  $$spl = 10 \log \frac{2|\hat{f}_\omega|}{S_1 p_0},$$

  where p_0 is the *reference sound pressure* and S_1 is the normalization factor depending on the chosen window function.

- **Power Spectral Density (PSD).** Power spectral density differs from Sound pressure level in the aspects that it is *power* so $\hat{f}_\omega$ appears here in second power and it is also *density*, we divide with sampling frequency f_s. The equation is

  $$psd = 10 \log \frac{2|\hat{f}_\omega|^2}{f_s S_2 p_0^2},$$

  where S_2 is (another) normalization factor depending on the chosen window function.

- **Loudest frequencies (both, in SPL and PSD meaning).** For both SPL and PSD we find peaks the quantity attains its highest values. The number of peaks is chosen by user.

20.2.3 Sound reconstruction

One can also reconstruct the sound from the processed signal. TCAA module automatically generates MP3 and WAV files for each observer located at `<project>/simulationRunX/TCAA/{Speedline}_{Point}/signalProcessing/{Observer}-sound.mp3.`

20.3 TCAA directory structure

TCAA directory is subdirectory of TCAE case (see 3.2).

The whole directory should be read-only, and its basic structure is shown in the table 20.1, because user might want to extract some useful data for postprocessing or to look at the logs.

🗀 project_folder	Project directory
└ 🗀 simulationRun	
└ 🗀 TCAA	all files, that belong to TCAA module
└ 🗀 report	HTML report with its images
└ 🗀 1_1	data and results for the first speedline, first point
└ 🗀 system	directory with basic settings of the case
└ 🗀 acousticData	directory with acoustic results in time domain
└ 🗀 signalProcessing	directory with acoustic results in frequency domain

Table 20.1: TCAA – Directory structure

Chapter 21

TCAA – GUI Setup & Options

Module TCAA in the GUI can be added / removed by using the modules buttons (4.1) and is selected by the clicking on the *TCAA* item in the *Pipeline Browser*, or on one of its *Output Ports*, which are:

- Settings
  Using this port the keywords and their values of TCAA module are displayed as a table in *SpreadSheet View*.

- Source Surface
  If the input data for CAA analysis is loaded, it is shown in *RenderView* through this port.

- Report
  If the CAA calculation if finished and the HTML report is done, it will be displayed in *HTML View* through this port.

After selection the TCAA, the user can see what is depicted in the Fig. 21.1. In the *Properties Panel*, there are menus, their contents are thoroughly described in following sections: "SIMULATION", "PHYSICAL CONSTANTS", "SIGNAL PROCESSING" and "POST-PROCESSING".

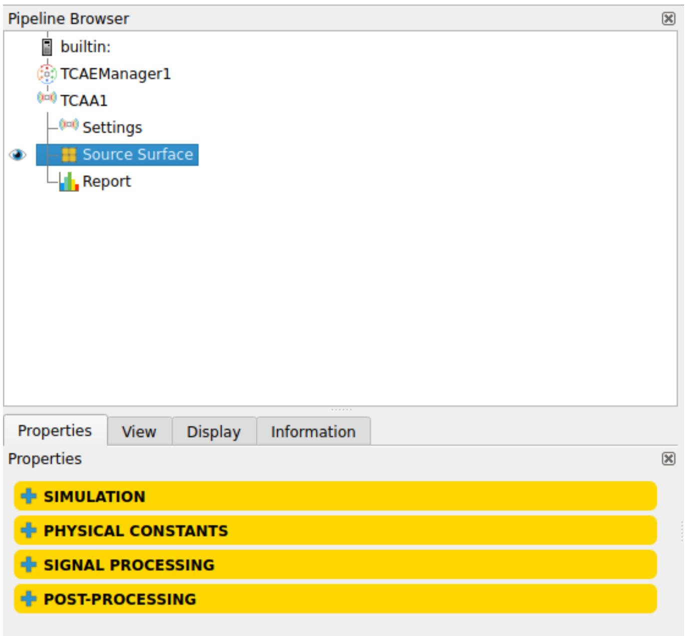
21.1 SIMULATION

Under this section the general CAA settings as well as acoustic observers are to be set (Fig.21.2).

21.1.1 CAA Analysis

The layout of the CAA Analysis card is shown in Figure 21.3. Here the user is asked to choose acoustic analogy used for the simulation and source data.

- The selection box "CAA source" lets the user choose which data will be used as a source to acoustic analysis. The possibilities are "External data" or "CFD simulation". If the first option is chosen, the source surfaces needs to be selected later, whereas if the second option is selected the data saved in the TCFD module (in this project) will be used.



Pipeline Browser

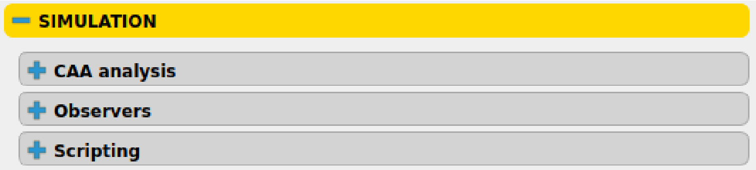
- builtin:
- TCAEManager1
- TCAA1
  - Settings
  - Source Surface
  - Report

Properties | View | Display | Information

Properties

- SIMULATION
- PHYSICAL CONSTANTS
- SIGNAL PROCESSING
- POST-PROCESSING

Figure 21.1: TCAA – *Pipeline Browser* and *Properties Panel*



- SIMULATION
  - CAA analysis
  - Observers
  - Scripting

Figure 21.2: TCAA – Simulation.

- In the selection box "Acoustic analogy" the acoustic analogy used to get the acoustic data is chosen. Currently only "Ffowcs-Williams Hawkings" analogy is supported.

- "Equation formulation" selection box is visible only if "Ffowcs-Williams Hawkings" analogy is used and selects the formulation of the acoustic equations. Currently only "Farassat 1A" formulation is possible.

- The "Custom time range" checkbox controls whether we use all input data (unchecked) or we use shorter time rage (selected).

- "Time start" option controls the simulatino time where the acoustic analogy evaluation starts. It is only visible if th checkbox "Custom time range" is selected.

- "Time end" option controls the simulatino time where the acoustic analogy evaluation ends. It is only visible if th checkbox "Custom time range" is selected.

- "Number of surfaces" says how many external surfaces are used for the acoustic simulation. All surfaces are then used as sources to one simulation. So the final result can be seen as a superposition of the partial results corresponding to single surface. This option is only visible if "External data" are chosen as "CAA source".

- The entry "Surface name" sets name of the corresponding acoustic surface.

- The entry "Data directory" reads path to the directory containig the source data of the corresponding surface. One directory per time step named as the time step value in seconds is assumed to be found here.

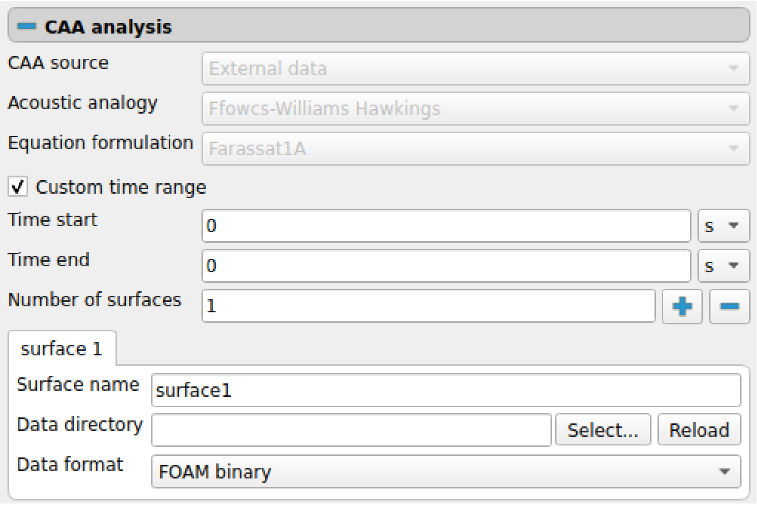
- The option "Data format" tells what are the format of the data that are stored on the path "Data directory". The three possibilities are "FOAM binary", "FOAM ascii" and "FOAM ascii". We recommend to use "FOAM binary" format as it is faster and more memory efficient then other the other options. The ascii formats are meant only for test and research purposes.

21.1.2 Observers

In the section "Observers" the locations where we are interested for the acoustic analysis are specified. "Number of observers" determines how many of these locations ("observers") will be processed. For each observer we then need to give the following data.

- "Observer name" is the name of the observer, by this name the observers will be reffered to in the final CAA report.

- "Position" specifies the location of the observer. This should be outside of the acoustic surface (in cases the acoustic surface is a closed domain). The observers lying on the acoustic surface



CAA analysis

CAA source: External data

Acoustic analogy: Ffowcs-Williams Hawkings

Equation formulation: Farassat1A

☑ Custom time range

Time start: 0 s

Time end: 0 s

Number of surfaces: 1

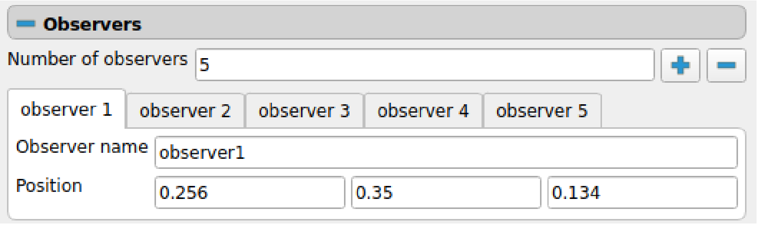
surface 1

Surface name: surface1

Data directory:

Data format: FOAM binary

Figure 21.3: TCAA – CAA analysis.



Observers

Number of observers: 5

observer 1 | observer 2 | observer 3 | observer 4 | observer 5

Observer name: observer1

Position: 0.256 0.35 0.134

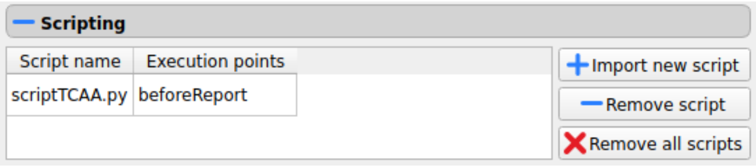
Figure 21.4: TCAA – SIMULATION: Observers.

21.1.3 Scripting

Similarly as in the TCFD or TFEA module, TCAA offers usage of external (user-defined) Python scripts (see figure 21.5). They are expected to be written in basic Python3 and the user choose one or more "Execution points". The script is executed in all chosen execution points, these are:

- "afterWrite
- "beforeCalculation"
- "afterCalculation"
- "beforeReport"
- "afterReport"

We believe the names of evaluation points are descriptive enough and their meanings are clear.

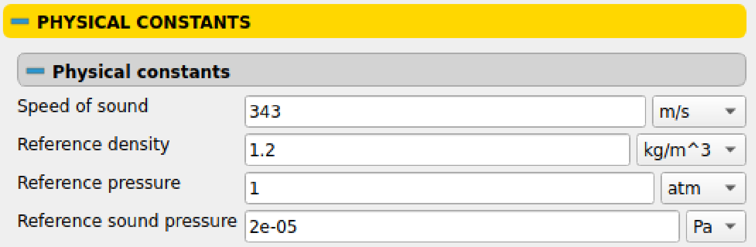


Script name	Execution points
scriptTCAA.py	beforeReport

Figure 21.5: TCAA – SIMULATION: Scripting.

21.2 PHYSICAL CONSTANTS

In this section (Fig. 21.6) the physical properties of the medium in which the sound is propagated are prescribed:



Speed of sound	343	m/s
Reference density	1.2	kg/m^3
Reference pressure	1	atm
Reference sound pressure	2e-05	Pa

Figure 21.6: TCAA – Material settings.

- "Speed of sound" simply defines the speed of sound.
- "Reference pressure" is the pressure of the undisturbed medium.
- "Reference denstiy" is the denstiy of the undisturbed medium.
- "Reference acoustic pressure" is the reference pressure for decibel calculation. In aeroacoustics the usually used value is $2e - 5$ Pa.

21.3 SIGNAL PROCESSING

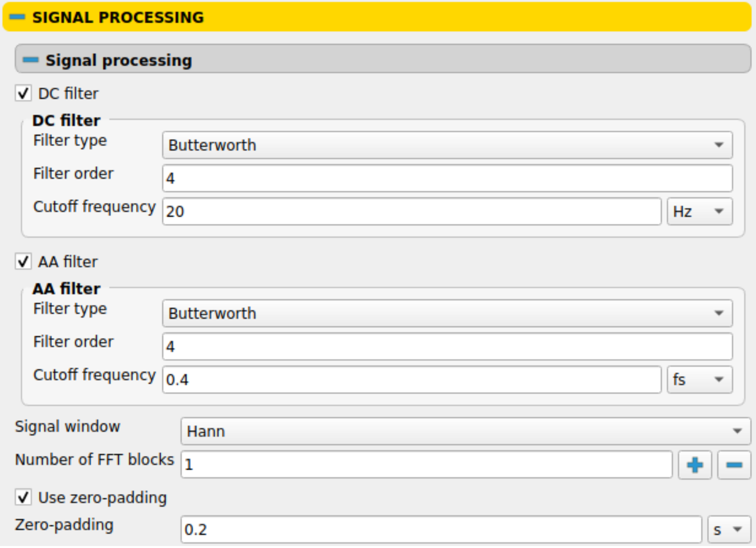
The signal processing step picks up the results of Acoustic Analogy step and converts them to frequency domain as described earlier. The tools to use available in TCAA are discussed below.

21.3.1 Signal processing

The signal processing procedure sketched in the previous paragraphs is controled by the following setting. Note that the entries "Filter type", "Filter order", "Cutoff frequency", "Passband ripple" and "Stopband ripple" apply for both "DC filter" and "AA filter" boxes.

- "DC filter" checkbox control whether DC filter is used or not. DC filter is a high-pass filter filtering out the low frequencies.
- "AA filter" checkbox controls the usage of AA filter. AA (Anti-Aliasing) filter is a low-pass filter filtering out the high frquencies that can cause undesired aliasing effect.
- "Filter type" controls the type of the filter applied on the time signal. The options are "Butterworth", "Chebyshev1" and "Chebyshev2" (also known as "Inverse Chebyshev filter").
- "Filter order" sets the order of the selected filter.
- "Cutoff frequency" Specifies the cutoff frequency of corresponding filter. It can be given in frequency units or as multiple of sampling frequency (indicated as "fs" in the unit selection).
- "Passband ripple" sets the maximum loss in the pass band for "Chebyshev1" filter.
- "Stopband ripple" sets the minimum attenuation in the stop band for "Chebyshev2" filter.
- "Signal window" selects the window function applied on the signal/signal segment. For the shapes and properties of window functions please discuss [34] The "Rectangle" option means all entries has the same weight, in othe words, no filter is applied.
- "Number of segments" controls number of segments used in the Welch method (see discussion above). One segment means the signal is not divided hence no averaging of the results is needed.
- "Overlap" If more then one segment is used, the segment overlap (percentual value) can be used. Each signal window has its optimal overlap that is default for chosen window. This optimal overlap balances between the effectivness of data use.
- "Use zero-padding" checkbox controls the usage of zero-padding before entering the Fourier Transform.
- "Zero-padding" entry sets the length of the input to Fourier Transform. The entry is set as length of time interval, the array intering the Fourier Transform is then padded with zeros to have length corresponding to this time value.

The example of the signal processing settings is shown in the Figure 21.7.



SIGNAL PROCESSING

Signal processing

☑ DC filter

DC filter
Filter type: Butterworth
Filter order: 4
Cutoff frequency: 20 Hz

☑ AA filter

AA filter
Filter type: Butterworth
Filter order: 4
Cutoff frequency: 0.4 fs

Signal window: Hann
Number of FFT blocks: 1

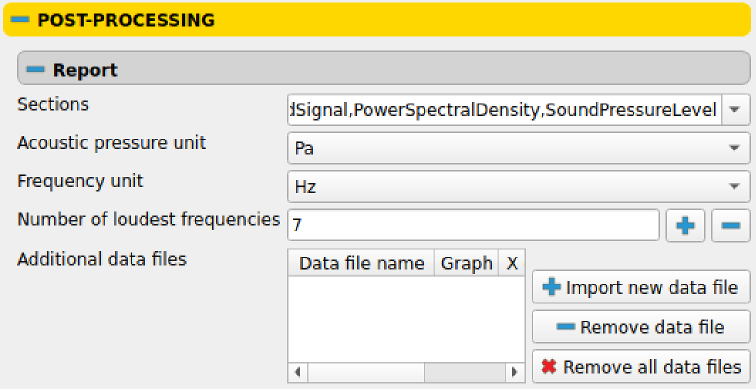
☑ Use zero-padding
Zero-padding: 0.2 s

Figure 21.7: TCAA – Zero displacement regions

21.4 POST-PROCESSING

This section allows user to define how the post-processing (generation of a report) will be done.

21.4.1 Report



POST-PROCESSING

Report

Sections: ıSignal,PowerSpectralDensity,SoundPressureLevel
Acoustic pressure unit: Pa
Frequency unit: Hz
Number of loudest frequencies: 7
Additional data files

Data file name	Graph	X

Figure 21.8: TCAA – Report

In this section the properties of the HTML report, which is created when the calculation is finished, are to be set. The section is depicted in Fig 21.8.

- The selection "Sections" defines sections to appear in the calculation report.
- The selection "Acoustic pressure unit" defines the acoustic pressure unit that will be used in the calculation report.
- The selection "Frequancy unit" defines the frequancy unit that will be used in the calculation report.

- The selection "Number of loudest frequencies" sets the number of loudest frequencies that will be listed in tables together with their Sound Pressure Level and Power Spectral Density values.
- The table "Additional data files" is used to add user data to the graphs in the final report from the calculation. The data should be provided in the form of Gnuplot-readable data files, i.e. text files with white-space-separated equal-length columns of numbers. The graph into which the data are to be added is chosen in the second column, the column indices to be used in the remaining two (first column has index 1).

Chapter 22

TCAA – Configuration File Options

This section contains the list of all keywords of the TCAA module, that can appear in the *.tcae file. General format of the *.tcae file is described in section 3.3.

Keyword	Allowed / sample values	Units	Mandatory
▶ ***Description***			
CAAsource	CFDSimulation	–	yes
▶ *selects whether the input data to acoustic simulation are read from TCFD case of this project or from external files*			
tcaa-rhoInf	1	kg/m^3	yes
▶ *density of the fluid in rest*			
tcaa-pInf	1	atm	yes
▶ *pressure in the fluid in rest*			
tcaa-p0	2e-5	Pa	yes
▶ *reference pressure for the acoustic (decibel-based) quantity calculation*			
speedOfSound	343	m/s	yes
▶ *speed of sound in the simulated fluid*			
analogy-type	FfowcsWilliamsHawkings	–	yes
▶ *Name of the analogy that will be used in the acoustic simulation*			
analogy-formulation	Farassat1A	–	yes
▶ *type of the formulation of chosen analogy*			
numberOfObservers	4	–	yes
▶ *number of locatinos where the acoustic quantities are to be calculated*			
N_observer-position	(1 1 1)	–	yes
▶ *the location of the N-th observer*			
N_observer-name	obs01	–	yes
▶ *name of the N-th observer*			
numberOfSurfaces	3	–	yes*
▶ *number of external source surfaces*			
N_surface-name	surf01	–	yes*
▶ *name of the N-th external source surface*			
N_surface-format	FOAMbinary	–	yes*
▶ *format of the input data for the N-th surface*			
customTimeRange	yes	–	yes
▶ *choice whether we will do acoustic analysis for all the times in source surfaces or only for limited time range*			

Keyword ▶ ***Description***	Allowed / sample values	Units	Mandatory
N_N_startTime	1	s	yes*
▶ *time where the custom time range for specific speedline point starts*			
signalProcessing-DCfilter	yes	–	yes
▶ *Use DC filter on acoustic signal*			
signalProcessing-DCfilter-filterType	butterworth	–	yes
▶ *type of the DC filter applied on the raw acoustic signal*			
signalProcessing-DCfilter-cutoff	60	Hz	yes*
▶ *cutoff frequency of DC filter*			
signalProcessing-DCfilter-order	4	–	yes*
▶ *order of DC filter*			
signalProcessing-DCfilter-passbandRipple	3	dB	yes*
▶ *Maximum loss in the passband, if Chebyshev1 filter is used*			
signalProcessing-DCfilter-stopbandRipple	40	dB	yes*
▶ *Minimum attenuation in the stopband, if Chebyshev2 filter is used*			
signalProcessing-AAfilter	yes	–	yes
▶ *Use AA filter on acoustic signal*			
signalProcessing-AAfilter-filterType	butterworth	–	yes
▶ *type of the AA filter applied on the raw acoustic signal*			
signalProcessing-AAfilter-cutoff	60	Hz	yes*
▶ *cutoff frequency of AA filter*			
signalProcessing-AAfilter-order	4	–	yes*
▶ *order of AA filter*			
signalProcessing-AAfilter-passbandRipple	3	dB	yes*
▶ *Maximum loss in the passband, if Chebyshev1 filter is used*			
signalProcessing-AAfilter-stopbandRipple	40	dB	yes*
▶ *Minimum attenuation in the stopband, if Chebyshev2 filter is used*			
signalProcessing-numberOfSegments	9	–	yes
▶ *number of segments for the Welch method*			

Keyword ▶ ***Description***	**Allowed / sample values**	**Units**	**Mandatory**
signalProcessing-window	hann	–	yes
▶ *window function that will be applied on the acoustic signal or its segment*			
signalProcessing-overlap	0.5	–	yes
▶ *overlap between the segments in Welch method*			
signalProcessing-useZeroPadding	yes	–	yes
▶ *controls when zero-padding before FFT is used*			
signalProcessing-zeroPadding	0.5	s	yes
▶ *the input to FFT is padded with zeros to have length corresponding to given time interval length*			
reportSections	SimulationStats ObserverSummary	–	yes
▶ *selects the sections for the report generated at the end of simulation*			
numberOfLoudestFrequencies	8	–	yes
▶ *number of loudest frequencies that will be tabled in the final report*			
reportUnit-frequency	1_Hz	–	yes
▶ *frequency unit used in the final report*			
reportUnit-pressure	1_Pa	–	yes
▶ *pressure unit used in the final report*			

Part VI

TOPT

Chapter 23

TOPT – Introduction

What is TOPT?

TOPT is an optional software module inside the engineering simulation environment TCAE. TOPT was designed to enable multiple simulation runs based on variable parameters. The TOPT running mode can be either "DOE" (Design Of Experiment) or "optimize". Mode DOE can be either explicit or implicit. Mode optimize can be either external or built-in. TOPT running loop can easily include external software for CAD model creation, or for parametric space transformation, or optimization. All the external software pieces can be added to the TOPT work scheme. TOPT is flexible enough to work with in-house codes, commercial codes, and open-source. The following scheme shows how TOPT works.

23.1 TOPT case directory structure

TOPT project case directory is subdirectory of TCAE case.

TOPT workflow typically runs multiple simulations. Individual simulations are called *Runs*. The very first Run is written down directly in the main case directory the same way as a normal TCAE simulation (directories TMESH,TCFD, TFEA and TOPT). All the next simulations are always written down to the subdirectories with case directory name with indexes. The subdirectory for every next TOPT Run has the same structure as a normal TCAE case (see 3.2).

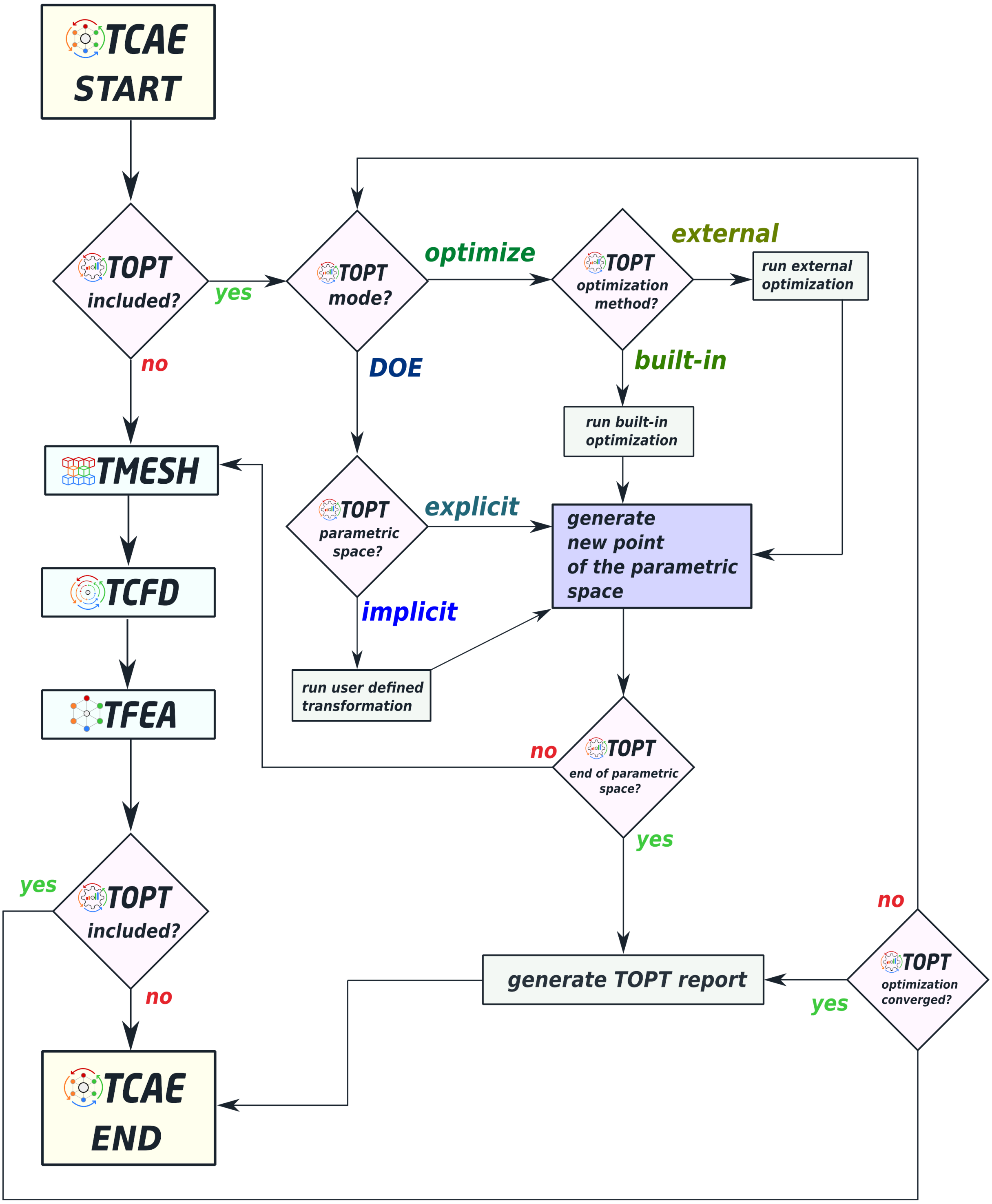


Figure 23.1: TOPT – workflow scheme

🗀 project_folder	Project directory
└ 🗀 simulationRun1	Run 1, case directory
└ 🗀 TMESH	Run 1, all files, that belong to TMESH module
└ 🗀 TCFD	Run 1, all files, that belong to TCFD module
└ 🗀 TFEA	Run 1, all files, that belong to TFEA module
└ 🗀 simulationRun2	Run 2, case directory
└ 🗀 TMESH	Run 2, all files, that belong to TMESH module
└ 🗀 TCFD	Run 2, all files, that belong to TCFD module
└ 🗀 TFEA	Run 2, all files, that belong to TFEA module
└ . . .	
└ 🗀 simulationRunN	Run N, case directory
└ 🗀 TMESH	Run N, all files, that belong to TMESH module
└ 🗀 TCFD	Run N, all files, that belong to TCFD module
└ 🗀 TFEA	Run N, all files, that belong to TFEA module
└ 🗀 commonData	Data common to the project, for example TOPT report
└ 🗀 logs	Folder containing text files with output logs produced by the TCAE
└ 🗋 setup	Setup file, that was used to initialize this case
└ 🗋 info	
└ 🗋 pointIDIndices	

Table 23.1: TOPT – Case directory structure

Chapter 24

TOPT – Configuration File Options

This section contains the list of all keywords of the TOPT module, that can appear in the *.tcae file. General format of the *.tcae file is described in section 3.3.

Keyword	Allowed / sample values	Units	Mandatory
▶ ***Description***			
TOPT-operationMode	DOE | optimize	–	no
▶ *The operation mode can be either DOE (Design of Experiment) or optimize.*			
TOPT-optimizationMethod	built-in | external	–	no
▶ *If the operation mode is optimize, then the used optimization method can be either built-in or external.*			
TOPT-optimizationAlgorithm	golden section search	–	no
▶ *If the optimization method is built-in then the used method is golden-section-search (currently the only option).*			
TOPT-parametricSpaceType	implicit | explicit	–	no
▶ *Parametric space type can be either explicit or implicit.*			
TOPT-initializationScript	path	–	no
▶ *Script that is executed in the beginning of every Run.*			
TOPT-optimizationFunction	name	–	no
▶ *Name of the optimization function.*			
TOPT-convergenceCheckingScript	path	–	no
▶ *Script that may customize the convergence of optimization function.*			
TOPT-relativeTolerance	1e-3	–	no
▶ *Relative tolerance of convergence of optimization function.*			
TOPT-absoluteTolerance	1e-5	–	no
▶ *Absolute tolerance of convergence of optimization function.*			
TOPT-optimizeFor	maximum | minimum	–	no
▶ *Searching for minimum or maximum of the optimization function.*			
TOPT-numberOfParameters	2	–	no
▶ *Number of parameters.*			
TOPT-N_parameter-kind	custom | keyword	–	no
▶ *Nth parameter kind. It can be a TCAE keyword or uder defined - custom.*			
TOPT-N_parameter-type	real | integer | string	–	no
▶ *Nth parameter type.*			
TOPT-N_parameter-group	0	–	no
▶ *Nth parameter group ID.*			

Keyword ▶ ***Description***	**Allowed / sample values**	**Units**	**Mandatory**
TOPT-N_parameter-keyword	2_referenceFrame-angularVelocity	–	no
▶ *Nth parameter name in TCAE (if TOPT-N_parameter-kind is keyword).*			
TOPT-N_parameter-name	R1	–	no
▶ *Nth parameter name (if TOPT-N_parameter-kind is custom).*			
TOPT-N_parameter-datatype	linspace | logspace | list	–	no
▶ *Nth parameter type of distribution in the parametric space.*			
TOPT-N_parameter-values	1 2 3	–	no
▶ *Nth parameter list of values (if TOPT-N_parameter-datatype is list).*			
TOPT-N_parameter-rangeMin	0	–	no
▶ *Nth parameter minimal value.*			
TOPT-N_parameter-rangeMax	1	–	no
▶ *Nth parameter maximal value.*			
TOPT-N_parameter-rangeSamples	10	–	no
▶ *Nth parameter - number of points.*			
TOPT-transformationScript	path	–	no
▶ *Script that is before every Run.*			
TOPT-maxIterations	100	–	no
▶ *Maximal number of Runs.*			
TOPT-numberOfTrackedQuantities	3	–	no
▶ *Number of evaluated quantities.*			
TOPT-N_trackedQuantity-type	built-in | custom	–	no
▶ *Nth tracked quantity type.*			
TOPT-N_trackedQuantity-acceptRange	0 1	–	no
▶ *Nth tracked quantity acceptable range of values.*			
TOPT-N_trackedQuantity-script	path	–	no
▶ *Nth tracked quantity script (if TOPT-N_trackedQuantity-type is custom).*			
TOPT-N_trackedQuantity-speedline	1	–	no
▶ *Nth tracked quantity speedline ID.*			

Keyword ▶ ***Description***	Allowed / sample values	Units	Mandatory
TOPT-N_trackedQuantity-point	1	–	no
▶ *Nth tracked quantity point ID.*			
TOPT-N_trackedQuantity-builtinQuantity	axial force	–	no
▶ *Nth built-in tracked quantity name.*			
TOPT-meshingPolicy	automatic | never | always	–	no
▶ *Mesh can be always newly generated for every new Run, or never, or automatic.*			

Index

Bibliography

[1] H. Versteeg, W. Malalasekera, *An Introduction to Computational Fluid Dynamics: The Finite Volume Method (2nd Edition)*, (Feb 26, 2007)

[2] LeVeque R. J., *Numerical Methods for Conservation Laws* Birkhauser Verlag, 1990

[3] H. Lomax, David W. Zingg, Thomas Pulliam, *Fundamentals of Computational Fluid Dynamics*, 2001

[4] Dvořák R., Kozel K., *Matematické modelovaní v aerodynamice* Vydavatelství ČVUT, 1996.

[5] Schnerr G. H. And Sauer J., Physical and Numerical Modeling of Unsteady Cavitation Dynamics, Proc. 4th International Conference on Multiphase Flow, New Orleans, U.S.A., 2001.

[6] Schiavello B., Visser F., Pump Cavitation - Various NPSHR Criteria, NPSHA Margins, and Impeller Life Expectancy, Proc. of the twenty-fifth INTERNATIONAL PUMP USERS SYMPOSIUM, 2009

[7] CFD-online web pages: *Navier-Stokes equations*
(*http://www.cfd-online.com/Wiki/Navier-Stokes_equations*, date: 30. 1. 2014).

[8] Wilcox D.C., *Turbulence Modeling for CFD* , DCW Industries

[9] OpenFOAM web pages: *OpenFOAM Documentation*
(*http://www.openfoam.org/docs/*, date: 30. 1. 2014).

[10] Wikipedia web pages: *Finite Volume Method*
(*http://en.wikipedia.org/wiki/Finite-volume_method*, date: 30. 1. 2014).

[11] NASA web pages: *Isentropic Flow*
(*http://www.grc.nasa.gov/WWW/BGH/isentrop.html*, date: 30. 1. 2014).

[12] OpenFOAM web pages: *OpenFOAM mesh description*
(*http://www.openfoam.org/docs/user/mesh-description.php*, date: 30. 1. 2014).

[13] Wikipedia web pages: *Message Passing Interface*
(*http://en.wikipedia.org/wiki/Message_Passing_Interface*, date: 30. 1. 2014).

[14] CFD-online web pages: *Best practice guidelines for turbomachinery CFD* (*http://www.cfd-online.com/Wiki/Best_practice_guidelines_for_turbomachinery_CFD*, date: 30. 1. 2014).

[15] Wikipedia web pages: *ParaView* (*http://en.wikipedia.org/wiki/ParaView*, date: 25. 2. 2015).

[16] ParaView web pages: *ParaView License* (*http://www.paraview.org/paraview-license/*, date: 25. 2. 2015).

[17] Wikipedia web pages: *Euler Number* (*https://en.wikipedia.org/wiki/Euler_number_(physics)*), date: 6. 7. 2017).

[18] Wikipedia web pages: *Vapor Pressure* (*https://en.wikipedia.org/wiki/Vapor_pressure*), date: 6. 7. 2017).

[19] NIST web pages: *Water Coefficients* (*http://webbook.nist.gov/cgi/cbook.cgi?ID=C7732185&Mask=4&Type=ANTOINE&Plot=on#AN* date: 6. 7. 2017).

[20] Wikipedia web pages: *Cauchy Stress* (*http://en.wikipedia.org/wiki/Cauchy_stress_tensor*, date: 9. 6. 2020).

[21] Wikipedia web pages: *von Mises Stress* (*http://en.wikipedia.org/wiki/Von_Mises_yield_criterion*, date: 9. 6. 2020).

[22] G. Dhondt: *CalculiX CrunchiX USER'S MANUAL version 2.16* (*http://www.dhondt.de/ccx_2.16.pdf*, Nov 24, 2019)

[23] R. Asaro, V. Lubarda, *Mechanics of Solids and Materials*, 2006

[24] J. Schöberl, *NETGEN An advancing front 2D/3D-mesh generator based on abstract rules*, 1997

[25] Epikhin, A., Evdokimov, I., Kraposhin, M., Kalugin, M., Strijhak, S., *Development of a Dynamic Library for Computational Aeroacoustics Applications Using the OpenFOAM Open Source Package*, 2015

[26] Wikibooks web pages: *Analogies in Aeroacoustics* (it https://en.wikibooks.org/wiki/Engineering_Acoustics/Analogies_in_aeroacoustics), date: 29. 9. 2022

[27] Kenneth S. Brentner, F. Farassat *Modeling aerodynamically generated sound of helicopter rotors*, 2003

[28] Peter D. Welch *The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms*, 1967

[29] Virtanen, Pauli and Gommers, Ralf and Oliphant, Travis E. and Haberland, Matt and Reddy, Tyler and Cournapeau, David and Burovski, Evgeni and Peterson, Pearu and Weckesser, Warren and Bright, Jonathan and van der Walt, Stéfan J. and Brett, Matthew and Wilson, Joshua and Millman, K. Jarrod and Mayorov, Nikolay and Nelson, Andrew R. J. and Jones, Eric and Kern, Robert and Larson, Eric and Carey, C J and Polat, İlhan and Feng, Yu and Moore, Eric W. and VanderPlas, Jake and Laxalde, Denis and Perktold, Josef and Cimrman, Robert and Henriksen, Ian and Quintero, E. A. and Harris, Charles R. and Archibald, Anne M. and Ribeiro, Antônio H. and Pedregosa, Fabian and van Mulbregt, Paul and SciPy 1.0 Contributors *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*, 2020

[30] Stewart Glegg and William Devenport *Aeroacoustics of Low Mach Number Flows*, 2017

[31] F. Farassat *Derivation of Formulations 1 and 1A of Farassat*, 2007

[32] G. Heinzel, A. Rüdiger and R. Schilling *Spectrum and spectral density estimation by the Discrete Fourier transform (DFT), including a comprehensive list of window functions and some new flat-top windows.*, 2002

[33] Wikipedia web pages: *Signal Filters*
(*https://en.wikipedia.org/wiki/Filter_(signal_processing)*), date 29. 9. 2022

[34] Wikipedia web pages: *Window Funcitons*
(*https://en.wikipedia.org/wiki/Window_function*), date 29. 9. 2022

[35] C. Geuzaine and J.-F. Remacle *Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities*, 2009.

[36] Gmsh manual: *Gmsh manual*
(*https://gmsh.info/doc/texinfo/gmsh.html*), date 12.10.2022

[37] Benigni, Helmut and Schiffer-Rosenberger, Jürgen and Penninger, Gerhard and Weichselbraun, Christian and Artmann, Michael and Juhrig, Lutz and Becker, Martin and Krappel, Timo and Jaberg, Helmut *Cavitation as a limiting factor for the empowering of a Kaplan turbine–CFD calculations, test rig results and operational experience*, Hydro 2020, 1–10, 2020