



AutorInnen:
Thomas Mitterlehner
Eva Kobler
Georg Steinbichler

Institut:
Institut für Polymer-
Spritzgießtechnik und
Prozessautomatisierung

Datum:
März 2019

HANDBUCH



Einstieg in OpenFoam® und chtMultiRegion anhand eines anwendungsorientierten Beispiels

Vorwort

Im Rahmen des Projekts AM 4 Industry wurde die Auslegung und Simulation von konturnahen Kühlkanälen erforscht. Unter anderem wurde das Ziel verfolgt, WerkzeugbauingenieurenInnen ein Tool zur weiteren Optimierung von Kühlkanalgeometrien zur Verfügung zu stellen. Insbesondere bei der Gestaltung von konturnahen und regellosen Kühlkanälen kann die Simulation mit kommerziellen Simulationsprogrammen nicht ausreichend sein, weshalb zusätzliche Simulationsschritte erforderlich sind. Für diese zusätzlichen oder ergänzenden Simulationsschritte wurde bewusst der Fokus auf ein nicht kommerzielles Simulationsprogramm gelegt. Daher wurde für die Simulationen das Programm OpenFoam® (**Open Source Field Operation and Manipulation**) gewählt. OpenFoam® ist frei erhältlich und wird vorwiegend für die Lösung von Strömungsproblemen (CFD) verwendet. Es ist in C++ geschrieben und bringt bereits in der Basisversion nützliche Löser (Solver) mit sich. Darüber hinaus kann eine Vielzahl an weiteren Lösern adaptiert werden. Einer der größten Vorteile besteht darin, dass der Sourcecode und somit auch die Algorithmen frei zugänglich sind. Des Weiteren können die Codes und Berechnungen nahezu beliebig erweitert werden.

Anhand eines anwendungsorientierten Beispiels wird in diesem Handbuch der Aufbau und die Durchführung einer Simulation ausführlich erklärt. Für die Simulationen wurde der Löser „chtMultiRegion“ verwendet. Dieser wird allgemein für die Berechnung des Wärmeaustausches zwischen einem Festkörper und einem Fluid verwendet.

Ziel dieses Handbuchs ist es, PraktikernInnen in der Entwicklung, SimulationsingenieurenInnen und StudentenInnen einen anwendungsorientierten Einstieg in sowie einen Überblick über die Arbeit mit OpenFoam® zu geben. Die einzelnen Schritte wurden in neun Kapitel mit folgenden Inhalten eingeteilt:

- Kapitel 1 und 2 geben einen allgemeinen Überblick über das Simulationsbeispiel sowie über die Simulationsstrukturen von OpenFoam®.
- Kapitel 3 bis 6 zeigen, wie ein Simulationsfall in OpenFoam® aufgesetzt wird und welche Einstellungen erforderlich sind. Dabei wird genau auf die Erstellung des Rechengitters und die Zuordnung der Flächen eingegangen.
- Kapitel 7 und 8 behandeln die Durchführung der Simulation. Dabei wird auf die Wahl der Randbedingung sowie auf die Theorie der Strömungssimulation eingegangen.
- Kapitel 9 behandelt abschließend die Auswertungsmethoden, welche von OpenFoam® geboten werden sowie die Möglichkeiten zum Export der Ergebnisse für die weitere Verarbeitung.

Abschließend ist zu erwähnen, dass die OpenFoam®-Umgebung auf den ersten Blick durchaus befremdlich wirken kann, insbesondere, wenn üblicherweise auf Windows-Betriebssystemen gearbeitet wird. Hier ist mit Sicherheit Geduld gefragt. Es dauert eine gewisse Zeit, bis die Abläufe des Programms logisch erscheinen und bis Befehle, die für die Bedienung und Ausführung des Programms nötig sind, leicht von der Hand gehen. Dennoch wird sich eine intensive Einarbeitung in die Thematik lohnen, da sich so Simulationen weiter verfeinern und bessere Vorhersagen treffen

lassen. Dies wiederum kann einen klaren Wettbewerbsvorteil bieten. In diesem Sinne: Happy Foaming!

Inhaltsverzeichnis

Vorwort.....	2
Inhaltsverzeichnis.....	4
1. Einleitung.....	5
2. Aufbereitung.....	8
2.1. Konstruktion der Simulationsgeometrie.....	8
2.2. Kontrolle und Nachbearbeitung.....	9
3. Pre-Processing mit OpenFoam®	14
3.1. Gittererstellung allgemein.....	15
4. Erstes Rechengitter – „blockMesh“	17
5. Zweites Rechengitter – „snappyHexMesh“	22
6. Erstellung und Zuordnung der Regionen.....	27
7. Solving – Simulation mit „chtMultiRegion“.....	33
7.1. Theorie Strömungssimulation	33
7.2. Input-Parameter	34
7.2.1. Eingabeparameter Laminar	35
7.2.2. Eingabeparameter Turbulent.....	35
8. Ausführen der Simulation.....	39
9. Ergebnisse	40
9.1. Aufheiz- und Abkühlvorgang.....	40
9.2. Export der Messwerte	44
10. Anhang.....	46
10.1. Ordner „0“ > fluid“	46
10.2. Ordner „0“ > Solid“	50
10.3. Ordner „constant“ > fluid“	51
10.4. Ordner „constant“ > solid“	54
11. Literatur.....	56

1. Einleitung

Das Softwarepaket OpenFoam® wird üblicherweise auf Linux betrieben. In dem hier beschriebenen Fall erfolgt der Betrieb auf Ubuntu Version 14.04.5 LTS. Für die Durchführung von Simulationen ist eine Installation eines Linux-Betriebssystems erforderlich. Folgende Möglichkeiten zum Betrieb von OpenFoam® bzw. eines Linux-Systems sind möglich:

- Direkte Installation von OpenFoam® auf einem Rechner mit Linux-Betriebssystem
- Dualboot: Zwei oder mehrere Betriebssysteme (z.B. Windows und Ubuntu) sind auf einem Rechner installiert und bei jedem Systemstart kann ausgewählt werden, mit welchem Betriebssystem gestartet wird.
- Verwendung eines Serverrechners: Installation eines Linux-Betriebssystems auf einem separaten Rechner, auf den über ein Netzwerk mit z.B. Putty zugegriffen werden kann.

Hierbei ist auch anzumerken, dass OpenFoam® standardmäßig keine Benutzeroberfläche (GUI) besitzt. Somit müssen sämtliche Befehle über die Kommandozeile eingegeben werden. Es sind jedoch Benutzeroberflächen kommerziell erhältlich, die die Bedienung erleichtern können. Dazu zählen beispielsweise CAESES [1], Visual-CFD [2], SimFlow [3] oder MantiumFlow [4]. Die allgemeine Struktur von OpenFoam® ist in Abbildung 1 dargestellt.

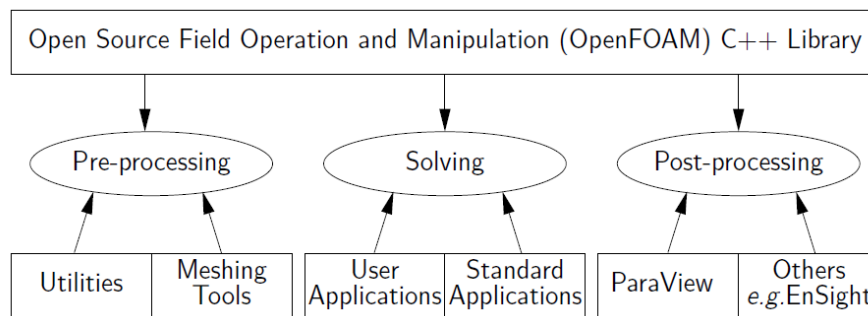


Abbildung 1: Übersicht der OpenFoam® Struktur [5]

Diese setzt sich aus den drei Teilgebieten Pre-Processing, Solving und Post-Processing zusammen. Zum Pre-Processing zählen nicht die Programme, wie z.B. Blender oder PTC Creo, sondern viel mehr die Softwarepakete, welche die Geometrie für die Löser (Solver) vorbereiten. Dazu gehören zum Beispiel „blockMesh“ oder „snappyHexMesh“. Zu den Lösern zählen die Softwarepakete, welche die Rechnungen bzw. Simulationen durchführen. OpenFoam® ist standardmäßig mit zahlreichen Lösern ausgestattet und es können aus diversen Internetquellen oder –foren weitere hinzugefügt werden. Der Löser „chtMultiRegion“ ist standardmäßig in OpenFoam® integriert und unter den „Heat transfer and buoyancy-driven flows“-Lösern zu finden.

Exakt um diesen „chtMultiRegion“-Solver dreht sich dieses Handbuch. Hierzu werden anhand eines anschaulichen Beispiels Schritt für Schritt die Vorgänge beschrieben. Zu diesem Zweck wurde ein Anwendungsfall aus der Kunststoffverarbeitung bzw. aus der Spritzgießtechnik gewählt. Die Simulation soll dazu beitragen, die Temperaturverteilung in einem Spritzgießwerkzeug darzustellen. Im Einzelnen wird ein sogenannter variothermer Spritzgießprozess dargestellt. Grundsätzlich wird bei einem Spritzgießprozess das Kunststoffgranulat, welches bei Raumtemperatur in fester Form vorliegt, in einem beheizten Zylinder aufgeschmolzen. Dies geschieht bei etwa 200°C. Nach dem Plastifizieren liegt der Kunststoff in flüssiger Form vor und

wird mit ca. 500 bar in eine Metallform injiziert. Die mittels Temperiergeräten gekühlte Metallform sorgt nicht nur für die geometrischen Abformung, sondern auch für die Abkühlung des Formteils. Weist das Formteil jedoch sehr feine Strukturen (Mikrostrukturen) auf, reicht das konventionelle Werkzeugkühlkonzept nicht aus. Hier kommt dann der zuvor genannte Variothermprozess zur Anwendung. Dabei wird die Metallform vor dem Einspritzvorgang in etwa auf die Schmelztemperatur beheizt. Noch während des Einspritzvorganges wird diese dann wieder abgekühlt. In Abbildung 2 sind die Vorgänge während des Füllvorganges beider Prozessvarianten dargestellt.

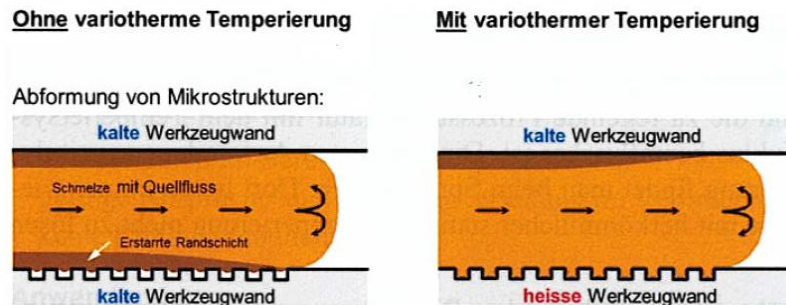


Abbildung 2: Füllvorgang beim Spritzgießen, Vergleich ohne und mit variothermer Temperierung [6]

Hierbei wird ersichtlich, dass die Kunststoffschmelze bei Kontakt mit der kalten Werkzeugwand erstarrt. Dies mindert die Fähigkeit, Mikrostrukturen abzuformen. Ist die Werkzeugwand jedoch beheizt, hat die Kunststoffschmelze genügend Zeit, um in die feinen Strukturen zu fließen. Bewerkstelligt wird der Aufheiz- und Abkühlvorgang durch zwei getrennte Kühlgeräte. Als Kühl- bzw. Heizmedium wird zumeist Wasser oder Öl eingesetzt. In der nachfolgenden Abbildung ist ein solches Temperierkonzept beispielhaft dargestellt.

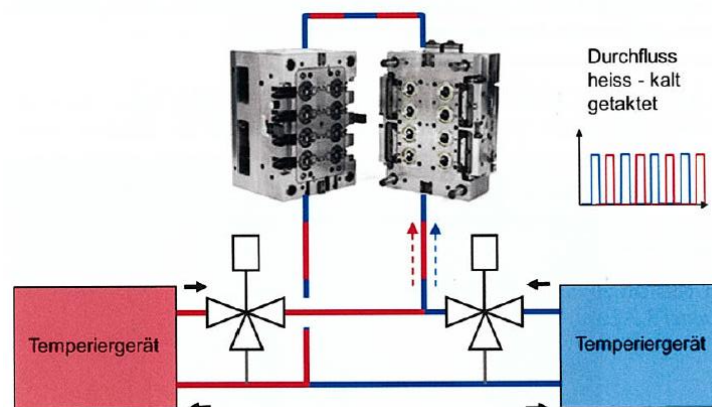


Abbildung 3: Prinzip der variothermen Flüssigtemperierung mit zwei Temperiergeräten und einer Ventil-Umschalteneinheit [6]

Um qualitativ hochwertige Kunststoffprodukte herstellen zu können, ist die Kenntnis der exakten Werkzeugwandtemperatur essenziell. Aus diesem Grund lohnt es sich, diesen Prozess zum Beispiel mithilfe von OpenFoam® zu simulieren und genauer zu betrachten.

Dieses Handbuch soll als erste Grundlage dienen, um die beschriebenen Vorgänge mittels OpenFoam® abbilden zu können. Für einen ersten Überblick sind in Tabelle 1 die einzelnen Schritte der Herangehensweise, eingeteilt in Aufbereitung, Pre-Processing, Solving und Post-Processing, dargestellt. Hinsichtlich der Modellierung und Einbindung der Geometriedaten in

OpenFoam® bieten sich verschiedenste Vorgehensweisen. Der in dieser Arbeit beschriebene Simulationsfall wird mit folgender Herangehensweise gelöst:

Tabelle 1: Vorgehensweise bei der Simulation mit OpenFoam® und chtMultiRegion

Schritt	Beschreibung	Einteilung
1	Konstruktion der Simulationsgeometrie mit PTC Creo und Export des CAD-Modells als .stl-Datei	Aufbereitung (Seite 8)
2	Kontrolle und Nachbearbeitung des CAD Modells mit Blender	Aufbereitung (Seite 9)
3	Erstellung eines ersten Rechengitters mit „ blockMesh “	Pre-Processing (Seite 17)
4	Verfeinerung des Rechengitters mit „ snappyHexMesh “	Pre-Processing (Seite 22)
5	Zuordnung der Regionen mit „ splitMeshRegions “ und „ surfaceToPatch “	Pre-Processing (Seite 27)
6	Simulation des Wärmeflusses bzw. -austausches zwischen Werkzeug und Kühlmittel mit „ chtMultiRegion “	Solving (Seite 33)
7	Darstellung der Ergebnisse mit „ ParaView “	Post-Processing (Seite 40)

Als Geometrie für das Simulationsbeispiel wurde ein länglicher Quader gewählt, welcher stirnseitig in der oberen Hälfte eine Temperierbohrung besitzt. Durch diese Bohrung soll Wasser mit einer definierten Geschwindigkeit und Temperatur fließen. Ziel ist es den Aufheiz- und Abkühlvorgang an der oberen Fläche zu bestimmen. In der nachfolgenden Abbildung ist die Simulationsgeometrie dargestellt.

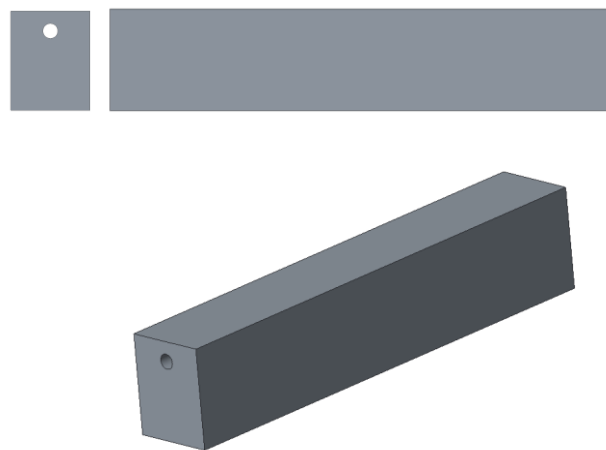


Abbildung 4: Darstellung der Simulationsgeometrie

In den folgenden Kapiteln ist das beschriebene Simulationsbeispiel Schritt für Schritt dokumentiert. Dabei werden auch immer wieder die auszuführenden Befehle erwähnt. Diese sind farblich dargestellt, wobei *PTC-Creo-Befehle* in blau, *Blender-Befehle* in gelb und *OpenFoam®-Befehle* in rot gekennzeichnet sind.

2. Aufbereitung

Unter den Punkt Aufbereitung fallen alle Schritte, die nicht mit OpenFoam® oder in einer Linux-Umgebung durchgeführt werden müssen. In diesem Handbuch bzw. im gezeigten Beispiel wurden alle Konstruktionen mit PTC Creo umgesetzt. Die Konstruktion ist jedoch nicht auf das Programm PTC Creo begrenzt, sondern kann beliebig durchgeführt werden. Einzige Bedingung ist, dass die Geometrie im .stl-Format an OpenFoam® übergeben wird. Da beim Export der Geometrie oftmals Fehler auftreten, wird diese danach zusätzlich mit dem frei erhältlichen Programm Blender [7] kontrolliert und gegebenenfalls repariert. Ein typischer Fehler ist hierbei zum Beispiel eine nicht geschlossene Fläche.

2.1. Konstruktion der Simulationsgeometrie

In einem ersten Schritt wurde damit begonnen, die Grundgeometrie mit PTC Creo zu konstruieren. Als zu simulierende Geometrie wurde der in Abbildung 4 gezeigte Stahlblock gewählt, welcher eine in Längsrichtung verlaufende Bohrung besitzt. Dies soll einen stark vereinfachten Aufbau eines Spritzgießwerkzeuges mit einer darin verlaufenden Kühlbohrung darstellen. In der nachfolgenden Abbildung sind die Maße dieser Grundgeometrie illustriert.

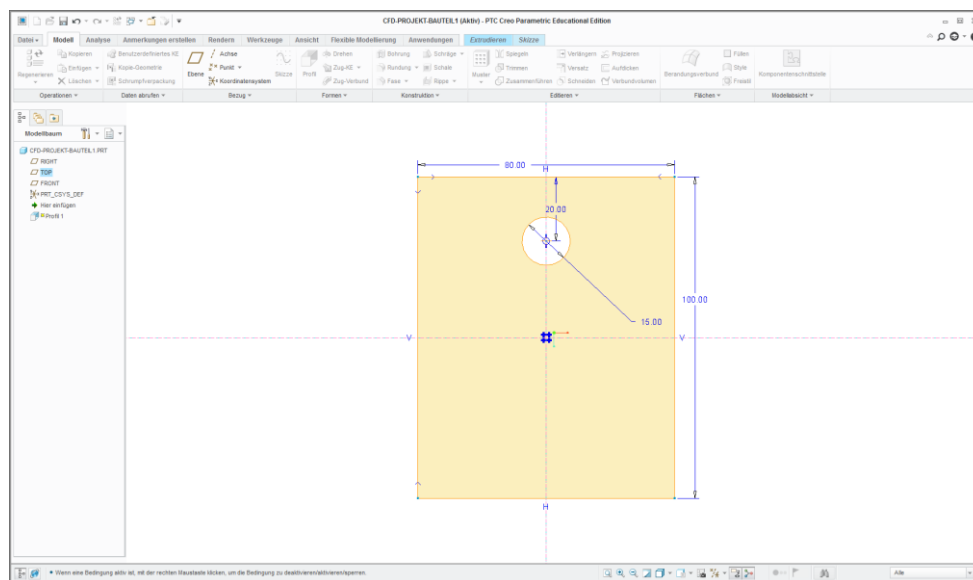


Abbildung 5: Grundgeometrie des Simulationsmodells im CAD-Programm PTC Creo

Die Breite der Grundgeometrie beträgt 80 mm und die Höhe 100 mm. Die Bohrung besitzt einen Durchmesser von 15 mm und wurde 20 mm unterhalb der oberen Fläche platziert. Diese Grundgeometrie wurde auf eine Länge von 500 mm extrudiert. Als Werkstoff wurde Stahl des Typs 1.2343 (X38 CrMoV 51) gewählt, da dieser üblicherweise für Formeinsätze bei Spritzgießwerkzeugen verwendet wird.

Im Anschluss daran wurde dieses Model als .stl-File exportiert. Die Exportierfunktion befindet sich im CAD-Programm PTC Creo unter

Datei > Speichern als > Kopie speichern

Danach erscheint der Speicherexplorer. In diesem ist der Dateityp „Stereolithographie (.stl)“ zu wählen. Nachdem mit „OK“ bestätigt wurde, erscheint ein Fenster, in dem Exporteinstellungen

gewählt werden können. Dabei ist es für die spätere Verarbeitung mit OpenFoam® notwendig, dass beim Format „ASCII“ (Amerikanischer Standard-Code für den Informationsaustausch) ausgewählt wird. Dabei wird bestimmt, ob die Datei nur in der Maschinensprache (Binär) oder auch mit einem Texteditor (ASCII) lesbar sein soll. Des Weiteren sind die Sehnenhöhe und die Winkelsteuerung einzustellen. Bei der Sehnenhöhe ist ein Wert im Bereich von 0.01 und bei der Winkelsteuerung der Wert 1 einzugeben. Durch die Angabe der Sehnenhöhe wird der maximale Abstand zwischen einer Sehne und einer Oberfläche angegeben. Je geringer die Sehnenhöhe gewählt wird, desto geringer ist die Abweichung von der tatsächlichen Bauteiloberfläche. Zum besseren Verständnis ist in Abbildung 6 eine Erläuterung abgebildet.

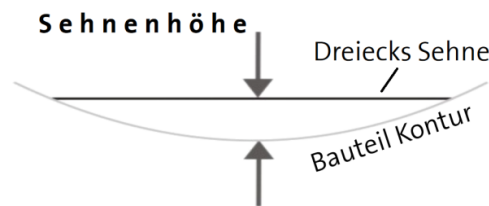


Abbildung 6: Erläuterung Sehnenhöhe [8]

Beachten Sie jedoch: Je kleiner die Sehnenhöhe gewählt wird, desto größer wird die exportierte .stl-Datei. Die Winkelsteuerung kann in einem Bereich von 0 bis 1 eingestellt werden. Dieser Wert bestimmt die Anpassung der Sehnenhöhe bei kleinen Radien. Der Wert 1 bewirkt eine genaue Anpassung der Dreiecke an kleine Radien. Hingegen bewirkt ein Wert von 0 keine Anpassung. In der nachfolgenden Abbildung 7 ist ein Vergleich zweier verschiedener Einstellungen dargestellt.

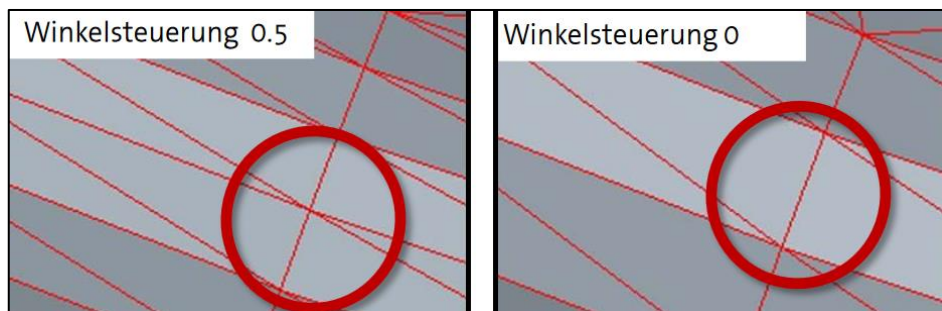


Abbildung 7: Vergleich zweier Winkelsteuerungseinstellungen [8]

Wurde das Format, Sehnenhöhe und Winkelsteuerung eingestellt, bestätigt man die Einstellungen mit „OK“ und die Geometrie wird als stl-File exportiert.

2.2. Kontrolle und Nachbearbeitung

Liegt die Geometrie als .stl-File vor, kann mit der Arbeit in Blender begonnen werden. Hierzu muss zunächst die Geometrie importiert werden. Dies kann mit

File > Import

durchgeführt werden. Um mit dem Modell optimal arbeiten zu können, muss zunächst der richtige Modus eingestellt werden. Standardmäßig ist zu Beginn der Modus „object mode“ eingestellt. Dieser muss auf „edit mode“ umgestellt werden. Einzustellen ist dies im unteren Bildschirmbereich (Abbildung 8).

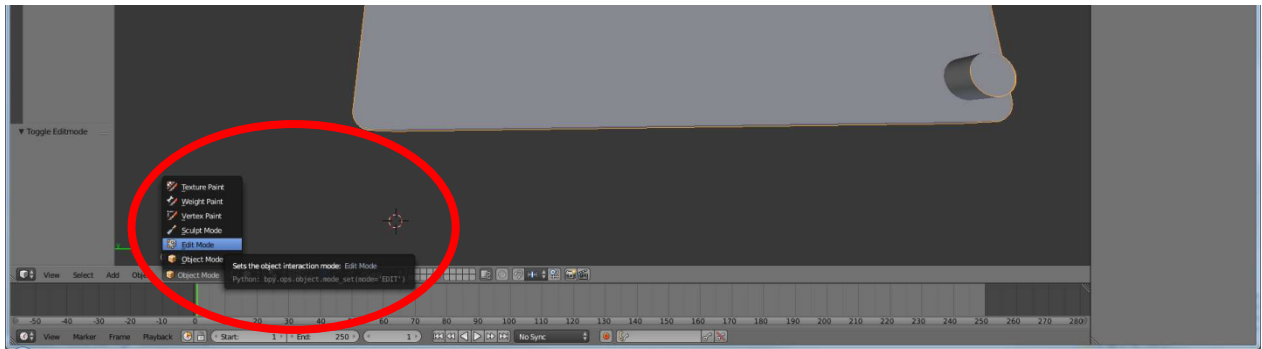


Abbildung 8: Einstellung „edit mode“ in Blender

Des Weiteren ist die Auswahl auf Fläche („face“) einzustellen. Dies kann wiederum am unteren Bildschirmrand eingestellt werden. Mit der

Taste 5

am Nummernblock kann zudem noch das Zoomverhalten verbessert werden. Sind diese Einstellungen erfolgt, kann mit der rechten Maustaste eine Fläche ausgewählt werden. Um mehrere Flächen auszuwählen, muss die Shift-Taste gedrückt und gehalten werden.

Damit OpenFoam® mit der CAD-Geometrie arbeiten kann, müssen zunächst offene Geometrieflächen geschlossen werden. Bei der angezeigten Geometrie ist beispielsweise beim Kühlkanaleinlass und –auslass die Geometrie nicht geschlossen. Um diese Öffnungen zu schließen, ist die Füllfunktion eine geeignet Wahl. Hierfür ändert man zunächst den Auswahlmodus auf Punktauswahl („vertex select“). Danach betätigt man die

c-Taste

woraufhin ein Auswahlkreis erscheint. Durch Betätigen der linken Maustaste werden alle Punkte im Auswahlkreis markiert. Mit dem Mausekranz kann der Auswahlkreis verkleinert oder vergrößert werden. Sind alle notwendigen Punkte ausgewählt, muss die rechte Maustaste betätigt werden, um die Auswahl zu beenden. Danach ist die

f-Taste

zu betätigen, um die Fläche, die die Punkte einschließt, zu füllen.

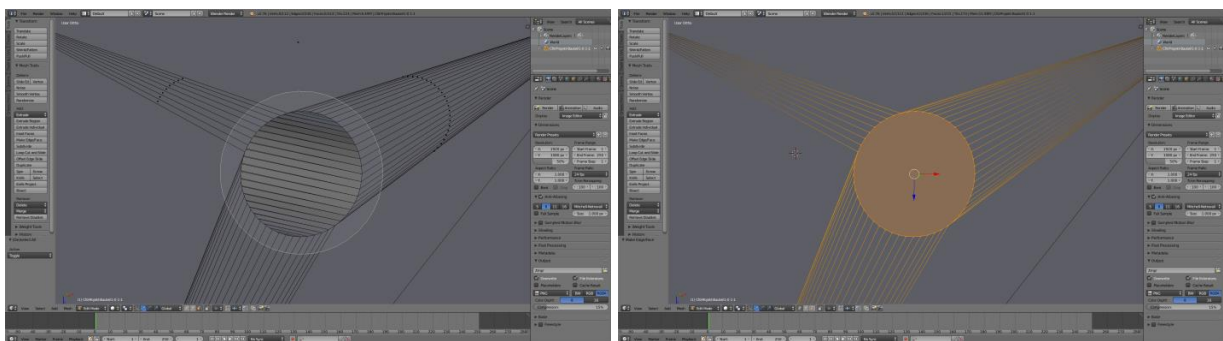


Abbildung 9: Auswahlkreis durch Betätigung der c-Taste (links), Füllen der Fläche durch Betätigung der f-Taste

Zwischen den einzelnen Füllbefehlen muss immer darauf geachtet werden, dass nicht auch noch andere Flächen ausgewählt sind. Um sicherzustellen, dass keine anderen Punkte angewählt sind, sollte zwischen verschiedenen Befehlen die

a-Taste

betätigt werden, da auf diese Weise alle gewählten Punkte abgewählt werden. Nachdem alle Geometrien erfolgreich geschlossen wurden, kann mit der Funktion

Non manifold

überprüft werden, ob tatsächlich alle Flächen geschlossen sind. Zu finden ist diese Funktion am linken unteren Bildschirmrand unter

Select > Non Manifold oder Select > Select All by Trait > Non Manifold oder durch Drücken von Shift+Ctrl+Alt+M

Ist der „edit mode“ aktiviert und die Ansicht auf „wireframe“ eingestellt, werden die offenen Bereiche hervorgehoben. Durch Drücken der

n-Taste

können zusätzlich die xyz-Koordinaten des offenen Bereichs angezeigt werden.

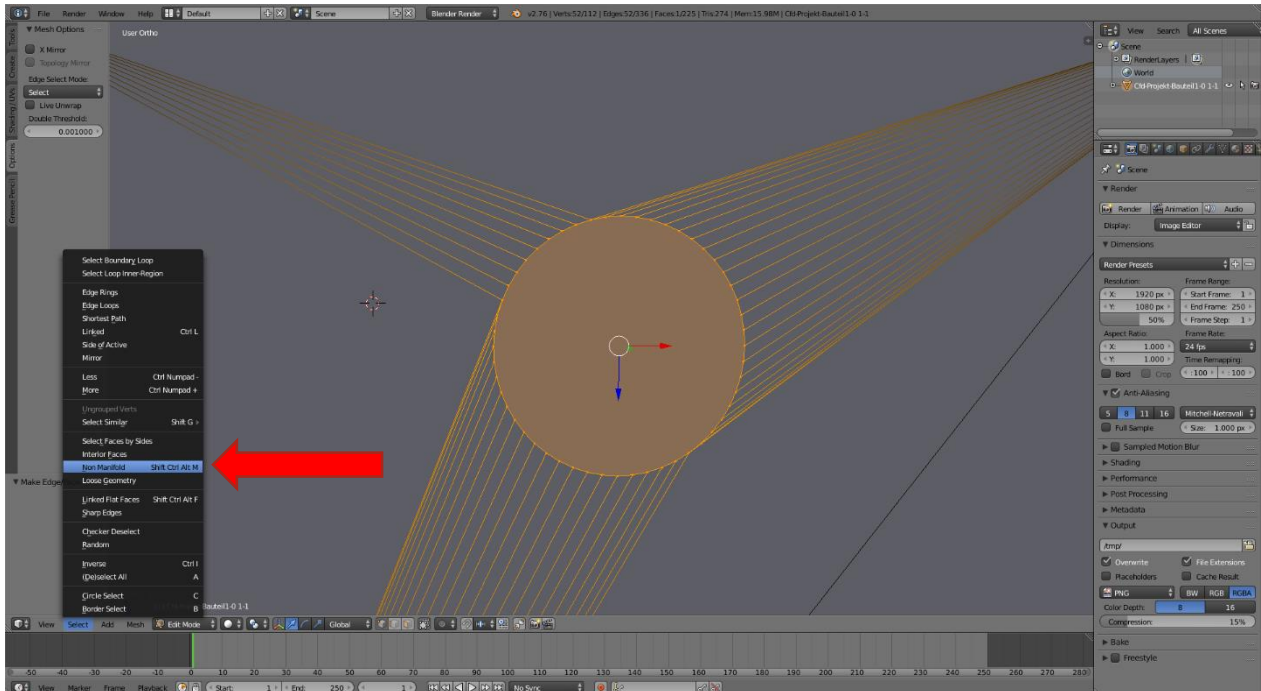


Abbildung 10: Menüpunkt „non manifold“

Nachdem sichergestellt wurde, dass alle Geometrien geschlossen sind, müssen die einzelnen Flächen definiert werden, damit diesen später in OpenFoam® bestimmte Eigenschaften, zum Beispiel Einlass und Auslass des Kühlmediums, zugewiesen werden können. Das heißt, jede Fläche, die eine eigene Eigenschaft besitzen soll, muss separat als .stl-Datei exportiert werden. Dazu zählen der Einlass, der Auslass, das Kühlrohr und der Stahlblock. Im dies durchzuführen, muss auf den Modus der Flächenauswahl gewechselt werden und die zu definierende Fläche ausgewählt werden. Anschließend ist die

p-Taste

zu betätigen und „section“ auszuwählen. Danach erscheint die ausgewählte Fläche separat im rechten Strukturbaum. Nun kann jeder Fläche ein Name zugeordnet werden. Für OpenFoam® sind folgende Definitionen wichtig:

- „inlet“
- „outlet“
- „tube“
- „wall“

Im 6. Kapitel sind die zu exportierenden Flächen bzw. Regionen dargestellt und ausführlich beschrieben. Werden beim Auswählen der Flächen einzelne Teilflächen übersehen, können diese im Nachhinein hinzugefügt werden. Dabei müssen die übersehenen Flächen ausgewählt und die p-Taste gedrückt werden. Danach sind die übersehenen Flächen sowie die Fläche, zu der diese hinzugefügt werden sollen, auszuwählen. Schließlich ist links im Menübaum

Tools > Edit > Join

der „join“-Befehl auszuführen. Dadurch werden die beiden ausgewählten Flächen zusammengeführt. Damit das Modell ordnungsgemäß simuliert werden kann, müssen für den Import in OpenFoam® folgenden .stl-Files vorhanden sein:

- „inlet“ (einzeln)
- „outlet“ (einzeln)
- „tube“ (einzeln)
- „wall“ (alle, die nicht zu „inlet“, „outlet“ oder „wall“ zählen)
- „solid“ („snappyHexMesh wall“ - alle Flächen, die später die gleiche „solid“-Randbedingung besitzen sollen)
- „fluid“ („inlet“, „outlet“ und „wall“ – alle Flächen, die später die „fluid“-Randbedingung besitzen sollen)

Damit mehrere Flächen als eine einzelne .stl-Datei exportiert werden können (z.B. „fluid“), müssen diese rechts im Strukturbaum ausgewählt werden. Die Auswahl mehrere Dateien erfolgt mit

Shift + linker Maustaste

Nach Auswahl der einzelnen Flächen können diese unter

File > Exportieren > Stl (.stl)

exportiert werden. Beim Exportieren ist darauf zu achten, dass im Exportmenü der Haken bei ASCII gesetzt wird.

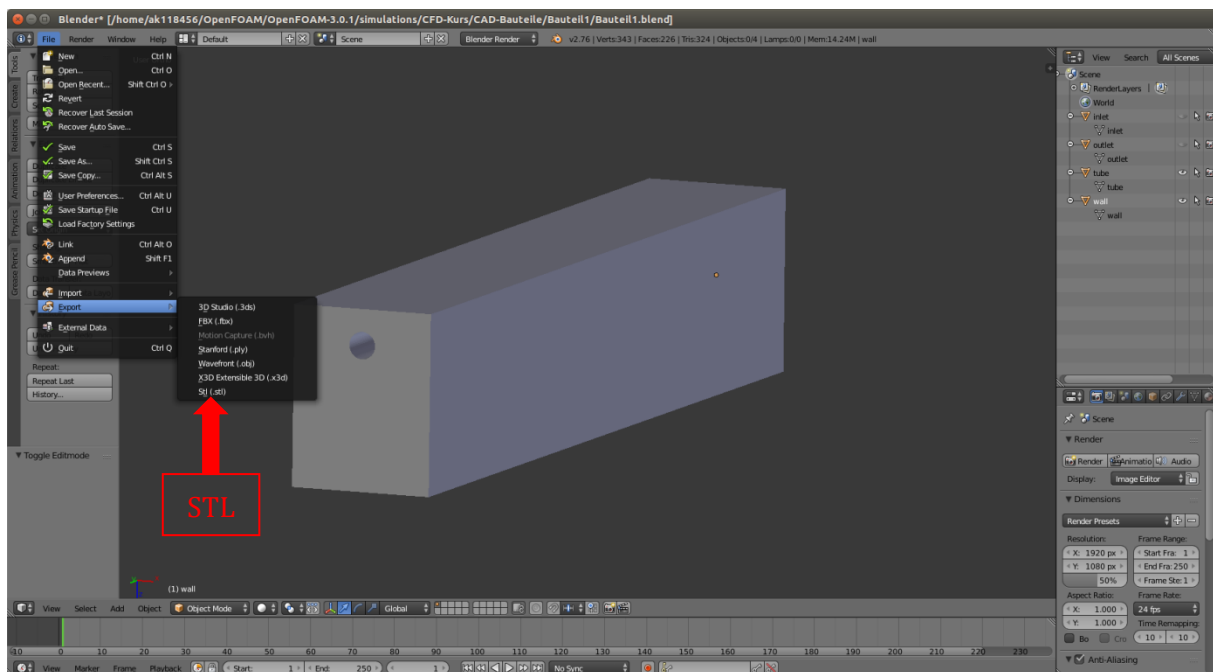


Abbildung 11: Blender Menüpunkt des stl-Exports

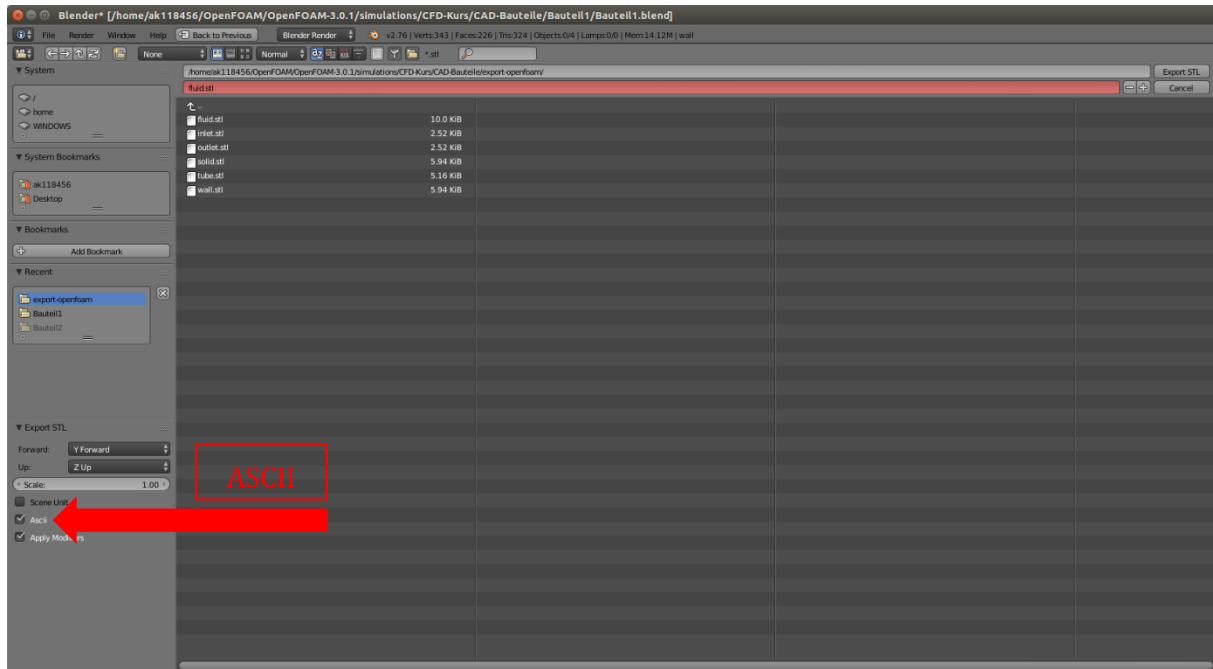


Abbildung 12: Blender stl-Exporteinstellungen, ASCII

3. Pre-Processing mit OpenFoam®

Um mit dem Pre-Processing in OpenFoam® beginnen zu können, müssen alle Geometrien separat vorhanden sein, d.h. es müssen „inlet.stl“, „outlet.stl“, „tube.stl“, „wall.stl“, „fluid.stl“ und „solid.stl“ vorliegen.

Wenn alle Geometriedateien vorhanden sind, kann mit dem Aufbau der Simulation begonnen werden. Der in diesem Simulationsbeispiel beschriebene Fall soll den Aufheiz- und Abkühlvorgang eines Stahlblocks zeigen, der von einem Medium durchströmt wird. Hierfür wird zunächst mittels „blockMesh“ und „snappyHexMesh“ die Vernetzung der Geometrie durchgeführt. Danach soll mithilfe des Löser „chtMultiRegionFoam“ die Temperaturverteilung dargestellt werden.

Simulationen, die mit OpenFoam® durchgeführt werden, besitzen oftmals folgende Standardstruktur:

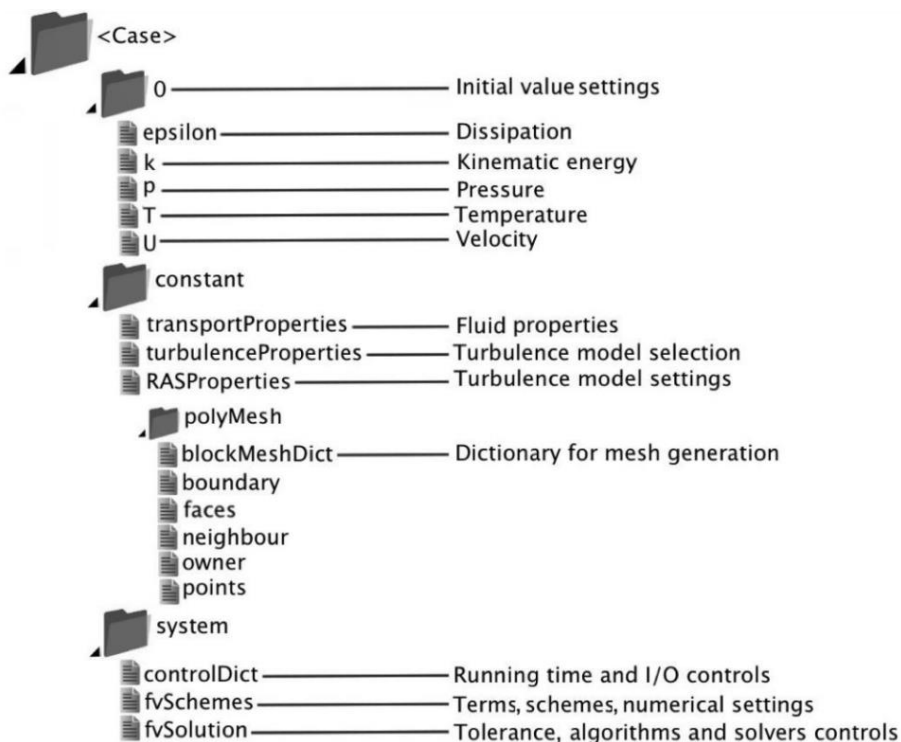


Abbildung 13: Ordnerstruktur der meisten OpenFoam® Simulationen

Wird mit dem Aufsetzen einer Simulation in OpenFoam® begonnen, empfiehlt es sich, einen bereits vorhandenen Simulationsfall zu kopieren und diesen anzupassen. Für die meisten Löser sind in der Standardinstallation von OpenFoam® Beispiele vorhanden, welche im Ordner „tutorials“ zu finden sind.

Das Simulationsbeispiel, das in den nachfolgenden Schritten genauer beschrieben wird, ist folgendermaßen gegliedert:

Name des Simulationsbeispiels:	cfdkurs
Speicherort des Simulationsbeispiels:	~/OpenFOAM/OpenFOAM-3.0.1/simulations/CFD-Kurs/cfdkurs\$

Im Speicherort des Simulationsbeispiels, auch Home-Verzeichnis genannt, werden in weiterer Folge alle Befehle im Terminal (Shell) ausgeführt. Muss also zum Beispiel das erste Rechengitter mittels „blockMesh“ erstellt werden, muss zuerst der Ordner „cfdkurs“ geöffnet und danach der Befehl >blockMesh eingegeben werden. Im Verzeichnis „cfdkurs“ befinden sich die Ordner „0“, „constant“ und „system“.

3.1. Gittererstellung allgemein

In OpenFoam® kann das Gitter mithilfe der „snappyHexMesh“-Funktion erstellt werden. Der Ablauf einer solchen Erstellung soll anhand eines einfachen Beispiels erklärt werden [9], [10].

- Schritt 1: Für den ersten Schritt wird eine .stl-Datei der Geometrie benötigt. In dieser mithilfe des CAD-Programms erstellten Datei wird die Oberfläche der Geometrie anhand von Dreiecken beschrieben.
- Schritt 2: Über diese Geometrie wird mittels „blockMesh“ ein erstes Hintergrundgitter gelegt. Mithilfe der „blockMesh“-Funktion kann ein Quader erstellt werden, welcher in mehrere kubische oder quaderförmige Zellen eingeteilt wird. In „blockMeshDict“, dem zum Ausführen von „blockMesh“ verwendeten Textfile, werden die Koordinaten der Eckpunkte des Quaders und die Anzahl der sich dazwischen befindenden Zellen festgelegt.
- Schritt 3: Die „snappyHexMesh“-Funktion verfeinert nun sämtliche die .stl-Oberfläche umgebenden Zellen.
Wie oft diese Zellen verfeinert werden sollen, wird in „snappyHexMeshDict“ festgelegt.
- Schritt 4: Anschließend werden entweder alle innerhalb oder alle außerhalb des Betrachtungsraums liegenden Zellen entfernt und die restlichen Zellen weiter verfeinert.
- Schritt 5: In einem letzten Schritt werden die Kanten und Flächen geglättet. Außerdem können zusätzliche Schichten (engl. „layers“) eingeführt werden, um den Übergang zwischen den Zellen sowie der Geometrieoberfläche zu verfeinern und so die Auflösung zu erhöhen.

Die Schritte 3 bis 5 werden alle in einem Vorgang während der Ausführung von „snappyHexMesh“ durchgeführt und im Ordner „system“ in „snappyHexMeshDict“ genau definiert.

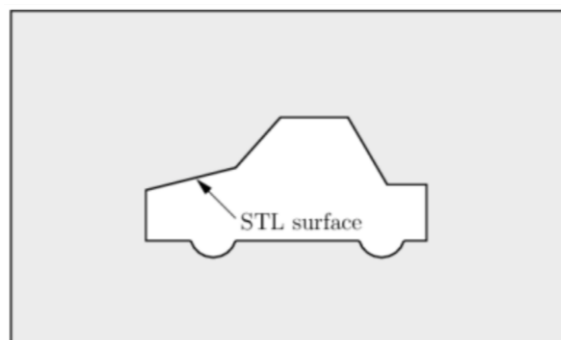


Abbildung 14: Schritt 1, vorhandene stl-Datei

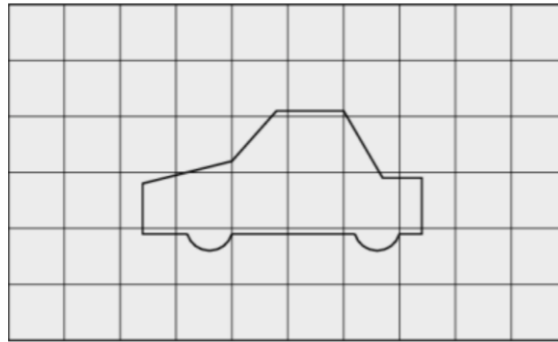


Abbildung 15: Schritt 2, Erzeugung eines „blockMesh“-Blocks (Hintergrundgitter)



Abbildung 16: Schritt 3, Verfeinerung des Gitternetzes mithilfe der „snappyHexMesh“-Funktion

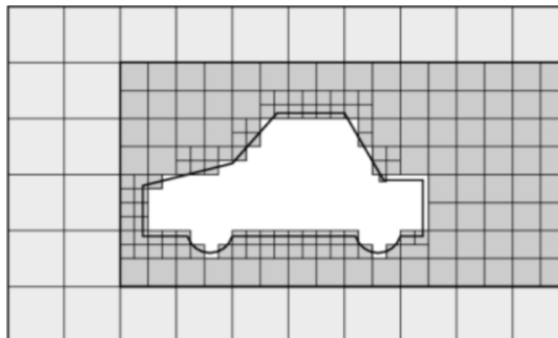


Abbildung 17: Schritt 4, Entfernung von innerhalb oder außerhalb des Betrachtungsraum liegenden Zellen

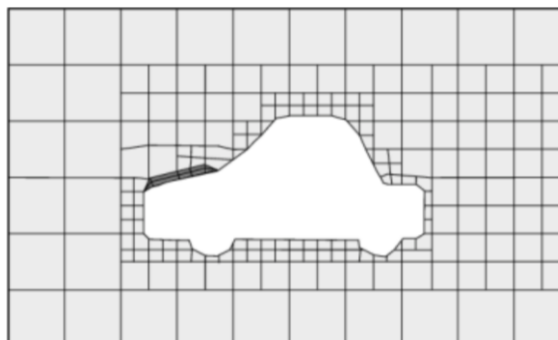


Abbildung 18: Schritt 5, Glättung und Verfeinerung des Gitternetzes

In den nachfolgenden Abschnitten wird das an dieser Stelle übersichtsmäßig dargestellte Vorgehen genauer beschrieben.

4. Erstes Rechengitter – „blockMesh“

Bevor mit dem Erstellen eines ersten Rechengitters begonnen wird, muss überprüft werden, ob die Geometrie die richtige Skalierung besitzt. Beim Erstellen der .stl-Datei bzw. beim Exportieren sollte darauf geachtet werden, mit welchen Einheiten die Geometrie exportiert wird. OpenFoam® arbeitet mit SI-Einheiten, daher werden alle Geometrieabmessungen in Metern übernommen. Sollte die Geometrie beispielsweise in Millimetern exportiert worden sein, kann eine Transformation in OpenFoam® vorgenommen werden. Dies kann mit dem Befehl >transformPoints durchgeführt werden:

```
>transformPoints -scale '(0.001 0.001 0.001)' -region fluid  
>transformPoints -scale '(0.001 0.001 0.001)' -region solid
```

Mithilfe von „blockMesh“ wird wie zuvor beschrieben ein Hintergrundgitter über die Geometrie gelegt. Zum Ausführen des Befehls „blockMesh“ werden das dazugehörige Textfile „blockMeshDict“ sowie das Textfile „controlDict“ benötigt. Beide werden im Ordner „system“ gespeichert und üblicherweise von anderen Simulationsvorlagen übernommen sowie anschließend an den jeweiligen neuen „case“ angepasst. Mit folgendem Befehl kann ein Textfile in einen neuen Ordner kopiert werden:

```
>cp Ort-wo-sich-die-Datei-befindet Zielordner/
```

Beispiel:

```
~/OpenFOAM/OpenFOAM-3.0.1/simulations/CFD-Kurs$ cp  
StandardVersuchBaseCaseVH/system/blockMeshDict system/
```

Im „blockMeshDict“-Textfile werden die Koordinaten der Eckpunkte des Quaders und die Anzahl der sich dazwischen befindenden Zellen festgelegt. Nachfolgende Abbildung 19 zeigt das „blockMeshDict“-File.

Beispiel:

~/OpenFOAM/OpenFOAM-3.0.1/simulations/CFD-Kurs/cfdkurs\$ nano foam.foam

Daraufhin öffnet sich der Texteditor und ein leeres Textfile erscheint. Mit

Strg + o

wird das File gespeichert und mit

Strg + x

wird der Editor wieder geschlossen. Daraufhin kann ParaView mit folgendem Befehl geöffnet werden:

>paraview

In ParaView müssen danach die stl-Geometriedaten und das Dummyfile „foam.foam“ geöffnet werden:

File > Open > foam.foam

Falls die Datei nicht sofort angezeigt wird, muss das Öffnen noch mit „apply“ bestätigt werden.

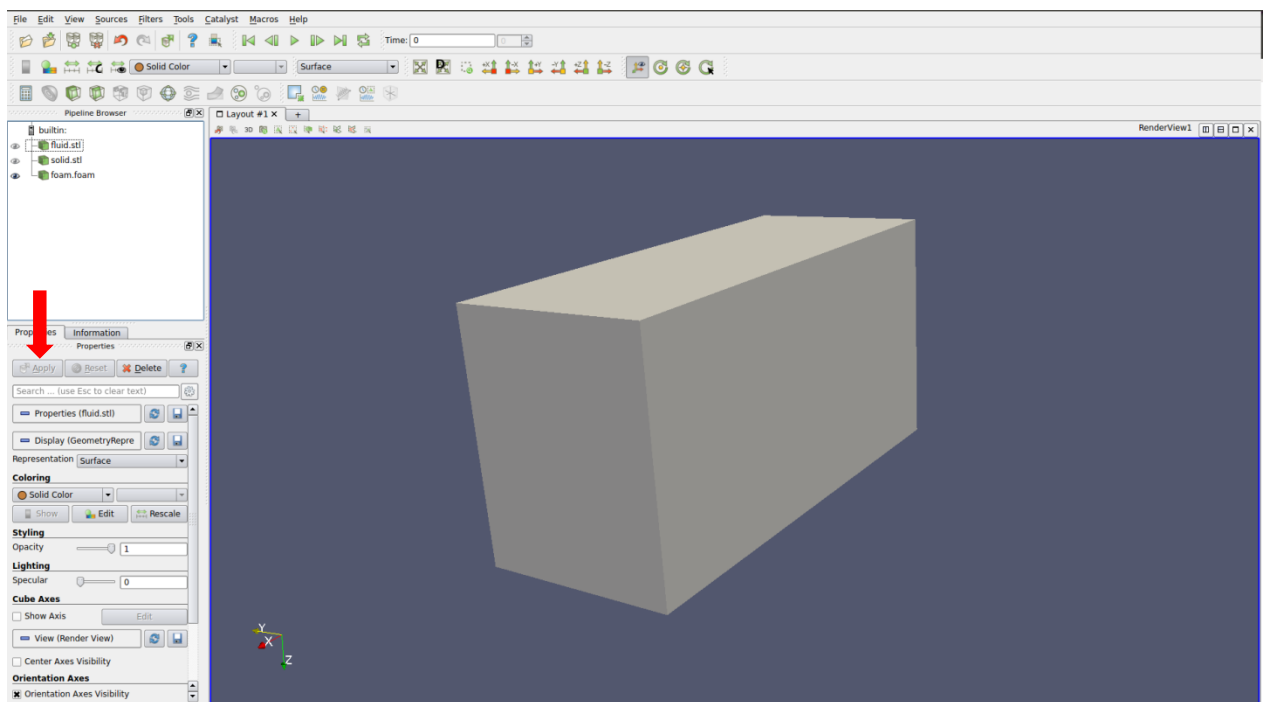


Abbildung 20: Erzeugter Block in ParaView

Im linken, unteren Menübereich werden unter „Informationen“ die Eckpunktkoordinaten der ausgewählten Geometrie angezeigt. In Abbildung 21 sind die Eckpunktkoordinaten der „solid.stl“-Geometrie veranschaulicht.

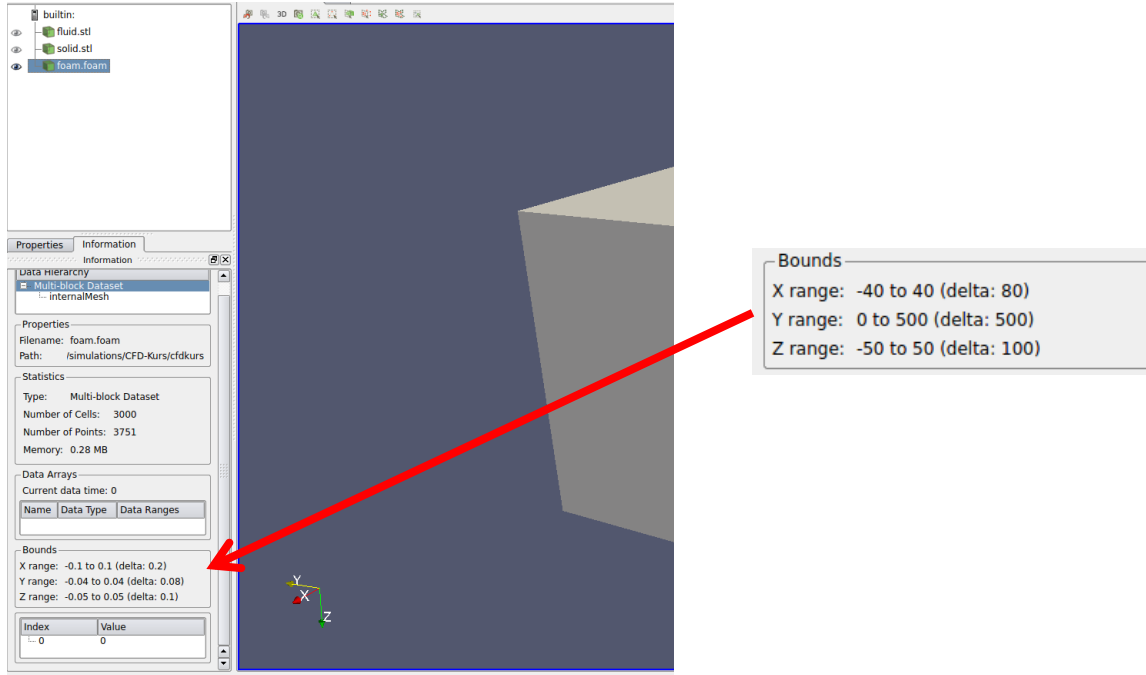


Abbildung 21: Eckpunktkoordinaten der „solid.stl“-Geometrie

Nun müssen die Koordinaten der Simulationsgeometrie, welche aus ParaView abgelesen werden, in das Textfile „blockMeshDict“ eingetragen werden. Dabei sollte der „blockMesh“-Block immer etwas größer als die Simulationsgeometrie gewählt werden. Im gezeigten Beispiel ist der blockMesh-Block um 5 Einheiten größer als die Simulationsgeometrie. Der erzeugte Block besitzt somit folgende Abmessungen:

```

19 vertices
20 (
21   (-45  -5  -55)           // 0
22   ( 45  -5  -55)           // 1
23   ( 45  505 -55)           // 2
24   (-45  505 -55)           // 3
25   (-45  -5   55)           // 4
26   ( 45  -5   55)           // 5
27   ( 45  505  55)           // 6
28   (-45  505  55)           // 7
29 );
    
```

Abbildung 22: Eckpunktkoordinaten des erzeugten „blockMesh“-Blocks, eingetragen im „blockMeshDict“-File

Nachdem die Eckpunktkoordinaten in das „blockMeshDict“-File eingetragen wurden, muss die Anzeige in ParaView mit „refresh“ noch aktualisiert werden. Daraufhin erscheint der zuvor definierte Block im Ansichtsbereich (Abbildung 23).

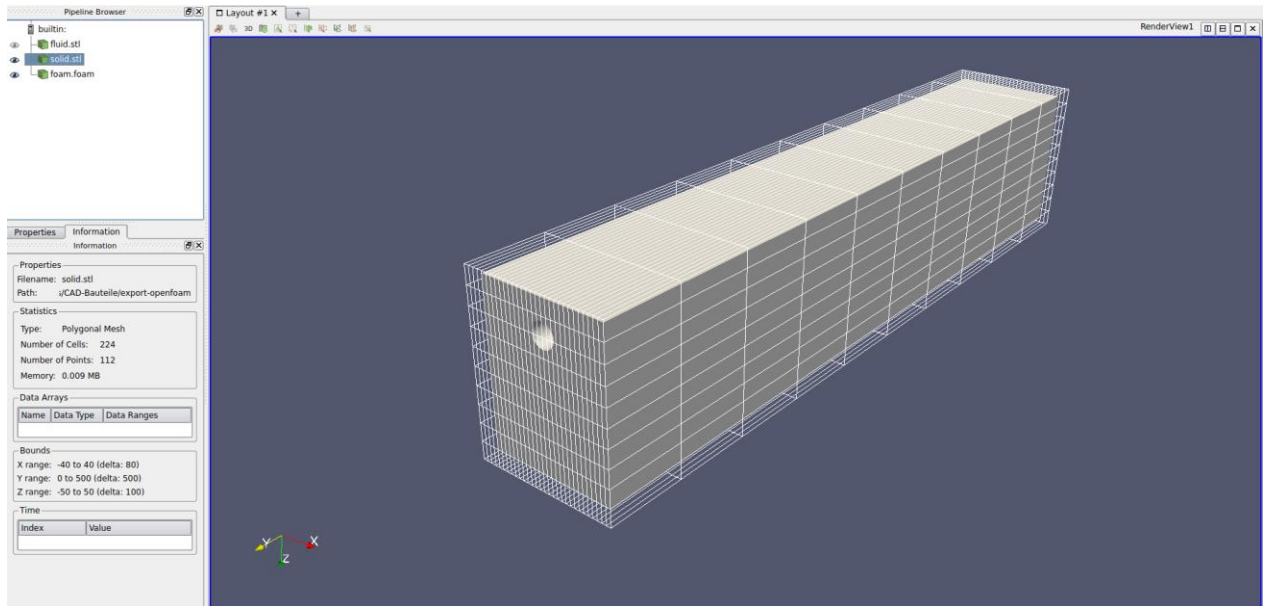


Abbildung 23: Simulationsgeometrie mit dem BlockMesh-Block (Gitter) darüber

Die Form der Rechtecke sollte optimalerweise würfelförmig sein. Dies kann durch Anpassen der Aufteilung im „blockMeshDict“-Textfile vorgenommen werden. Dazu muss das Textfile „blockMeshDict“ geöffnet werden und die Zeile 33 „(30 10 10) – Anzahl der Zellen zwischen den Eckpunkten“ geändert werden. Durch erneutes Ausführen des „blockMesh“-Befehls werden die eingestellten Daten übernommen:

>blockMesh

Beispiel:

~/OpenFOAM/OpenFOAM-3.0.1/simulations/CFD-Kurs/cfdkurs\$ blockMesh

5. Zweites Rechengitter – „snappyHexMesh“

Mit dem Programm „snappyHexMesh“ werden anschließend die Zellen verfeinert, welche die stl-Geometrie umschließen. Damit die Verfeinerung mittels „snappyHexMesh“ durchgeführt werden kann, müssen zunächst folgende Dateien in den Ordner „system“ kopiert werden:

- „system“
 - „snappyHexMeshDict“
 - „meshQualityDict“
 - „surfaceFeatureExtractDict“
 - „fvSchemes“
 - „fvSolution“

Des Weiteren müssen sich die stl.-Dateien im Ordner „constant“ befinden:

- „constant“
 - Ordner „triSurface“ (muss gegebenenfalls erstellt werden)
 - „solid.stl“
 - „fluid.stl“

Falls der Ordner „triSurface“ nicht vorhanden ist, kann dieser ohne Weiteres erstellt werden. Möglich ist dies über folgenden Befehl:

```
>mkdir triSurface
```

Beispiel:

```
~/OpenFOAM/OpenFOAM-3.0.1/simulations/CFD-Kurs/cfdkurs/constant$ mkdir triSurface
```

Anschließend kann mit der Anpassung der Textdateien begonnen werden. Der Inhalt der Textdatei „snappyHexMeshDict“ ist in den folgenden Abbildungen dargestellt:


```

233 // Of limited use in this case since faceZone faces not handled.
234 nFeatureSnapIter 10;
235 }
236
237
238
239 // Settings for the layer addition.
240 addLayersControls
241 {
242     relativeSizes true;
243
244     // Per final patch (so not geometry!) the layer information
245     layers
246     {
247         {
248             maxY
249             {
250                 nSurfaceLayers 3;
251             }
252         }
253     }
254     // Expansion factor for layer mesh
255     expansionRatio 1.3;
256
257     // Wanted thickness of final added cell layer. If multiple layers
258     // is the thickness of the layer furthest away from the wall.
259     // Relative to undistorted size of cell outside layer.
260     // See relativeSizes parameter.
261     finalLayerThickness 1;
262
263     // Minimum thickness of cell layer. If for any reason layer
264     // cannot be above minThickness do not add layer.
265     // Relative to undistorted size of cell outside layer.
266     minThickness 0.1;
267
268     // If points get not extruded do nGrow layers of connected faces that are
269     // also not grown. This helps convergence of the layer addition process
270     // close to features.
271     // Note: changed(corrected) w.r.t 17x! (didn't do anything in 17x)
272     nGrow 0;
273
274     // Advanced settings
275     // When not to extrude surface. 0 is flat surface, 90 is when two faces
276     // are perpendicular
277     featureAngle 30;
278
279     // Maximum number of snapping relaxation iterations. Should stop
280     // before upon reaching a correct mesh.
281     nRelaxIter 3;
282
283     // Number of smoothing iterations of surface normals
284     nSmoothSurfaceNormals 1;
285
286     // Number of smoothing iterations of interior mesh movement direction
287     nSmoothNormals 3;
288
289     // Smooth layer thickness over surface patches
290     nSmoothThickness 2;
291
292     // Stop layer growth on highly warped cells
293     maxFaceThicknessRatio 0.5;
294
295     // Reduce layer growth where ratio thickness to medial
296     // distance is large
297     maxThicknessToMedialRatio 1;
298
299     // Angle used to pick up medial axis points
300     // Note: changed(corrected) w.r.t 17x! 90 degrees corresponds to 130 in 17x.
301     minMedialAxisAngle 90;
302
303     // Create buffer region for new layer terminations
304     nBufferCellsNoExtrude 0;
305
306     // Overall max number of layer addition iterations. The mesher will exit
307     // if it reaches this number of iterations; possibly with an illegal
308     // mesh.
309     nLayerIter 50;
310 }
311
312 // Generic mesh quality settings. At any undoable phase these determine
313 // where to undo.
314 meshQualityControls
315 {
316     #include "meshQualityDict"
317     // Advanced
318     // Number of error distribution iterations
319     nSmoothScale 4;
320     // Amount to scale back displacement at error points
321     errorReduction 0.75;
322 }
323
324 // Advanced
325 // Merge tolerance. Is fraction of overall bounding box of initial mesh.
326 // Note: the write tolerance needs to be higher than this.
327 mergeTolerance 1e-6;
328
329 // *****
330
331 // *****
332
333 // *****
334
335 // *****
336
337 // *****
338
339 // *****
340
341 // *****
342
343 // *****
344
345 // *****
346
347 // *****
348
349 // *****
350
351 // *****
352
353 // *****
354
355 // *****
356
357 // *****
358
359 // *****
360
361 // *****
362
363 // *****
364
365 // *****
366
367 // *****
368
369 // *****
370
371 // *****
372
373 // *****
374
375 // *****
376
377 // *****
378
379 // *****
380
381 // *****
382
383 // *****
384
385 // *****
386
387 // *****
388
389 // *****
390
391 // *****
392
393 // *****
394
395 // *****
396
397 // *****
398
399 // *****
400
401 // *****
402
403 // *****
404
405 // *****
406
407 // *****
408
409 // *****
410
411 // *****
412
413 // *****
414
415 // *****
416
417 // *****
418
419 // *****
420
421 // *****
422
423 // *****
424
425 // *****
426
427 // *****
428
429 // *****
430
431 // *****
432
433 // *****
434
435 // *****
436
437 // *****
438
439 // *****
440
441 // *****
442
443 // *****
444
445 // *****
446
447 // *****
448
449 // *****
450
451 // *****
452
453 // *****
454
455 // *****
456
457 // *****
458
459 // *****
460
461 // *****
462
463 // *****
464
465 // *****
466
467 // *****
468
469 // *****
470
471 // *****
472
473 // *****
474
475 // *****
476
477 // *****
478
479 // *****
480
481 // *****
482
483 // *****
484
485 // *****
486
487 // *****
488
489 // *****
490
491 // *****
492
493 // *****
494
495 // *****
496
497 // *****
498
499 // *****
500
501 // *****
502
503 // *****
504
505 // *****
506
507 // *****
508
509 // *****
510
511 // *****
512
513 // *****
514
515 // *****
516
517 // *****
518
519 // *****
520
521 // *****
522
523 // *****
524
525 // *****
526
527 // *****
528
529 // *****
530
531 // *****
532
533 // *****
534
535 // *****
536
537 // *****
538
539 // *****
540
541 // *****
542
543 // *****
544
545 // *****
546
547 // *****
548
549 // *****
550
551 // *****
552
553 // *****
554
555 // *****
556
557 // *****
558
559 // *****
560
561 // *****
562
563 // *****
564
565 // *****
566
567 // *****
568
569 // *****
570
571 // *****
572
573 // *****
574
575 // *****
576
577 // *****
578
579 // *****
580
581 // *****
582
583 // *****
584
585 // *****
586
587 // *****
588
589 // *****
590
591 // *****
592
593 // *****
594
595 // *****
596
597 // *****
598
599 // *****
600
601 // *****
602
603 // *****
604
605 // *****
606
607 // *****
608
609 // *****
610
611 // *****
612
613 // *****
614
615 // *****
616
617 // *****
618
619 // *****
620
621 // *****
622
623 // *****
624
625 // *****
626
627 // *****
628
629 // *****
630
631 // *****
632
633 // *****
634
635 // *****
636
637 // *****
638
639 // *****
640
641 // *****
642
643 // *****
644
645 // *****
646
647 // *****
648
649 // *****
650
651 // *****
652
653 // *****
654
655 // *****
656
657 // *****
658
659 // *****
660
661 // *****
662
663 // *****
664
665 // *****
666
667 // *****
668
669 // *****
670
671 // *****
672
673 // *****
674
675 // *****
676
677 // *****
678
679 // *****
680
681 // *****
682
683 // *****
684
685 // *****
686
687 // *****
688
689 // *****
690
691 // *****
692
693 // *****
694
695 // *****
696
697 // *****
698
699 // *****
700
701 // *****
702
703 // *****
704
705 // *****
706
707 // *****
708
709 // *****
710
711 // *****
712
713 // *****
714
715 // *****
716
717 // *****
718
719 // *****
720
721 // *****
722
723 // *****
724
725 // *****
726
727 // *****
728
729 // *****
730
731 // *****
732
733 // *****
734
735 // *****
736
737 // *****
738
739 // *****
740
741 // *****
742
743 // *****
744
745 // *****
746
747 // *****
748
749 // *****
750
751 // *****
752
753 // *****
754
755 // *****
756
757 // *****
758
759 // *****
760
761 // *****
762
763 // *****
764
765 // *****
766
767 // *****
768
769 // *****
770
771 // *****
772
773 // *****
774
775 // *****
776
777 // *****
778
779 // *****
780
781 // *****
782
783 // *****
784
785 // *****
786
787 // *****
788
789 // *****
790
791 // *****
792
793 // *****
794
795 // *****
796
797 // *****
798
799 // *****
800
801 // *****
802
803 // *****
804
805 // *****
806
807 // *****
808
809 // *****
810
811 // *****
812
813 // *****
814
815 // *****
816
817 // *****
818
819 // *****
820
821 // *****
822
823 // *****
824
825 // *****
826
827 // *****
828
829 // *****
830
831 // *****
832
833 // *****
834
835 // *****
836
837 // *****
838
839 // *****
840
841 // *****
842
843 // *****
844
845 // *****
846
847 // *****
848
849 // *****
850
851 // *****
852
853 // *****
854
855 // *****
856
857 // *****
858
859 // *****
860
861 // *****
862
863 // *****
864
865 // *****
866
867 // *****
868
869 // *****
870
871 // *****
872
873 // *****
874
875 // *****
876
877 // *****
878
879 // *****
880
881 // *****
882
883 // *****
884
885 // *****
886
887 // *****
888
889 // *****
890
891 // *****
892
893 // *****
894
895 // *****
896
897 // *****
898
899 // *****
900
901 // *****
902
903 // *****
904
905 // *****
906
907 // *****
908
909 // *****
910
911 // *****
912
913 // *****
914
915 // *****
916
917 // *****
918
919 // *****
920
921 // *****
922
923 // *****
924
925 // *****
926
927 // *****
928
929 // *****
930
931 // *****
932
933 // *****
934
935 // *****
936
937 // *****
938
939 // *****
940
941 // *****
942
943 // *****
944
945 // *****
946
947 // *****
948
949 // *****
950
951 // *****
952
953 // *****
954
955 // *****
956
957 // *****
958
959 // *****
960
961 // *****
962
963 // *****
964
965 // *****
966
967 // *****
968
969 // *****
970
971 // *****
972
973 // *****
974
975 // *****
976
977 // *****
978
979 // *****
980
981 // *****
982
983 // *****
984
985 // *****
986
987 // *****
988
989 // *****
990
991 // *****
992
993 // *****
994
995 // *****
996
997 // *****
998
999 // *****
1000 // *****

```

Abbildung 25: Inhalt „snappyHexMeshDict“-Datei, Zeile 233-336

Darin müssen zunächst die Geometriedateien angepasst werden. Hierbei müssen unter „geometry“ (Zeile 30) die verwendeten .stl-Geometrien angegeben werden. Im durchgeführten Beispiel zählen dazu folgende .stl-Dateien:

- „solid.stl“
- „fluid.stl“

Die restlichen Einträge unter „geometry“ können gelöscht werden. Des Weiteren müssen die Einträge bei „features“ (Zeile 103) analog zur Änderung bei „geometry“ angepasst werden.

```

geometry
{
    solid.stl
    {
        type triSurfaceMesh;
        name solid;
    }
    fluid.stl
    {
        type triSurfaceMesh;
        name fluid;
    }
};

features
(
    {
        file "solid.eMesh";
        level 1;
    }
    {
        file "fluid.eMesh";
        level 1;
    }
);

```

Im Abschnitt „refinementSurfaces“ (Zeile 138) müssen ebenfalls Anpassungen vorgenommen werden. Dabei sind die Namen wiederum in „solid“ und „fluid“ umzubenennen. In den Einträgen „faceZone“ und „cellZone“ sind die Namen der stl-Dateien einzutragen. Darüber hinaus müssen Koordinaten eines Punktes angegeben werden, der sich im Inneren der zu vernetzenden Geometrie befindet. Dies ist für „solid“ und „fluid“ einzutragen. In der Zeile 201 befindet sich der Eintrag „locationInMesh“, wo wiederum ein Punkt im Inneren der zu vernetzenden Geometrie anzugeben

ist. Die Angabe eines Punktes im Inneren einer Geometrie, dient der Definition des Vernetzungsstarts bzw. ob die innere oder die äußere Geometrie vernetzt werden soll.

In der nachfolgenden Abbildung 26 ist der Inhalt der „meshQualityDict“-Datei dargestellt. Diese wird ebenfalls für die Simulation auch benötigt, es sind jedoch keine Änderungen erforderlich.

```

1 /*----- C++ -----*/
2 =====
3 \ \ \ \ \ Field | OpenFOAM: The Open Source CFD Toolbox
4 \ \ \ \ \ Operation | Version: 3.0.1
5 \ \ \ \ \ And | Web: www.OpenFOAM.org
6 \ \ \ \ \ Manipulation |
7 /*-----*/
8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        dictionary;
13     object       meshQualityDict;
14 }
15 // *****
16
17 // Include defaults parameters from master dictionary
18 #include "SWM_PROJECT_DIR/etc/caseDicts/meshQualityDict"
19
20 // #includeEtc "caseDicts/meshQualityDict"
21
22
23 // *****
    
```

Abbildung 26: Inhalt „meshQualityDict“-Datei

Im Textfile „surfaceFeatureExtractDict“ (Abbildung 27) sind wiederum die Namen der stl-Dateien einzutragen (Zeile 17 und 33).

```

1 /*----- C++ -----*/
2 =====
3 \ \ \ \ \ Field | OpenFOAM: The Open Source CFD Toolbox
4 \ \ \ \ \ Operation | Version: 3.0.1
5 \ \ \ \ \ And | Web: www.OpenFOAM.org
6 \ \ \ \ \ Manipulation |
7 /*-----*/
8 FoamFile
9 {
10     version      2.0;
11     format       ascii;
12     class        dictionary;
13     object       surfaceFeatureExtractDict;
14 }
15 // *****
16
17 solid.stl
18 {
19     extractionMethod    extractFromSurface;
20
21     extractFromSurfaceCoeffs
22     {
23         // Mark edges whose adjacent surface normals are at an angle less
24         // than includedAngle as features
25         // - 0 : selects no edges
26         // - 180: selects all edges
27         includedAngle    150;
28     }
29
30     // Write options
31     writeFeatureEdgeMesh    yes;
32 }
33 fluid.stl
34 {
35     extractionMethod    extractFromSurface;
36
37     extractFromSurfaceCoeffs
38     {
39         // Mark edges whose adjacent surface normals are at an angle less
40         // than includedAngle as features
41         // - 0 : selects no edges
42         // - 180: selects all edges
43         includedAngle    150;
44     }
45
46     // Write options
47     writeFeatureEdgeMesh    yes;
48 }
49 // *****
    
```

Abbildung 27: Inhalt „surfaceFeatureExtractDict“-File

Nachdem alle Textdateien angepasst wurden, muss noch der Befehl

>surfaceFeatureExtract

Beispiel:

~/OpenFOAM/OpenFOAM-3.0.1/simulations/CFD-Kurs/cfdkurs\$ surfaceFeatureExtract

ausgeführt werden. Damit werden die Dateien

- „solid.eMesh“
- „fluid.eMesh“

im Ordner „triSurface“ generiert. Danach kann der Befehl

>snappyHexMesh -overwrite

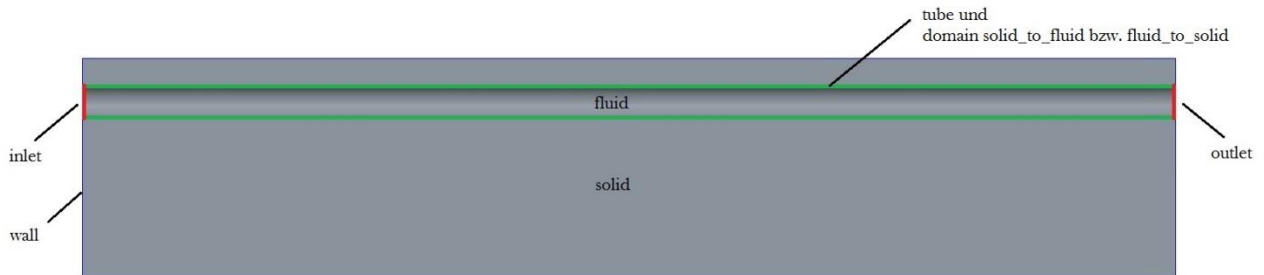
Beispiel:

~/OpenFOAM/OpenFOAM-3.0.1/simulations/CFD-Kurs/cfdkurs\$ snappyHexMesh -overwrite

ausgeführt werden. Der Befehl „overwrite“ wird verwendet, da ansonsten ein neuer Ordner erstellt wird, falls bereits ein Gitter („mesh“) existiert.

6. Erstellung und Zuordnung der Regionen

Da in diesem Beispiel der Wärmeübergang von einem flüssigen zu einem festen Stoff simuliert wird, müssen die einzelnen Geometrien eingeteilt werden. Dies dient dazu, dem Programm mitzuteilen, um welche Geometrie bzw. um welchen Volumenkörper es sich bei einem Feststoff oder einer Flüssigkeit handelt. Zudem sollen den jeweiligen Stoffen im Anschluss angepasste Materialparameter zugeteilt



Mit Blender wurden in einem vorherigen Schritt die Geometrien einzeln als .stl-Files exportiert. Dabei setzt sich die Geometrie „solid.stl“ aus „wall.stl“ und „tube.stl“ zusammen. Die Geometrie „fluid.stl“ setzt sich aus „tube.stl“, „inlet.stl“ und „outlet.stl“ zusammen. Somit müssen insgesamt 7 einzelne .stl-Dateien vorhanden sein. Je nach Simulationssetup kann es sein, dass die Geometrie „tube.stl“ nicht benötigt wird. In Abbildung 28 sind die einzelnen stl-Geometrien bildlich dargestellt.

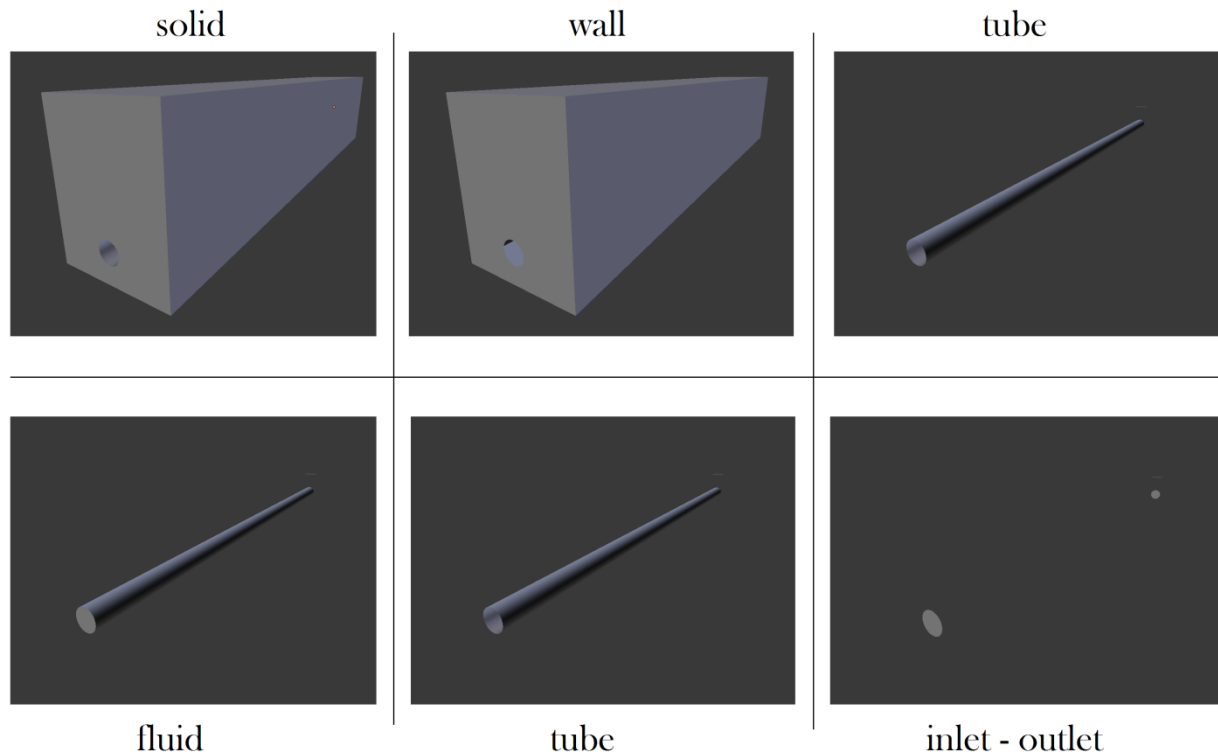


Abbildung 28: Veranschaulichung der einzelnen stl-Dateien

Die einzelnen stl-Dateien müssen bestimmten Regionen zugeordnet werden. Eine Auflistung der stl-Dateien mit den dazugehörigen Regionen ist in Tabelle 2 dargestellt.

Tabelle 2: Einteilung der stl-Dateien in Regionen

stl-Datei	Region
„tube.stl“	„fluid“
„inlet.stl“	„fluid“
„outlet.stl“	„fluid“
„wall.stl“	„solid“

Bevor diese zugeteilt werden, müssen zunächst die Zonen („fluid“ und „solid“) aufgeteilt werden. Dies erfolgt mit folgendem Befehl:

```
>splitMeshRegions -cellZones -overwrite
```

Beispiel:

```
~/OpenFOAM/OpenFOAM-3.0.1/simulations/CFD-Kurs/cfdkurs$ splitMeshRegions -cellZones -overwrite
```

Danach müssen die einzelnen stl-Dateien bzw. Flächen den Regionen zugeordnet werden. Dies erfolgt mit dem Befehl:

```
>surfaceToPatch „Ort der stl-Datei“ -region „fluid oder solid“
```

Beispiel:

```
~/OpenFOAM/OpenFOAM-3.0.1/simulations/CFD-Kurs/cfdkurs$ surfaceToPatch constant/triSurface/inlet.stl -region fluid
```

```
~/OpenFOAM/OpenFOAM-3.0.1/simulations/CFD-Kurs/cfdkurs$ surfaceToPatch constant/triSurface/outlet.stl -region fluid
```

```
~/OpenFOAM/OpenFOAM-3.0.1/simulations/CFD-Kurs/cfdkurs$ surfaceToPatch constant/triSurface/tube.stl -region fluid
```

```
~/OpenFOAM/OpenFOAM-3.0.1/simulations/CFD-Kurs/cfdkurs$ surfaceToPatch constant/triSurface/wall.stl -region solid
```

Dabei kann es vorkommen, dass folgende Fehlermeldung erscheint (Abbildung 29):

```
-->FOAM FATAL ERROR:
Wrong number of arguments, expected 1 found 2
Invalid options: -region
```

```

ak118456@ak118456-HP-ElliteBook-8570p:~/OpenFOAM/OpenFOAM-3.0.1/simulations/CFD-Kurs/cfdkurs$ surfaceToPatch constant/triSurface/inlet.stl -region fluid
Usage: surfaceToPatch [OPTIONS] <surfaceFile>
options:
  -case <dir>      specify alternate case directory, default is the cwd
  -faceSet <name>  only repatch the faces in specified faceSet
  -noFunctionObjects
                  do not execute functionObjects
  -tol <scalar>    search tolerance as fraction of mesh size (default 1e-3)
  -srcDoc          display source code in browser
  -doc            display application documentation in browser
  -help           print the usage

reads surface and applies surface regioning to a mesh

Using: OpenFOAM-3.0.1 (see www.OpenFOAM.org)
Build: 3.0.1-bc1922f074df

--> FOAM FATAL ERROR:
Wrong number of arguments, expected 1 found 2
Invalid option: -region

FOAM exiting

```

Abbildung 29: Fehlermeldung bei Eingabe des Befehls „surfaceToPatch constant/triSurface/inlet.stl -region fluid“

In diesem Fall muss ein zusätzlicher Schritt durchgeführt werden bzw. sind die Regionen separat zuzuordnen. Dies bedeutet, dass der „surfaceToPatch“-Befehl in einem separat angelegten Ordner ausgeführt werden muss. Dies dient dazu, die Regionen separat voneinander zuordnen („patchen“) zu können. Aus diesem Grund werden Dummy-Ordnerstrukturen angelegt („fluid > constant > polyMesh“), da OpenFoam® die Ordnerstruktur ansonsten nicht erkennt. Würde lediglich eine Region vorhanden sein, also nur „solid“ oder „fluid“, könnte der „surfaceToPatch“-Befehl direkt im Hauptordner (im Beispiel „cfdkurs“) ausgeführt werden.

Da im vorliegenden Beispiel jedoch zwei Regionen vorhanden sind, müssen folgende Schritte ausgeführt werden:

1. Zuerst müssen im Hauptordner die Ordner „solid“ und „fluid“ erstellt werden.
2. In diesen Ordnern muss dann jeweils ein „constant“-Ordner angelegt werden. Ein Ordner wird mit dem Befehl **>mkdir Ordnername** angelegt.
3. In diesen „constant“-Ordner muss danach der „polyMesh“-Ordner vom Hauptordner (~ / OpenFoam / OpenFoam-3.0.1 / simulations / CFD-Kurs / cfdkurs / constant) kopiert werden.
4. Des Weiteren muss im jeweiligen Ordner („solid“ und „fluid“) der Ordner „system“ angelegt werden und in diesen die Datei „controlDict“ kopiert werden. Diese ist im Hauptordner („cfdkurs“) im „system“-Ordner zu finden.
5. Abschließend müssen noch die dazugehörigen .stl-Dateien in die erstellten Ordner „fluid“ und „solid“ kopiert werden.

Nach dem Anlegen und Kopieren der Dateien muss die Ordnerstruktur wie folgt aussehen:

„cfdkurs“

- „constant“
 - „polyMesh“
 - „fluid“
 - „polyMesh“
 - „solid“
 - „polyMesh“
- „solid“
 - „constant“
 - „polyMesh“

- „system“
 - „controlDict“
- „wall.stl“
- **„fluid“**
 - „constant“
 - „polyMesh“
 - „system“
 - „controlDict“
 - „inlet.stl“
 - „outlet.stl“

Im Verzeichnis „constant > solid“ dürfen nur die .stl-Dateien der äußeren Wände enthalten sein. Das heißt zum Beispiel, die Wand eines inneren Rohres darf nicht enthalten sein. Nachdem die Ordnerstruktur angelegt worden ist, muss zum angelegten Ordner navigiert werden („solid“ oder „fluid“) und für jede enthaltene .stl-Datei der „surfaceToPatch“-Befehl ausgeführt werden:

>surfaceToPatch CAD-Datei.stl

Beispiel:

~/OpenFOAM/OpenFOAM-3.0.1/simulations/CFD-Kurs/cfdkurs/solid\$ surfaceToPatch wall.stl

~/OpenFOAM/OpenFOAM-3.0.1/simulations/CFD-Kurs/cfdkurs/fluid\$ surfaceToPatch inlet.stl

etc.

Zwischen dem Ausführen dieses Befehls für jede einzelne .stl-Datei muss der „patch0“-Eintrag in der „polyMesh-boundary“-Datei geändert werden. Nachfolgend sind die einzelnen Vorgänge Schritt für Schritt dargestellt:

Beispiel für „fluid“:

„cfdkurs > fluid > 0.1“

- Ersten Zeitschrittordner (0.1) öffnen (nicht 0)
- „polyMesh“-Ordner öffnen
- „boundary“-File öffnen
- „patch0“ in „inlet“ umbenennen (bzw. mit dem Namen versehen, der bei „surfaceToPatch“ verwendet wurde)
- >surfaceToPatch outlet.stl im Ordner „cfdkurs > fluid“ ausführen
- „cfdkurs > fluid > 0.2“ öffnen
- „polyMesh“-Ordner kopieren und in „cfdkurs > constant > fluid“ einfügen
- Danach kann der gesamte Ordner „cfdkurs > fluid“ als Sicherung abgelegt oder gelöscht werden

Danach müssen noch folgende Einträge im „boundary“-File gelöscht werden und die Zahl in Reihe 18 (Abbildung 30) von 4 auf 3 geändert werden, da sich die Anzahl der angegebenen Patches geändert hat. Wird dies nicht geändert, führt dies zu einer Fehlermeldung beim späteren Anzeigen

der Ergebnisse in ParaView. Die Fehlerfreiheit des erzeugten Gitters kann mit folgenden Befehl (beispielsweise für die Region „solid“) überprüft werden:

> checkMesh -region solid

Beispiel:

~/OpenFOAM/OpenFOAM-3.0.1/simulations/CFD-Kurs/cfdkurs\$ checkMesh -region solid

```
1 /*-----* C++ */
2
3 //
4 //   F i e l d   |   OpenFOAM: The Open Source CFD Toolbox
5 //   O p e r a t i o n   |   Version: 3.0.1
6 //   A n d   |   Web: www.OpenFOAM.org
7 //   M a n i p u l a t i o n
8
9 FoamFile
10 {
11     version      2.0;
12     format       ascii;
13     class        polyBoundaryMesh;
14     location     "0.02/polyMesh";
15     object       boundary;
16 }
17 // *****
18
19 fluid_to_solid
20 {
21     type          mappedWall;
22     inGroups
23 {
24     2
25     wall
26     mappedPatch
27     mappedPatch
28 )
29 ;
30     nFaces        4800;
31     startFace     16768;
32     sampleMode    nearestPatchFace;
33     sampleRegion  solid;
34     samplePatch   solid_to_fluid;
35 }
36 fluid_to_domain0
37 {
38     type          mappedWall;
39     inGroups
40 {
41     2
42     wall
43     mappedPatch
44 )
45 ;
46     nFaces        0;
47     startFace     21568;
48     sampleMode    nearestPatchFace;
49     sampleRegion  domain0;
50     samplePatch   domain0_to_fluid;
51 }
52 inlet
53 {
54     type          patch;
55     nFaces        32;
56     startFace     21568;
57 }
58 outlet
```

Abbildung 30: boundary-File des „fluids“

Um den Regionen nun ihre Eigenschaften und Anfangsbedingungen zuweisen zu können, wird in „constant“ ein „regionProperties“-File benötigt. Darin müssen folgende Einträge vorhanden sein:

7. Solving – Simulation mit „chtMultiRegion“

Bei dem ausgewählten Simulationsfall wird der Aufheiz- und Abkühlvorgang betrachtet. Beim Aufheizvorgang strömt Wasser mit einer Temperatur von 90°C 10 Sekunden lang durch das Rohr. Darauf folgend beginnt der Abkühlvorgang, bei dem Wasser mit einer Temperatur von 30°C 10 Sekunden lang durch das Rohr strömt. Zusätzlich dazu werden die Simulationen mit zwei verschiedenen Reynoldszahlen bzw. Strömungsgeschwindigkeiten durchgeführt, um den Wärmeübergang bei einer laminaren und turbulenten Strömung betrachten zu können.

7.1. Theorie Strömungssimulation

Für die Simulation der Strömung im Kanal wurde das Standard-k-ε-Turbulenzmodell verwendet. Es handelt sich um ein Zwei-Gleichungsmodell, das als meistverwendetes Modell zur Beschreibung von Strömungen in CFD-Simulationen gilt. Dabei werden in der Transportgleichung die Produktion und die Diffusion für turbulente kinetische Energie „k“ angenähert.

$$P_k = -(2\nu_T \bar{S}_{ij}) \bar{S}_{ij} = -2\nu_T (\bar{S}_{ij})^2 \quad (1)$$

$$D_k = \frac{\partial}{\partial x_j} \left(\nu \frac{\partial k}{\partial x_j} + \frac{\nu_T}{\sigma_k} \frac{\partial k}{\partial x_j} \right) = \left[\left(\nu + \frac{\nu_T}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] \quad (2)$$

Durch verschiedene Annahmen (z.B. Druck-Geschwindigkeitskorrelation) und Approximationen (z.B. Boussinesq-Ansatz) lauten die Gleichungen des Standard-k-ε-Modells schließlich [12]:

$$\frac{\partial}{\partial x_j} (\bar{u}_j k) = -\nu_T (\bar{S}_{ij})^2 - \varepsilon + \left[\left(\nu + \frac{\nu_T}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] \quad (3)$$

$$\frac{\partial}{\partial x_j} (\bar{u}_j \varepsilon) = \left\{ c_{\varepsilon 1} \frac{\varepsilon}{k} \right\} \left[-\nu_T (\bar{S}_{ij})^2 \right] - \left\{ c_{\varepsilon 2} \frac{\varepsilon}{k} \right\} \varepsilon + \left[\left(\nu + \frac{\nu_T}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_j} \right] \quad (4)$$

P_k	Produktion
D_k	Diffusion
ν_T	Wirbelviskosität
\bar{S}_{ij}	viskoser Spannungstensor
x_j	Ortskoordinate
\bar{u}_j	mittlere Geschwindigkeit
k	turbulente kinetische Energie
ε	isotrope Dissipationsrate
C_μ	Modellkonstante
$c_{\varepsilon 1}$	Modellkonstante
$c_{\varepsilon 2}$	Modellkonstante
σ_k	Modellkonstante
σ_ε	Modellkonstante

Typischerweise werden die Modellkonstanten folgendermaßen definiert: $C_\mu = 0,09$; $c_{\varepsilon 1} = 1,44$; $c_{\varepsilon 2} = 1,92$; $\sigma_k = 1,0$ und $\sigma_\varepsilon = 1,3$.

Um yPlus berechnen zu können werden turbulente Anfangswerte benötigt, welche wie folgt berechnet werden [13]:

$$k = \frac{3}{2} (I * |u_{ref}|)^2$$

dabei ist k die turbulente kinetische Energie und I die turbulente Intensität, welche mit 0,8 angenommen wird. u_{ref} ist die Geschwindigkeit,

$$\varepsilon = \frac{C_\mu^{0,75} * k^{1,5}}{L}$$

ε ist die turbulente Dissipationsrate, C_μ eine Konstante mit dem Wert 0.09 und L die Länge des Kanals.

$$v_t = C_\mu \frac{k^2}{\varepsilon}$$

v_t ist dabei die turbulente Viskosität.

Aus den veranschaulichten Formeln ergeben sich für das Simulationsbeispiel folgende Werte:

$$k = 0.273$$

$$\varepsilon = 1.563$$

$$\nu = 0.00324$$

7.2. Input-Parameter

Um die Simulation fertig aufzusetzen, werden nachfolgend die Eingangsparameter für die laminare und die turbulente Simulation angeführt. Für die Geschwindigkeit beim Einlass (Anfangsbedingung) wurde in beiden Fällen folgende Formel verwendet:

$$Re = \frac{u D}{\nu} \quad (5)$$

Dabei ist Re die Reynoldszahl in [1], u die Geschwindigkeit in [m/s], D der Durchmesser des durchströmten Kanals in [m] und ν (nu) die kinematische Viskosität in [m²/s]. Die kinematische Viskosität von Wasser bei verschiedenen Temperaturen wurde folgender Tabelle entnommen:

Tabelle 3: Stoffparameter von Wasser bei verschiedenen Temperaturen [14]

ϑ °C	ρ kg m ⁻³	c_p kJ kg ⁻¹ K ⁻¹	β 10 ⁻³ K ⁻¹	η 10 ⁻⁶ kg m ⁻¹ s ⁻¹	ν 10 ⁻⁶ m ² s ⁻¹	σ 10 ⁻³ N m ⁻¹
-30	983,78	4,817	-1,4497	8661,1	8,804	-
-20	993,62	4,418	-0,6576	4362,7	4,931	-
-10	998,14	4,277	-0,2887	2645,2	2,650	77,10
-5	999,27	4,242	-0,1665	2153,5	2,155	76,40
0	999,84	4,218	-0,0672	1792,3	1,793	75,62
5	999,97	4,203	0,0162	1518,7	1,519	74,90
10	999,70	4,192	0,0879	1306,4	1,307	74,20
15	999,10	4,185	0,1507	1138,0	1,139	73,48
20	998,21	4,181	0,2067	1002,0	1,004	72,75
25	997,05	4,179	0,2572	890,45	0,893	71,96
30	995,65	4,177	0,3034	797,68	0,801	71,15
35	994,03	4,177	0,3459	719,62	0,724	70,35
40	993,22	4,177	0,3855	653,25	0,658	69,55
45	990,21	4,178	0,4226	596,32	0,602	-
50	988,04	4,180	0,4578	547,08	0,554	67,90
55	985,69	4,182	0,4912	504,19	0,512	-
60	983,20	4,184	0,5232	466,59	0,475	66,17
65	980,55	4,187	0,5541	433,44	0,442	-
70	977,77	4,190	0,5840	404,06	0,413	64,41
75	974,84	4,193	0,613	377,9	0,388	-
80	971,79	4,197	0,6414	354,49	0,365	62,60
85	968,61	4,201	0,6693	333,48	0,344	-
90	965,31	4,206	0,6967	314,53	0,326	60,74
95	961,89	4,211	0,7238	297,4	0,309	-
99,63 ¹	958,61	4,216	0,7487	282,95	0,295	58,84

Der Druck des durchströmenden Wassers beträgt 5 bar ($5 \cdot 10^5$ Pa). Der durchströmte Metallbock besitzt zu Beginn der Simulation eine konstante Temperatur von 30°C.

7.2.1. Eingabeparameter Laminar

Reynoldszahl	Re =	500
Kanaldurchmesser	D =	0,015 m
Kinematische Viskosität, Wasser	ν =	0,801 *10 ⁻⁶ m ² /s bei 30°C 0,326 *10 ⁻⁶ m ² /s bei 90°C

Daraus ergeben sich folgende Geschwindigkeiten:

Strömungsgeschwindigkeit	u =	0,0267 m/s bei 30°C 0,011 m/s bei 90°C
--------------------------	-----	---

7.2.2. Eingabeparameter Turbulent

Reynoldszahl	Re =	10.000
Kanaldurchmesser	D =	0,015 m
Kinematische Viskosität, Wasser	ν =	0,801 *10 ⁻⁶ m ² /s bei 30°C 0,326 *10 ⁻⁶ m ² /s bei 90°C

Daraus ergeben sich folgende Geschwindigkeiten:

Strömungsgeschwindigkeit	u =	0,533 m/s bei 30°C 0,217 m/s bei 90°C
--------------------------	-----	--

Diese Eingabeparameter bzw. Anfangsbedingungen werden in Dokumenten definiert, welche sich im Ordner „0>fluid“ befinden. Darin sind die Dateien „alphi“, „epsilon“, „k“, „nut“, „p“, „p_rgh“, „T“ und „U“ vorhanden. Im Ordner „0>solid“ sind „p“ und „T“ vorhanden.

- „T“
- „U“
- „solid“
 - „p“
 - „T“

Die Screenshots der Textdateien sind im Anhang in den Abschnitten 10.1. bis 10.2. zu sehen.

Bei den „fluid“-Files müssen bei „epsilon“, „k“, „nut“, „p“, „p_rgh“, „T“ und „U“ unter dem Eintrag „boundaryFields“ die Namen der Geometrien angepasst werden. Im Simulationsbeispiel sind somit „inlet“, „outlet“ und „fluid_to_solid“ einzutragen. Bei den „solid-Files“ „p“ und „T“ sind „wall“ und „solid_to_fluid“ einzutragen. Des Weiteren sind die Anfangsbedingungen anzupassen, zu denen die Geschwindigkeit [m/s], der Druck [Pa] und die Temperatur [K] zählen.

Im Ordner „constant“ müssen folgende Ordner und Files vorhanden sein:

- „constant“
- „fluid“
 - „polyMesh“ (Ordner)
 - „fvSchemes“
 - „fvSolution“
 - „g“
 - „thermophysicalProperties“
 - „transportProperties“
 - „turbulenceProperties“
 - „solid“
 - „polyMesh“ (Ordner)
 - „fvSchemes“
 - „fvSolution“
 - „thermophysicalProperties“

Die Screenshots der Textdateien sind im Anhang in den Abschnitten 10.3. bis 10.4. zu sehen.

Nachdem alle Anpassungen vorgenommen wurden, kann der „yPlus“-Befehl schließlich für „solid“ und „fluid“ ausgeführt werden:

```
>yPlus -region fluid
>yPlus -region solid
```

Da der „yPlus“-Wert im veranschaulichten Beispiel über 100 lag, wurden die Zellen am „fluid-solid“-Grenzbereich mit folgendem Befehl verfeinert:

```
>refineWallLayer '(fluid_to_solid)' 0.5
```

(keine „-region“ Option muss deshalb im Ordner ausgeführt werden, in dem „inlet“ und „outlet“ aufgepatcht wurden)

Nach dem Verfeinern muss der erzeugte „polyMesh“-Ordner nach „constant > fluid“ kopiert werden.

Durch den refineWallLayer-Befehl wurden zusätzliche Zellreihen eingefügt, woraufhin sich „yPlus“ auf einen Wert von durchschnittlich 54.74426 verringert. In der Abbildung 32 ist das verfeinerte Mesh dargestellt:

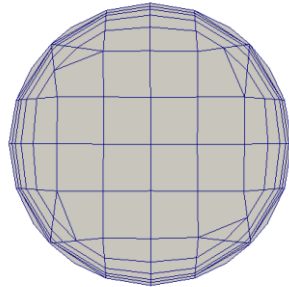


Abbildung 32: Verfeinerte Vernetzung an der Kühlrohr-Geometrie

Für den Festkörper („solid“) wird als Anfangstemperatur ein Wert von 30°C definiert. Des Weiteren sollte die Randbedingung an der äußeren Wand, durch die Änderung des Eintrags unter „externalHeatFluxTemperature“ definiert werden. Dabei wird festgelegt, dass beim Wärmeübergang Festkörper und Umgebung ein konstanter Wärmeübergangskoeffizient von 10 W/m²K verwendet wird. Das heißt, die Umgebung bzw. Luft wird in der Simulation nicht abgebildet, sondern nur mit der genannten, konstanten Randbedingung berücksichtigt.

8. Ausführen der Simulation

Um mit der Simulation starten zu können, müssen die Rahmenbedingungen definiert werden. Diese sind im Ordner „constant“ zu finden. Nachfolgende Eingaben sollten definiert werden:

- Materialkenndaten für „fluid“ und „solid“ werden in „thermophysicalProperties“ angegeben. Diese Files befinden sich im Ordner „constant“. Darin enthalten sind Werte wie die Wärmeleitfähigkeit oder -kapazität, welche für Stahl und Wasser gebräuchlich sind.
- Für die turbulente Simulation wird ein File namens „turbulenceProperties“ benötigt, in dem das verwendete Turbulenzmodell definiert ist.
- Die numerischen Einstellungen, welche in „fvSchemes“ und „fvSolution“ zu finden sind, wurden standardmäßig angenommen.
- In „controlDict“ wurde angegeben, dass der erste Part der Simulation (=Aufheizvorgang) 10 Sekunden dauert und dabei alle 0,1 s gespeichert wird.
- Um einen insgesamt 20 Sekunden dauernden Aufheiz- und Abkühlprozess simulieren zu können, wurde nach 10-sekündiger Aufheizphase die Einlasstemperatur und -geschwindigkeit in den Files „T“ und „U“ im Ordner 10 (Simulationszeitschrittordner) angepasst, um schließlich auch die Abkühlphase darstellen zu können.

Die Screenshots der Textdateien sind dem Anhang 10. zu entnehmen. Schlussendlich wird mit

>chtMultiRegionFoam

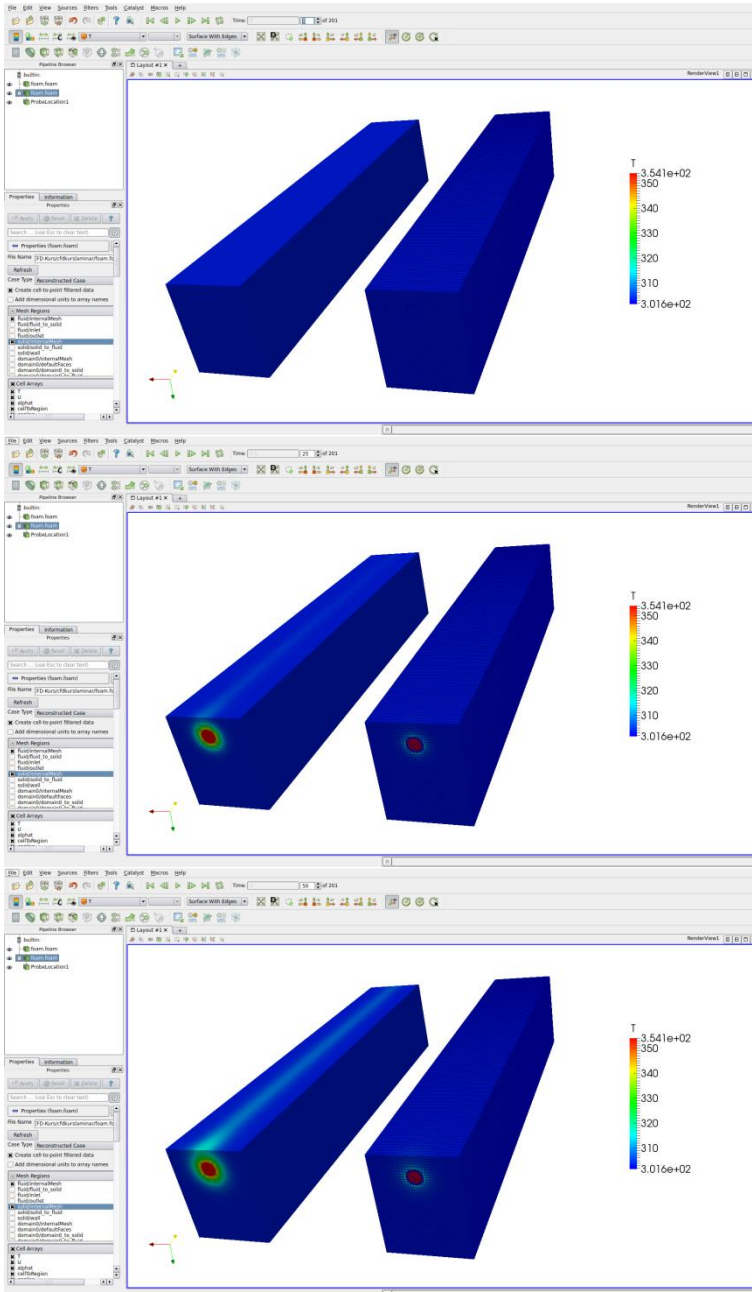
die Simulation gestartet. Nachdem die turbulente Simulation abgeschlossen ist, kann auch die laminare Simulation erstellt werden. Hierfür wird eine Kopie des gesamten turbulenten „case“ („cfdkurs“) erstellt („cfdkurslaminar“). Um mit diesem kopierten „case“ eine laminare Simulation durchführen zu können, müssen folgende Änderungen vorgenommen werden:

- Die Geschwindigkeit im Ordner „0 > U“ wird reduziert, um eine laminare Strömung abzubilden.
- Im File „turbulenceProperties“ im Ordner „constant“ wird die Strömungsart auf laminar geändert.

9. Ergebnisse

9.1. Aufheiz- und Abkühlvorgang

Nach der Simulation wurde der Aufheiz- und Abkühlvorgang in ParaView dargestellt. In den nachfolgenden Abbildungen sind die Ergebnisse in Schritten von 2,5 Sekunden grafisch dargestellt. Dabei ist der turbulente Simulationsfall links und der laminare rechts dargestellt.



Turbulent (links)
Laminar (rechts)

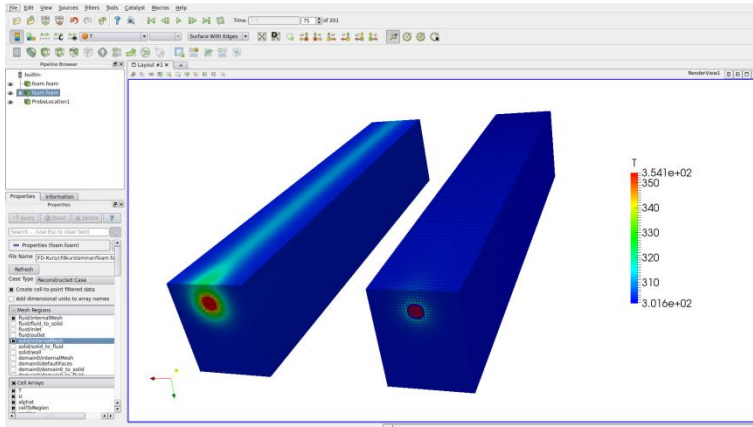
Ausgangszustand
Konstant 30°C
t = 0 Sekunden

Turbulent (links)
Laminar (rechts)

Aufheizvorgang
t = 2,5 Sekunden

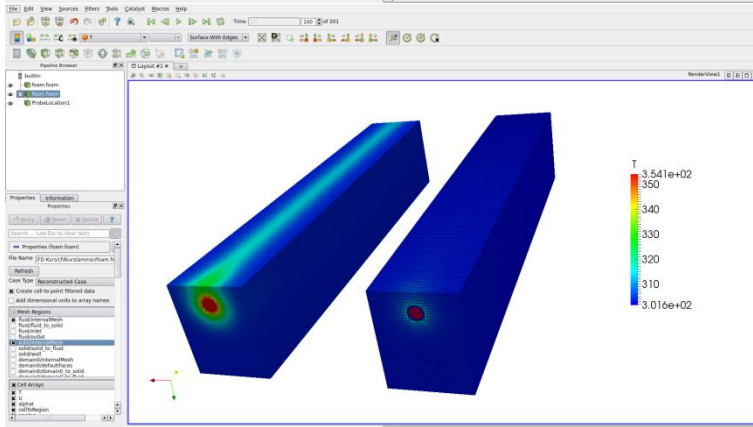
Turbulent (links)
Laminar (rechts)

Aufheizvorgang
t = 5 Sekunden



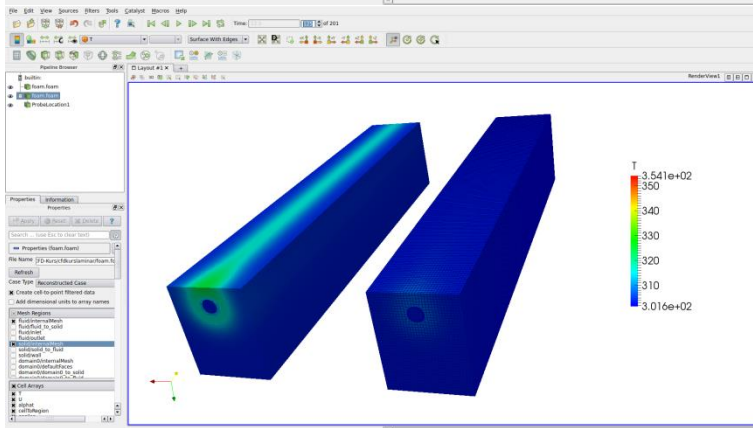
Turbulent (links)
Laminar (rechts)

Aufheizvorgang
 $t = 7,5$ Sekunden



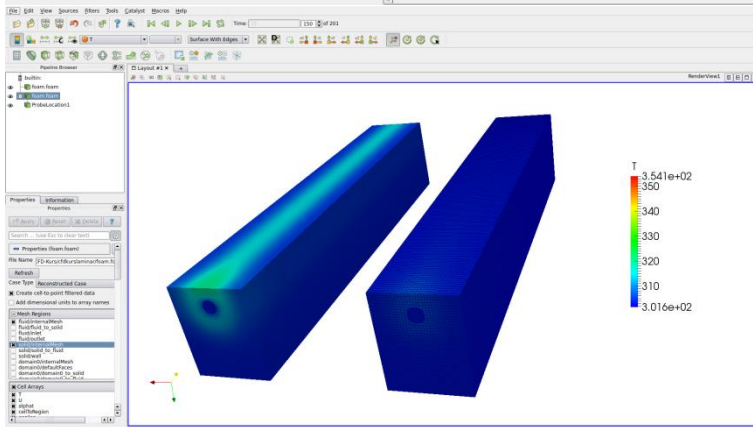
Turbulent (links)
Laminar (rechts)

Ende Aufheizvorgang und
Start der Abkühlung
 $t = 10$ Sekunden



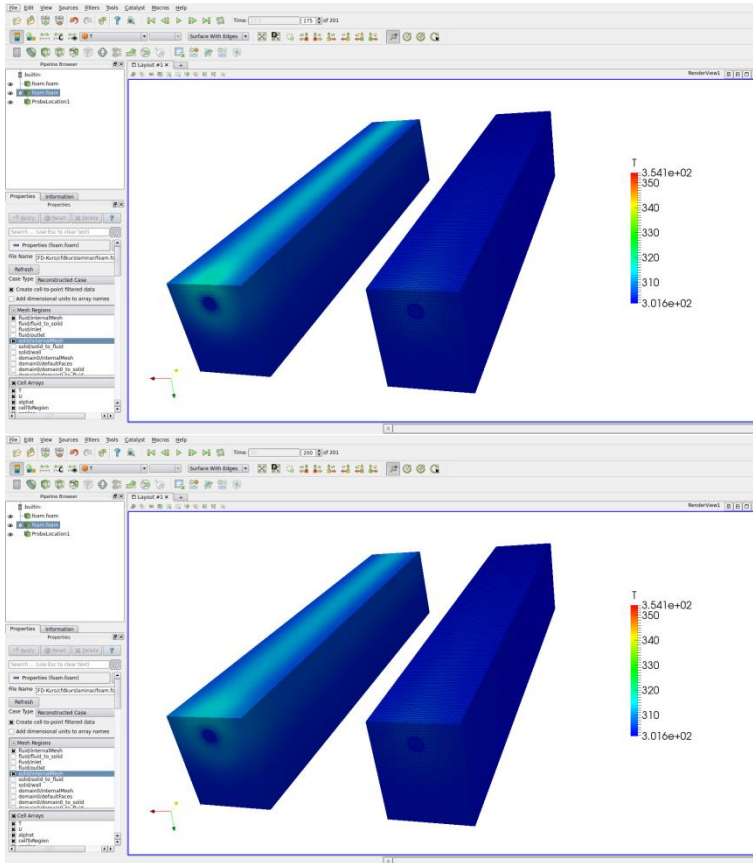
Turbulent (links)
Laminar (rechts)

Abkühlvorgang
 $t = 12,5$ Sekunden



Turbulent (links)
Laminar (rechts)

Abkühlvorgang
 $t = 15$ Sekunden



Turbulent (links)
Laminar (rechts)

Abkühlvorgang
t = 17,5 Sekunden

Turbulent (links)
Laminar (rechts)

Ende Abkühlung
t = 20 Sekunden

Beim Betrachten dieser Abbildungen wird ersichtlich, dass sich die turbulente Variante deutlich dynamischer auf die Temperaturänderung verhält. Um diesen Umstand zu verdeutlichen, können Diagramme von Messpunkten an der Oberfläche erstellt werden. Der Export von Simulationsdaten wird im nächsten Unterabschnitt Ogenauer beschrieben. Für den Export wurden zwei Punkte an der Oberfläche definiert (Abbildung 34).

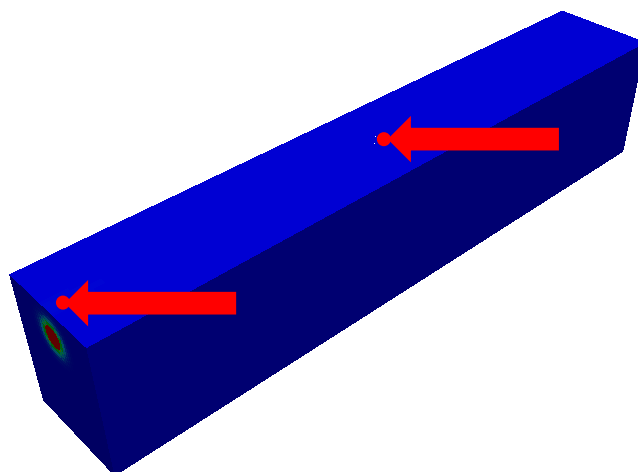


Abbildung 33: Definierte Messpunkte an der Oberfläche der simulierten Geometrie

Der erste Messpunkt befindet sich 10 mm hinter dem Einlass. Der zweite Messpunkt wurde genau in der Mitte des Blocks platziert und befindet sich deshalb 250 mm vom Einlass entfernt. Für diese Punkte werden anschließend die Messwerte generiert und in Diagrammen dargestellt (Abbildung

34 und Abbildung 35). In der Abbildung 34 ist der Temperaturverlauf bei einer Position von 10 mm hinter dem Einlass für den turbulenten und laminaren Fall dargestellt. Die Strömungsgeschwindigkeiten betragen im turbulenten Fall 0,533 m/s bei 30°C und 0,217 m/s bei 90°C. Beim laminaren Fall beträgt die Strömungsgeschwindigkeit 0,0267 m/s bei 30°C und 0,011 m/s bei 90°C.

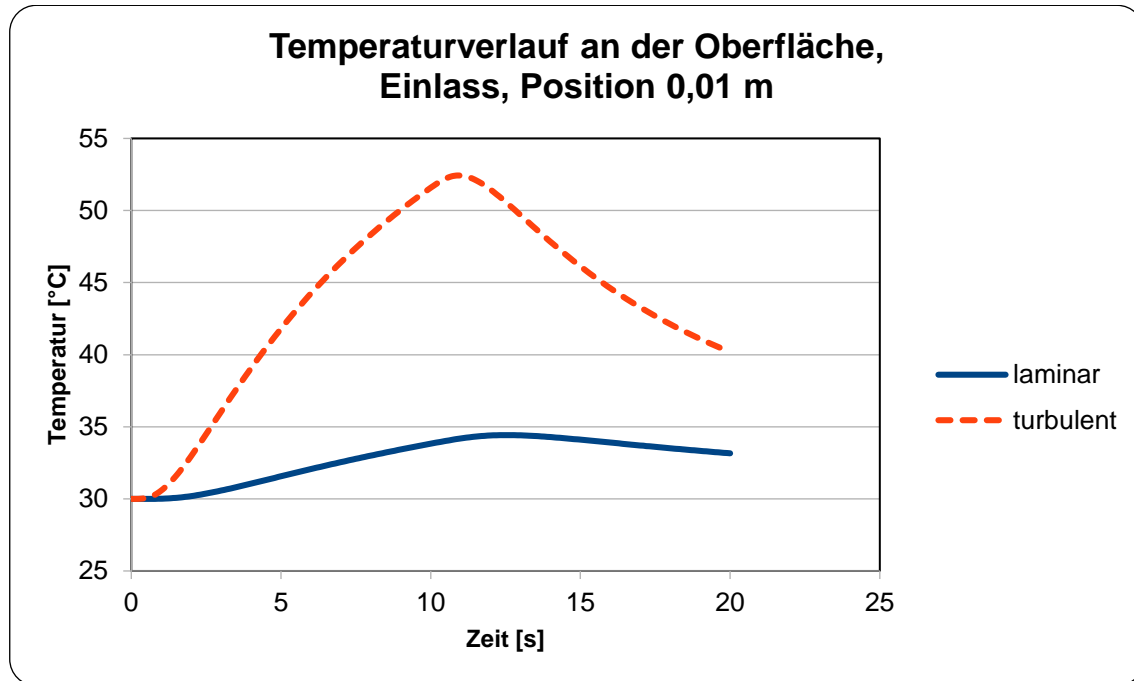


Abbildung 34: Temperaturverlauf an der Oberfläche, Einlass, Position 0,01 m

Erkennbar ist, dass sich der turbulente Fall deutlich dynamischer verhält als der laminare. Dies ist durch die Erhöhung des Wärmeübergangskoeffizienten zwischen Festkörper und Fluid bei hohen Reynolds-Zahlen begründet. In Abbildung 35 ist der Temperaturverlauf bei einer Position von 250 mm hinter dem Einlass dargestellt.

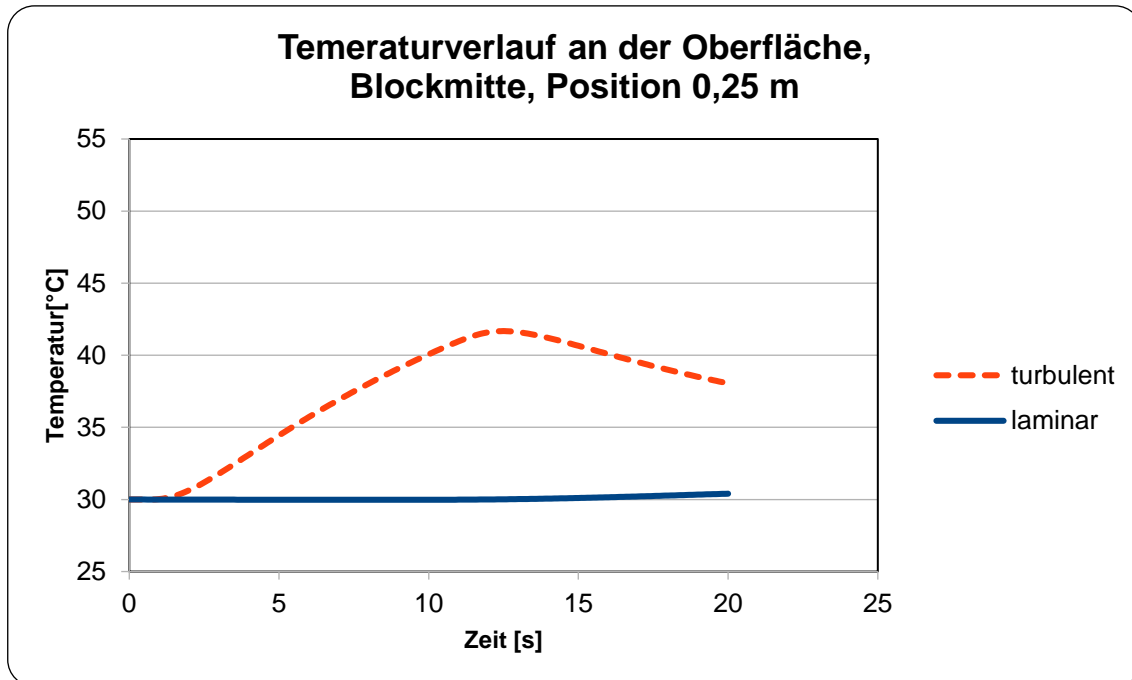


Abbildung 35: Temperaturverlauf an der Oberfläche, Blockmitte, Position 0,25 m

Dabei ist zu erkennen, dass der Temperaturanstieg im Vergleich zum Messpunkt bei 10 mm um ca. 10°C niedriger ausfällt. Beim laminaren Simulationsfall findet kein Temperaturanstieg statt, da die Strömungsgeschwindigkeit so gering ist, dass das warme Wasser erst mitten im Abkühlvorgang beim Messpunkt bei 250 mm ankommt.

9.2. Export der Messwerte

Nachdem die Geometrie in ParaView angezeigt wird, kann mit der Funktion „probeLocation“ (Abbildung 36) ein Messpunkt auf der Oberfläche der simulierten Geometrie erzeugt werden.

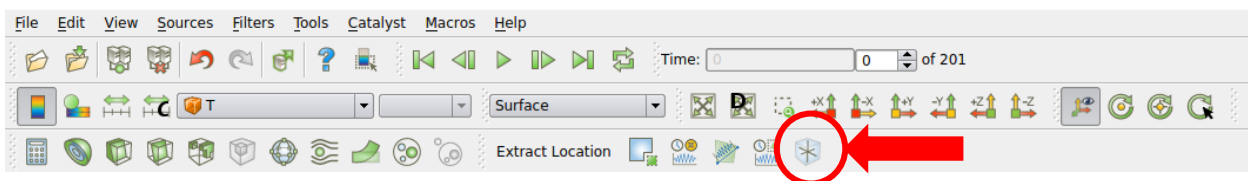


Abbildung 36: ParaView Funktion „probeLocation“

Mit dem Befehl unter „File > Save Data“ können danach ausgewählte Daten für die Zeitschritte ausgegeben werden. Im gezeigten Beispiel werden die Temperaturdaten ausgegeben. Der Nachteil beim Export der Daten auf diese Weise ist, dass für jeden Zeitschritt eine Excel-Datei erstellt wird. Diese können jedoch anschließend im Ordner, in dem die Files abgespeichert sind, mit dem Befehl

```
for i in `seq 0 1 200`; do cat T.$i.csv >> masterT01.csv; done
```

zu einer einzigen Excel-Datei zusammengeführt werden. Mit diesem Befehl werden jedoch auch die Überschriften der 200 Dateien zusammengeführt, sodass jede Messwertzeile eine überflüssige Überschriftenzeile besitzt. Um diese zu löschen, müssen folgende Befehle ausgeführt werden:

- Öffnen der Datei
>vim erstelltecsvdatei.csv
- Start der Makroaufzeichnung
Zweifaches Betätigen der q-Taste
- Zeile löschen
Zweifaches Betätigen der d-Taste
Pfeil nach unten, um Zeile zu überspringen
- Makroaufzeichnung beenden
Einfaches Betätigen der q-Taste
- Befehl mehrfach eine bestimmte Anzahl ausführen
Anzahl der Ausführungen am Nummernblock eingeben
Mit @q Ausführung starten
- Speichern und Beenden
:wq (write and quite)

ParaView besitzt umfangreiche Möglichkeiten zur Auswertung verschiedenster Simulationsfälle. Die hier veranschaulichten Beispiele stellen nur sehr kleine Teile der vorhandenen Möglichkeiten dar. Weitere Informationen zu den Auswertungsmethoden und Tutorials für die Bedienung von ParaView können direkt der Website von ParaView [15] entnommen werden.

10. Anhang

10.1. Ordner „0 > fluid“

```

1 /*----- C++ -----*/
2 =====
3 \      /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox
4  \    /  O p e r a t i o n | Version: 3.0.1
5   \  /  A n d              | Web:      www.OpenFOAM.org
6    \/  M a n i p u l a t i o n
7 -----*/
8 FoamFile
9 {
10  version      2.0;
11  format       ascii;
12  class        volScalarField;
13  object       alphas;
14 }
15 // *****
16
17 dimensions    [1 -1 -1 0 0 0];
18
19 internalField uniform 0.56;
20
21 boundaryField
22 {
23     ".*"
24     {
25         type          calculated;
26         value         uniform 0.56;
27     }
28 }
29
30
31 // *****

```

```

1 /*----- C++ -----*/
2 =====
3 \      /  F i e l d      | OpenFOAM: The Open Source CFD Toolbox
4  \    /  O p e r a t i o n | Version: 2.3.0
5   \  /  A n d              | Web:      www.OpenFOAM.org
6    \/  M a n i p u l a t i o n
7 -----*/
8 FoamFile
9 {
10  version      2.0;
11  format       ascii;
12  class        volScalarField;
13  location     "0/cylinder";
14  object       epsilon;
15 }
16 // *****
17
18 dimensions    [0 2 -3 0 0 0];
19
20 internalField uniform 1.563;
21
22 boundaryField
23 {
24     inlet
25     {
26         type          fixedValue;//epsilonWallFunction;
27         value         uniform 1.563;
28         /*
29         type          uniformFixedValue;
30         uniformValue  csvFile;
31         csvFileCoeffs//uniformValueCoeffs
32         {
33             fileName    "SFOAM_CASE/T.csv"
34             outOfBounds  clamp;
35             nHeaderLine  1;
36             refColumn    0;
37             componentColumns (5); // vector example
38             mergeSeparators 0;
39         */
40     }
41     outlet
42     {
43         type          zeroGradient;//epsilonWallFunction;
44     }
45     fluid_to_solid
46     {
47         type          epsilonWallFunction;
48         value         uniform 1.563;//3.512;
49     }
50 }
51
52
53 // *****

```

```

1 |/*-----*- C++ -*-----*/
2 |=====
3 | \   /   F i e l d   |   OpenFOAM: The Open Source CFD Toolbox
4 |  \ /    O p e r a t i o n   |   Version: 2.3.0
5 |   / \    A n d   |   Web: www.OpenFOAM.org
6 |  /   \    M a n i p u l a t i o n   |
7 |-----*/
8 FoamFile
9 {
10  version      2.0;
11  format       ascii;
12  class        volScalarField;
13  location     "0/cylinder";
14  object       k;
15 }
16 // ***** //
17
18 dimensions   [0 2 -2 0 0 0];
19
20 internalField uniform 0.273;
21
22 boundaryField
23 {
24   inlet
25   {
26     type          fixedValue;//kqRWallFunction;
27     value         uniform 0.273;
28 /*
29     type          uniformFixedValue;
30     uniformValue  csvFile;
31     csvFileCoeffs//uniformValueCoeffs
32     {
33       fileName    "$FOAM_CASE/T.csv"
34       outOfBounds  clamp;
35       nHeaderLine  1;
36       refColumn    0;
37       componentColumns (4); // vector example
38       mergeSeparators 0;
39 */
40   }
41 }
42   outlet
43   {
44     type          zeroGradient;//kqRWallFunction;
45   }
46   fluid_to_solid
47   {
48     type          kqRWallFunction;
49     value         uniform 0.273;
50   }
51 }
52
53
54 // ***** //

```

```

1 |/*-----*- C++ -*-----*/
2 |=====
3 | \   /   F i e l d   |   OpenFOAM: The Open Source CFD Toolbox
4 |  \ /    O p e r a t i o n   |   Version: 2.3.0
5 |   / \    A n d   |   Web: www.OpenFOAM.org
6 |  /   \    M a n i p u l a t i o n   |
7 |-----*/
8 FoamFile
9 {
10  version      2.0;
11  format       ascii;
12  class        volScalarField;
13  location     "0/inside";
14  object       nut;
15 }
16 // ***** //
17
18 dimensions   [0 2 -1 0 0 0];
19
20 internalField uniform 0.00324;
21
22 boundaryField
23 {
24   inlet
25   {
26     type          fixedValue;//calculated;
27     value         uniform 0.00324;
28   }
29   outlet
30   {
31     type          zeroGradient;//calculated;
32   }
33 }
34   fluid_to_solid
35   {
36     type          nutkWallFunction;
37     value         uniform 0.00324;
38   }
39 }
40
41
42 // ***** //

```

```

1 /*-----* C++ -*-----*/
2
3 ===== Field | OpenFOAM: The Open Source CFD Toolbox
4 Operation | Version: 3.0.1
5 And | Web: www.OpenFOAM.org
6 Manipulation |
7 /*-----*
8 FoamFile
9 {
10   version      2.0;
11   format       ascii;
12   class        volScalarField;
13   object       p;
14 }
15 // *****
16
17 dimensions     [1 -1 -2 0 0 0];
18
19 internalField  uniform 5e5;//1e5;
20
21 boundaryField
22 {
23   inlet
24   {
25     type        zeroGradient;
26     value       uniform 0;
27   }
28   outlet
29   {
30     type        fixedValue;
31     value       uniform 5e5;
32   }
33   fluid_to_solid
34   {
35     type        zeroGradient;//calculated;
36     value       uniform 0;
37   }
38 }
39 }
40
41 // *****
42
43 /*-----* C++ -*-----*/
44
45 ===== Field | OpenFOAM: The Open Source CFD Toolbox
46 Operation | Version: 3.0.1
47 And | Web: www.OpenFOAM.org
48 Manipulation |
49 /*-----*
50 FoamFile
51 {
52   version      2.0;
53   format       ascii;
54   class        volScalarField;
55   object       T;
56 }
57 // *****
58
59 dimensions     [0 0 0 1 0 0];
60
61 internalField  uniform 303.15;
62
63 boundaryField
64 {
65   inlet
66   {
67     type        fixedValue;
68     value       uniform 303.15;
69
70     /*
71     type        uniformFixedValue;
72     uniformValue csvFile;
73     csvFileCoeffs//uniformValueCoeffs
74     {
75       fileName      "$FOAM_CASE/T.csv"
76       outOfBounds    clamp;
77       nHeaderLine    0;//1;
78       refColumn      0;
79       componentColumns (1); // vector example
80       mergeSeparators 0;
81     }
82     */
83   }
84   outlet
85   {
86     type        zeroGradient;
87   }
88   fluid_to_solid
89   {
90     // type        fixedValue;
91     // value       uniform 303.15;
92
93     type        compressible::turbulentTemperatureCoupledBaffleMixed;
94     value       uniform 311.55;
95     phase       fluid;
96     d           0.01;
97     L           0.5;
98     Cp          4181;
99   }
100 }

```



```

57     Tnbr          T;
58     kappa         fluidThermo;
59     kappaName     k;
60     kappaMethod   fluidThermo;
61
62   }
63
64 }
65
66 // ***** //

1 /*-----* C++ -*-----*/
2 |=====|
3 | \ / \ / | F i e l d | OpenFOAM: The Open Source CFD Toolbox
4 | \ / \ / | O p e r a t i o n | Version: 2.3.0
5 | \ / \ / | A n d | Web: www.OpenFOAM.org
6 | \ / \ / | M a n i p u l a t i o n |
7 |-----*|
8 FoamFile
9 {
10  version      2.0;
11  format       ascii;
12  class        volVectorField;
13  location     "0";
14  object       U;
15 }
16 // ***** //
17
18 dimensions    [0 1 -1 0 0 0];
19
20 internalField uniform (0 0.533 0);
21
22 boundaryField
23 {
24   fluid_to_solid
25   {
26     type        fixedValue;
27     value       uniform (0 0 0);
28   }
29   inlet
30   {
31     type        fixedValue;//
32     value       uniform (0 0.533 0);//uniformFixedValue;
33 /*
34     uniformValue csvFile;
35     csvFileCoeffs
36     {
37       nHeaderLine 0;
38       refColumn   0;
39       componentColumns 3(3 2 3);
40       separator    ",";
41       mergeSeparators 0;
42       fileName     "$FOAM_CASE/T.csv";
43     }*/
44   }
45   outlet
46   {
47     type        zeroGradient;
48   }
49 }
50
51 // ***** //

```

10.2. Ordner „0 > Solid“

```

1 /*-----* C++ -*-----*/
2
3 =====
4 \ \ \ \ \ Field | OpenFOAM: The Open Source CFD Toolbox
5 \ \ \ \ \ O peration | Version: 3.0.1
6 \ \ \ \ \ A nd | Web: www.OpenFOAM.org
7 \ \ \ \ \ M anipulation |
8 -----*-----*/
9 FoamFile
10 {
11     version      2.0;
12     format       ascii;
13     class        volScalarField;
14     object       p;
15 }
16 // *****
17 dimensions      [1 -1 -2 0 0 0];
18
19 internalField   uniform 1e5;//12.66;
20
21 boundaryField
22 {
23     solid_to_fluid
24     {
25         type          zeroGradient;//calculated;//fixedValue;
26         value         uniform 1e5;//12.66;
27     }
28     wall
29     {
30         type          calculated;//fixedValue;
31         value         uniform 1e5;//12.66;
32     }
33 }
34
35 }
36
37 // *****

```

```

1 /*-----* C++ -*-----*/
2
3 =====
4 \ \ \ \ \ Field | OpenFOAM: The Open Source CFD Toolbox
5 \ \ \ \ \ O peration | Version: 2.3.0
6 \ \ \ \ \ A nd | Web: www.OpenFOAM.org
7 \ \ \ \ \ M anipulation |
8 -----*-----*/
9 FoamFile
10 {
11     version      2.0;
12     format       ascii;
13     class        volScalarField;
14     location     "0";
15     object       T;
16 }
17 // *****
18 dimensions      [0 0 0 1 0 0 0];
19
20 internalField   uniform 303.15;
21
22 boundaryField
23 {
24     wall
25     {
26         type          externalWallHeatFluxTemperature;
27         kappa         solidThermo;
28         Ta            uniform 293.15;
29         h             uniform 10.0;
30         value         uniform 303.15;
31         kappaName     none;
32     }
33     solid_to_fluid
34     {
35         type          compressible::turbulentTemperatureCoupledBaffleMixed;
36         value         uniform 303.15;
37         phase         solid;
38         d             0.01;
39         L             0.5;
40         Cp            490;
41         Tnbr         T;
42         kappa         solidThermo;
43         kappaName     k;
44         kappaMethod   solidThermo;
45     }
46 }
47
48 }
49
50
51 // *****

```

10.3. Ordner „constant > fluid“

```

1 /*-----* C++ */
2
3 ===== Field | OpenFOAM: The Open Source CFD Toolbox
4 \ \ \ \ \ Operation | Version: 4.0
5 \ \ \ \ \ And | Web: www.OpenFOAM.org
6 \ \ \ \ \ Manipulation |
7 /*-----*/
8 FoamFile
9 {
10   version      2.0;
11   format       ascii;
12   class        dictionary;
13   object       fvSchemes;
14 }
15 // ***** //
16
17 ddtSchemes
18 {
19   default Euler;
20 }
21
22 gradSchemes
23 {
24   default      Gauss linear;
25 }
26
27 divSchemes
28 {
29   default      none;
30
31   div(phi,U)   Gauss upwind;
32   div(phi,K)   Gauss linear;
33   div(phi,h)   Gauss upwind;
34   div(phi,k)   Gauss upwind;
35   div(phi,epsilon) Gauss upwind;
36   div(phi,R)   Gauss upwind;
37   div(R)       Gauss linear;
38   div((rho*nuEff)*dev2(T(grad(U)))) Gauss linear;
39   div((muEff*dev2(T(grad(U)))) Gauss linear;
40   div(phi,v,p) Gauss linear;
41   div(phi,e)   Gauss linear;
42 }
43
44 laplacianSchemes
45 {
46   default      Gauss linear corrected;
47 }
48
49 interpolationSchemes
50 {
51   default      linear;
52 }
53
54 snGradSchemes
55 {
56   default      corrected;
57 }
58 }
59
60 fluxRequired
61 {
62   default      no;
63   p_rgh;
64 }
65 // ***** //
66

```

```

1 |/*-----* C++ *-----*/
2 |
3 |-----|
4 | \   / | Field | OpenFOAM: The Open Source CFD Toolbox
5 |  \ /  | Operation | Version: 4.0
6 |   /   | And | Web: www.OpenFOAM.org
7 |  /    | Manipulation |
8 |-----|
9 |FoamFile
10| {
11|   version      2.0;
12|   format       ascii;
13|   class        dictionary;
14|   object       fvSolution;
15|}
16|// *****
17|solvers
18| {
19|   rho
20|   {
21|     solver      PCG;
22|     preconditioner DIC;
23|     tolerance   1e-7;
24|     relTol      0.1;
25|   }
26|
27|   rhoFinal
28|   {
29|     $rho;
30|     tolerance   1e-7;
31|     relTol      0;
32|   }
33|
34|   p_rgh
35|   {
36|     solver      GAMG;
37|     tolerance   1e-7;
38|     relTol      0.01;
39|
40|     smoother    GaussSeidel;
41|
42|     cacheAgglomeration true;
43|     nCellsInCoarsestLevel 10;
44|     agglomerator faceAreaPair;
45|     mergeLevels  1;
46|   }
47|
48|
49|   p_rghFinal
50|   {
51|     $p_rgh;
52|     tolerance   1e-7;
53|     relTol      0.01;
54|   }
55|
56|   "(e|U|h|k|epsilon|R)"
57|   {
58|     solver      PBiCG;
59|     preconditioner DILU;
60|     tolerance   1e-7;
61|     relTol      0.1;
62|   }
63|
64|   "(e|U|h|k|epsilon|R)Final"
65|   {
66|     $U;
67|     tolerance   1e-7;
68|     relTol      0;
69|   }
70| }
71|
72|PIMPLE
73| {
74|   momentumPredictor yes;
75|   nCorrectors        2;
76|   nNonOrthogonalCorrectors 1;
77| }
78|
79|relaxationFactors
80| {
81|   equations
82|   {
83|     "h.*"      0.1;
84|     "U.*"      0.5;
85|   }
86| }
87|
88|// *****

```


11. Literatur

- [1] „CAESES Software > CAESES“ . .
- [2] E. S. I. Group, „Visual-CFD for OpenFOAM“, *ESI Group*, 10-Juni-2016. [Online]. Verfügbar unter: <https://www.esi-group.com/software-solutions/virtual-environment/cfd-multiphysics/visual-cfd-openfoam>. [Zugegriffen: 03-Dez-2018].
- [3] „SimFlow CFD Software - OpenFOAM® GUI“, *simFlow CFD*. [Online]. Verfügbar unter: <https://sim-flow.com/>. [Zugegriffen: 03-Dez-2018].
- [4] „MantiumFlow, OpenFOAM CFD Simulations, download demo and tutorials“, *MantiumFlow, CFD Simulation Software with OpenFOAM*. [Online]. Verfügbar unter: <https://mantiumflow.com/>. [Zugegriffen: 03-Dez-2018].
- [5] „OpenFOAM® Documentation“. [Online]. Verfügbar unter: <https://www.openfoam.com/documentation/>. [Zugegriffen: 03-Dez-2018].
- [6] P. Geisser, *Temperiertechnik - Theorie und Praxis*, 3. Aufl. HB-Therm AG, 2016.
- [7] B. Foundation, „blender.org - Home of the Blender project - Free and Open 3D Creation Software“, *blender.org* . .
- [8] A. Pflanzler und F. Schneider, „Export von Creo-Dateien in das STL-Format“, 2017.
- [9] E. Kobler, „Abbildung des Maschinenverhaltens eines Spritzgießprozesses mittels Füllsimulation einer Stufenplatte in OpenFoam“. Johannes Kepler Universität Linz, 29-Mai-2015.
- [10] OpenCFD, „Mesh generation with the snappyHexMesh utility“. [Online]. Verfügbar unter: <http://www.openfoam.com/documentation/user-guide/snappyHexMesh.php>. [Zugegriffen: 22-Sep-2017].
- [11] C. Soulaine, „Introduction to computational fluid mechanics using the OpenFOAM® technology « Simulation in porous media from pore to large scale »“, 27-Juni-2016.
- [12] R. Schwarze, *CFD-Modellierung*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.
- [13] „OpenFOAM: k-epsilon“. [Online]. Verfügbar unter: <http://www.openfoam.com/documentation/cpp-guide/html/guide-turbulence-ras-k-epsilon.html>. [Zugegriffen: 27-Sep-2017].
- [14] „Stoffwerte von Wasser“. [Online]. Verfügbar unter: <http://www.uni-magdeburg.de/isut/LSS/Lehre/Arbeitsheft/IV.pdf>. [Zugegriffen: 26-Sep-2017].
- [15] „ParaView“. [Online] Verfügbar unter <http://www.paraview.com>