



## **KARLSRUHE INSTITUTE OF TECHNOLOGY**

**Department of Electrical Engineering and Information Technology**

**Light Technology Institute (LTI)**

Prof. Dr. rer. nat. Uli Lemmer

**Department of Mechanical Engineering**

**Institute of Microstructure Technology (IMT)**

Prof. Dr. Manfred Kohl

## **Master's Thesis**

Technological Evaluation of Printed, Large Surface  
Thermoelectric Generators for Waste Heat Recuperation:  
Design, simulation and optimization

by **Leonard Franke**

to

obtain the academic degree

## **MASTER OF SCIENCE**

**Mechanical Engineering**

**(M.Sc.)**

submitted by: Leonard Franke  
(born April 11, 1990 in Mölln)  
on: September 17, 2017

Advisor: M.Sc. Matthias Hecht (LTI)

1. Referent: Prof. Dr. rer. nat. Uli Lemmer (LTI)

2. Referent: Prof. Dr. Manfred Kohl (IMT)

---

## Declaration of Originality

I hereby declare that the thesis submitted is my own unaided work. I have not used other than the declared sources / resources and I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Karlsruhe, 17.09.2017

(place and date)

Leonard Frohn

(signature)

---

## Acknowledgement

Firstly, I would like to express my sincere gratitude to Prof. Uli Lemmer and Prof. Manfred Kohl for the opportunity to pursue this interdisciplinary research project for my master's thesis at the Light Technology Institute.

In addition, I want to express my gratitude to the otego GmbH who advised me during the whole research process and allowed this research to be my own work.

I particularly would like to thank my thesis advisor Matthias Hecht. He would always find the time whenever I had a question or ran into a trouble spot with my research or writing. He provided me with advice and assistance whenever I thought I needed it and gave me the freedom to follow my own approach in this project.

I am also grateful to all the department members for their help and support.

I would like to thank my family for supporting me throughout my studies and my life in general.

---

## Abstract

In future, there will be an increasing demand for waste heat recovery systems (WHRS). So far current technologies are not expected to fulfill the requirement expected for a broad deployment. Thermoelectric generators (TEGs) for the direct transformation from heat to electricity represent hereby a promising alternative. Novel printed TEGs based on organic and inorganic materials (OTEGs) will be brought to market maturity soon. The usage of new cost-efficient materials and the targeted automation as well as the scalability of the production process offer a considerable cost reduction compared to conventional TEGs, making this technology interesting for the recovery of large amounts of waste heat. Moreover, the used materials are environmentally compliant. For first time, a WHRS based on a simple plate heat exchanger design equipped with this new generation of TEGs was evaluated from a technological point of view. For this the thermal behavior of the system was simulated using the Simulink modelling environment and at the same time a fluid dynamical examination of the heat exchanger channels was conducted to determine the dissipated energy by the fluids in the heat exchanger using OpenFOAM. To determine the optimal operational conditions of the system the elaborated models were used. The models demonstrated that current generator efficiencies of  $ZT$  0.1 could deliver next to the auxiliary energy also a surplus power of approximately 250 W compared to a heat power transferred of 39.44 kW. For future TEGs with an efficiency of  $ZT$  0.5 an increase to 1.2 kW was determined by the models which is equivalent to a net system efficiency close to 2.5 %. The waste heat was bundled prior in a thermal oil cycle with a temperature of 230 °C while for the cooling cycle water at 15 °C was assumed.

---

## Zusammenfassung

In Zukunft wird es einen wachsenden Bedarf an Wärmerückgewinnungssystemen (WRG-Systeme) zur Verstromung anfallender Abwärme zur Effizienzsteigerung geben. Derzeitige Technologien erfüllen dabei noch nicht die Anforderungen, die in Zukunft einen breiten Einsatz von WRG-Systemen erwarten lassen. Thermoelektrische Generatoren (TEGs) zur direkten Umwandlung von Wärme in elektrische Energie repräsentieren hier eine vielversprechende Alternative. Neuartige gedruckte TEGs basierend auf organischen und anorganischen Materialien (OTEGs) werden in Kürze zur Marktreife gebracht. Die Verwendung günstiger Materialien und die Automatisierung sowie Skalierbarkeit im Herstellungsprozess dieser neuen Generatorgeneration bieten einen deutlichen Kostenvorteil gegenüber konventionellen TEGs und machen sie auch für die Rückgewinnung großer Wärmemengen mit niedrigen Wärmestromdichten interessant. Darüber hinaus kommen wesentlich umweltschonendere Materialien zum Einsatz. Im Zuge dieser Masterarbeit wurde erstmalig ein WRG-System evaluiert, welches diese neue Generation von TEGs in einem einfachen Plattenwärmeübertrager integriert. Dafür wurde das thermische Verhalten des WRG-Systems in Simulink simuliert und mittels strömungstechnischer Modelle der Druckverluste im Wärmeübertrager untersucht (OpenFOAM). Zur Ermittlung des optimalen Betriebspunktes fand anschließend eine Optimierung der variablen Systemparameter statt. Dabei zeigte sich, dass bereits die derzeitige Generation von gedruckten TEGs mit einem  $ZT$  von 0.1 nicht nur den Eigenenergiebedarf, nötig für die Förderung der beiden Wärmeträger decken kann, sondern darüber hinaus bei einer übertragenen Wärmeleistung von 39,44 kW eine elektrische Leistung von ca. 250 W generiert. Für zukünftige Generatoren mit einem durchschnittlichen  $ZT$  von 0.5 beträgt die maximale Systemleistung 1.26 kW was einer Systemeffizienz von ca. 2.5 % entspricht. Die Abwärme wurde hierbei in einem Thermoölkreislauf mit einer Temperatur von 230 °C gebündelt; im Kühlkreislauf wurde Wasser mit einer Starttemperatur von 15 °C verwendet.

---

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	General Introduction .....	1
1.2	Waste Heat Potential of the German Industry.....	2
1.3	Aims of this Work.....	3
<b>2</b>	<b>Theoretical Background</b>	<b>5</b>
2.1	Thermoelectric Effects .....	5
2.2	Thermoelectric Materials .....	6
2.2.1	State of Technology: Thermoelectric Materials .....	8
2.3	TEG Power Generation.....	9
2.3.1	Printed TEGs .....	10
2.3.2	State of Technology: Power Generation .....	12
2.4	Fundamentals of Heat Transfer .....	13
2.4.1	Conduction and Convection .....	14
2.4.2	Radiation .....	15
2.5	Computational Fluid Dynamics .....	15
2.6	Similitude .....	20
2.7	Heat Exchangers .....	22
2.8	Software.....	23
2.8.1	Creo Parametric 3.0 .....	23
2.8.2	Salome .....	23
2.8.3	Simulink.....	23
2.8.4	OpenFOAM .....	23
2.8.5	gnuplot .....	24
<b>3</b>	<b>Methodology</b>	<b>25</b>
3.1	Case Scenario .....	25
3.2	Constructive Concept.....	26
3.3	Thermal Model: Methodology .....	29
3.3.1	Reducing Complexity .....	30
3.3.2	Thermal oil and Water .....	31
3.3.3	Newton's Law on Cooling .....	31
3.3.4	Total Thermal Resistance $R_{tot}$ .....	31
3.3.5	TEG Thickness .....	33
3.3.6	Global Temperature Development .....	34
3.3.7	Local Temperature Development .....	35

---

3.3.8	Putting it all together.....	35
3.4	Fluid Dynamical Model: Methodology .....	38
3.4.1	Geometrical Model .....	38
3.4.2	Meshing.....	39
3.4.3	Solver .....	44
3.4.4	Boundary Conditions and Initial Conditions.....	45
3.4.5	Post Processing .....	47
3.4.6	OpenFOAM .....	48
3.5	Shell Scripting .....	51
3.6	Merging the Models .....	53
<b>4</b>	<b>Results</b>	<b>54</b>
4.1	Thermal Model .....	54
4.1.1	TEG Thickness .....	54
4.1.2	Global Temperature Development .....	55
4.1.3	Local Temperature Development .....	56
4.2	CFD Model .....	57
4.2.1	Meshing.....	57
4.2.2	Boundary Conditions and Initial Conditions.....	60
4.2.3	Pressure Drop.....	63
4.2.4	Validation of the Fluid Dynamical Model .....	66
4.2.5	Monitoring of the CFD Solutions .....	68
4.3	Merging the Models .....	69
4.4	Evaluation of the WHRS Design .....	73
<b>5</b>	<b>Conclusion</b>	<b>75</b>
	<b>References</b>	<b>78</b>
	<b>List of Tables</b>	<b>80</b>
	<b>List of Figures</b>	<b>81</b>
	<b>Appendix</b>	<b>83</b>
	Fluid Parameters	83
	Exemplary Script snappyHexMesh	84
	Shell Script	88
	Hardware	92

---

## List of Abbreviations

WHRS	Waste heat recovery system
WHR	Waste heat recovery
TEGs	Thermoelectric generators
OTEGs	Organic thermoelectric generators
WRG	Wärmerückgewinnung
RTGs	Radio isotopic thermoelectric generators
$\text{Bi}_2\text{Te}_3$	Bismuth telluride
$\text{Mg}_2\text{Si}$	Magnesium silicate
PEDOT:PSS	Poly(3,4-ethylenedioxythiophene) polystyrene sulfonate
ORC	Organic rankine cycle
NSE	Navier-Stockes equations
DNS	Direct numerical simulation
LES	Large eddy simulation
RANS	Reynolds-averaged Navier-Stockes equations
FVM	Finite volume method
CAD	Computer-aided design
$\text{CO}_2$	Carbon dioxide
MPPT	Maximum power point tracking
VDI	Verein Deutscher Ingenieure
CFD	Computational fluid dynamics
STEP	Standard for the Exchange of Product model data
IGES	Initial Graphics Exchange Specification
STL	Standard tessellation language
RAM	Random access memory
swak4Foam	Swiss army knife for Foam
BC	Boundary condition



---

VTK	Visualization Toolkit
SIMPLE	Semi-implicit Method for Pressure Linked Equations
GUI	General user interface
CLI	Command line interface
BASH	Bourne-again shell

---

# Nomenclature

## Physical entities

$U$	Voltage
$T$	Temperature
$S$	Seebeck coefficient
$\Pi$	Peltier coefficient
$Q$	Total amount of heat
$q$	Heat density
$t$	Time
$\tau$	Thomson coefficient or friction coefficient
$ZT$	Dimensionless figure of merit
$\overline{ZT}$	integral mean value of the dimensionless figure of merit
$Z$	Figure of merit
$\sigma$	Electrical conduction or Stefan-Boltzman constant
$\kappa, \lambda$	Thermal conductivity
$\alpha$	Heat transfer coefficient
$A$	Area or coefficient matrix
$e$	Emissivity or inner energy
$\rho$	Density
$u, v$	Velocity
$p$	Pressure
$h$	Enthalpy
$f$	External forces
$\mu$	Molecular viscosity
$r$	Residual
$\Phi$	Flow parameter
$\vec{X}$	Solution vector

---

$\vec{b}$	Source term
$Re$	Reynolds number
$Nu$	Nusselt number
$Pr$	Prandtl number
$NTU$	Number of transfer units
$\nu$	Kinetic viscosity
$\eta$	Efficiency
$L$	Length
$k$	Heat transmittance coefficient or turbulent kinetic energy
$\dot{m}$	Mass flow rate
$\dot{V}$	Volume flow rate
$R$	Thermal resistance
$l$	Streaming length
$c$	Heat capacity
$d$	Thickness
$b$	Channel height
$W$	Width
$\varepsilon$	Dissipation rate of the turbulent kinetic energy
$H$	Height

### Indices

$th$	Thermal
$h$	Hot
$c$	Cold
$tot$	Total
$cond$	Conduction
$conv$	Convection
$rad$	Radiation
$surf$	Surface

---

$\infty$	At large distance to
<i>car</i>	Carnot
<i>p</i>	At constant pressure
<i>char</i>	Characteristic
<i>TEG</i>	Thermoelectric generator
<i>effect, eff</i>	Effective
<i>OTEG</i>	Organic thermoelectric generator
$\tau$	Turbulent

# 1 Introduction

## 1.1 General Introduction

Since the Industrial Revolution, the progress of humankind went along with a steady increase in primary energy consumption primarily from fossil energy sources. According to the motto “The most efficient energy is the one that is simply not needed!” many energy transition policies are targeting an increase in energy efficiency to reduce the usage of fossil fuels and ultimately mitigate the impact of the anthropogenic climate change. To keep the global warming below an average temperature increase of 2 °C which represents the declared objective of the United Nations it needs more than just a political declaration of intent. The recuperation of waste heat and its transformation into electrical energy represents one possibility to increase the efficiency in many sectors. It is therefore inevitable to develop new concepts that will allow energy harvesting in a sustainable manner. Over the past decade, one technology that has gained much attention in this field is the transformation of heat using thermoelectric generators (TEGs).

Different to the established waste heat recovery systems (WHRS) in which heat is first transformed in mechanical and afterwards into electrical energy, TEGs convert thermal energy directly to electricity via the Seebeck effect. New printed TEGs, which rely on organic materials represent a new trend in the TEG development. The Karlsruhe Institute of Technology (KIT) with a research group at the Light Technology Institute (LTI) and its spin-off company otego GmbH are research and development leaders in this field. While the largest potential for this new type of TEGs is currently seen in low power applications ( $\mu\text{W}$  to  $\text{W}$ ) the question arises, how organic based large surface TEGs can be utilized to recuperate larger amounts of waste heat in the  $\text{kW}$  to  $\text{MW}$  range.

Despite their relatively low conversion efficiencies, TEGs offer several clear advantages in comparison to classical WHRS using a mechanical intermediate transformation stage, such as:

- The lack of moving components along with low maintenance offer a high system reliability.
- A high tolerance towards fluctuating heat supply.
- Higher safety due to the lack of potentially hazardous working fluids.

Up to now the research interest in the field of thermoelectric materials has almost exclusively concentrated its efforts on inorganic compounds based on rhodium, platinum, copper,

bismuth and tellurium. Due to its good thermoelectric characteristics bismuth telluride ( $\text{Bi}_2\text{Te}_3$ ) represents the most commonly used material. Especially the rare earth metal tellurium and the complex production processes of inorganic TEGs get in the way with a sustainable development of this technology. The tellurium supply suffers from volatile resource prices and an unequal distribution of the resource, making the technology dependent on a few production countries. What is more, is the environmental impact when tellurium is exposed to the nature.

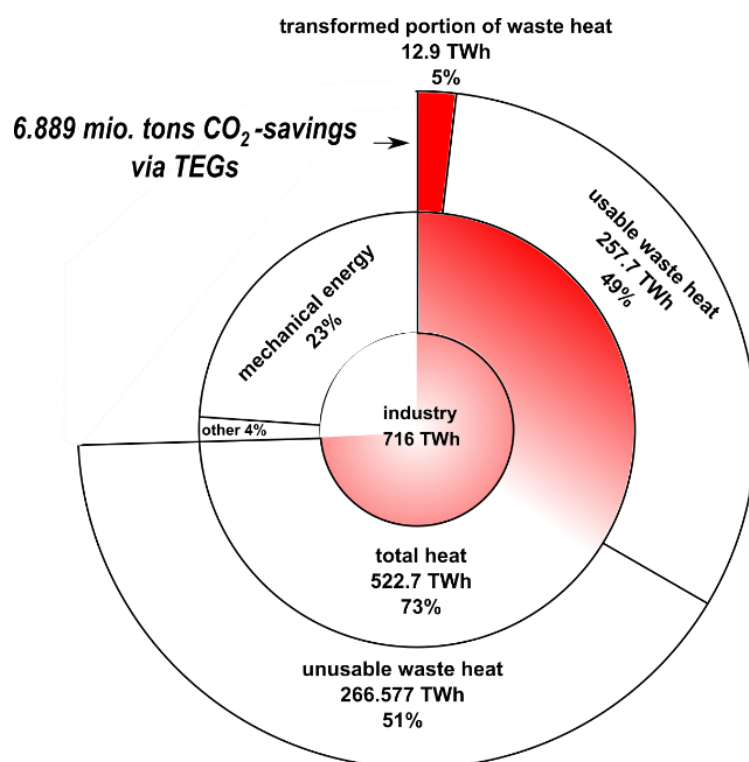
It is for these reasons that the current TEG research is turning towards the development of new, environmentally compatible, sustainable and inexpensive thermoelectric materials.

Organic thermoelectric Generators (OTEGs) represent hereby a promising alternative to conventional TEGs. The materials used are non-toxic, cost-efficient and widely accessible. The potential for an application of organic TEGs for large amounts of heat in the middle and low temperature range has so far been neglected mainly because of their relatively low conversion efficiencies. Nevertheless, from the recent progress in material synthesis as well as in production processes derive an opportunity and necessity for research in this field. Supposing that this new generation of printed TEGs allows a sustainable and economical heat recuperation for large amounts of waste heat, it can be expected that this technology might play a greater role in the transition towards a more sustainable energy sector.

## 1.2 Waste Heat Potential of the German Industry

Waste heat is a byproduct in a countless number of industrial processes. Alone in the year 2015, the industry in the federal republic of Germany accounted for 716 TWh (29 %) of the total final energy consumption of the country. In the same year three forth of the energy was used to provide process heat (65 %), room heat (7 %) and warm water (1 %). An examination from the year 2008 based on data provided by the statistical federal office quantifies the theoretical potential of the recoverable waste heat on 36 % of the final energy consumption in the industrial sector. This would mean an amount of 257.76 TWh per annum. Under the assumption that 5 % of this heat could be transformed into electricity, the generated energy would account for 12.9 TWh per year. Compared to the current emissions factor of the German electricity mix (534  $\text{g}_{\text{CO}_2}$  per kWh) the annual  $\text{CO}_2$  saving potential would add up to more than 6.88 million tons (**Figure 1**). Especially in the heavy energy intensive industry (steel, aluminum, cement and paper) the waste heat recovery potential can be tremendous. However, one of the key problems still is the lack of detailed information about

potential waste heat sources. According to the dena<sup>1</sup> (Deutsch Energie Agentur GmbH) every 3<sup>rd</sup> company in Germany is not aware of its own waste heat potential. It is therefore that a recent research project with the objective to create a consistent “Waste Heat Atlas” for Germany until 2018 is underway. As an interesting approach is seen the implementation of public platforms which provide information about waste heat sources and sinks. Some German federal states such as Bavaria<sup>2</sup> and Saxony<sup>3</sup> have already created platforms at which companies and local authorities offer their unused waste heat as a product and supply essential information (temperatures, heat amount, heat transfer medium, thermal power, operation hours). From a brief overview of the data provided by the platforms the vast majority of the waste heat is available as transient exhaust air below 220 °C. A WHRS developed for these conditions might have a mayor impact.



**Figure 1 | Industrial waste heat potential**

A consequente transformation of waste heat in the German industry using advanced printed TEGs that allow approximately 5 % conversion efficiency might allow CO<sub>2</sub> savings of 6.889 mio. tons per year. This is equivalent to the emissions of 4,039,876 cars in Germany with CO<sub>2</sub> emissions of 127.4 gkm<sup>-1</sup> and an average mileage of 13.385 kma<sup>-1</sup>.

## 1.3 Aims of this Work

Considering the new development in the field of thermoelectric, this work shall elaborate a first evaluation method of the potential for the power generation from industrial waste heat

<sup>1</sup> <https://industrienergieeffizienz.de/energiekosten-senken/energieeffiziente-technologien/abwaermenutzung/erfolgreiche-abwaermenutzung-im-unternehmen/>

<sup>2</sup> <http://geoportal.bayern.de/energieatlas-karten/>

<sup>3</sup> <http://www.saena.de/angebote/abwaermeatlas.html>

using a new generation of printed large surface organic based TEGs from a technological point of view.

To do so the work concentrates on the following research questions:

- “How can a WHRS based on printed TEGs be conceptionally realized?”
- “How can it be simulated?”
- “What maximal electrical output power can deliver the WHRS?”

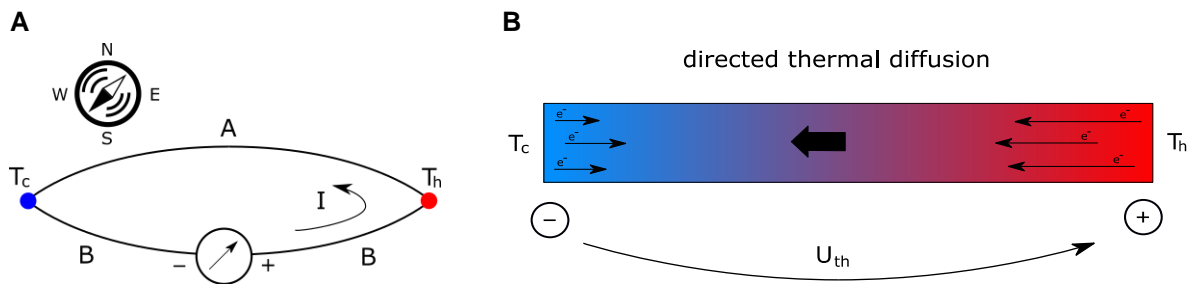
It will be the first step towards the proof-of-principle and the realization of a demonstrator and deliver results that might give a foundation for further techno-economic analysis.



## 2 Theoretical Background

### 2.1 Thermoelectric Effects

The discovery of the direct transformation from heat to electricity via the thermoelectric effect goes back almost 200 years. In 1821 Thomas Johannes Seebeck observed that a compass needle is deflected, when it comes close to a setup in which two different metallic conductors are joined in the order (B-A-B) and whose two junctions are held at different temperatures (**Figure 2, A**). The cause for this deflection lies in a magnetic field that is created due to a ring current flowing through the setup. This phenomenon is called Seebeck effect.



**Figure 2 | Seebeck effect**

Schematic setup of a thermocouple, in which the Seebeck effect occurs, **A**. Thermal diffusion of charge carriers in a material along whose length a temperature difference is applied, **B**.

The origin of the current flowing is the thermal diffusion of charge carriers in the materials. Considering first a simple metallic conductor whose both ends have a temperature difference. At the warm side, the thermal energy liberates more free electrons than on the cold side. This leads to a directed diffusion current from the hot side to cold side that is balanced out by the voltage difference  $U_{th}$  between the two ends (**Figure 2, B**).

The integrated thermal voltage of a thermocouple such as Thomas Seebecks setup where the temperature difference between the junctions is small can be described by the formula:

$$U_{th} = S_{AB} \cdot (T_h - T_c) \quad (1)$$

$S_{AB}$  describes hereby the thermoelectric force or Seebeck coefficient that is depending on the materials used. The Seebeck coefficient itself can be described using the two single Seebeck coefficients of the joined materials by:

$$S_{AB} = S_A - S_B \quad (2)$$

The unit of the Seebeck coefficient generally used is  $VK^{-1}$ . In case that the thermal current flows at the warmer end from the material A to the material B the Seebeck coefficient is positive. Using this convention electron conducting p-type semiconductors have a positive Seebeck coefficient whereas n-type semiconductors have a negative thermoelectric force. From relationship (2) it is evident that the thermal voltage gets particularly large when p-and n-type semiconductors are combined in a thermocouple.

The Peltier effect which describes the reversible Seebeck effect was observed 13 years after Seebecks discovery by Jean C. A. Peltier. Instead of applying a temperature difference at the two junctions Peltier demonstrated that a temperature difference occurs between the two junctions when current flows through the thermocouple. The amount of heat emitted per time by the junction is thereby proportional to the current  $I$  and the Peltier coefficient:

$$\frac{d}{dt}Q = \Pi_{AB} \cdot I \quad \text{with} \quad \Pi_{AB} = S_{AB} \cdot T \quad (3)$$

Besides the Peltier and the Joule heating there is a third heat that is emitted or absorbed along a current carrying conductor called the Thomson heat named after William Thomson (later Lord Kelvin). The heat over the current carrying conductor is described by the Thomson coefficient  $\tau$ , the current  $I$  and the temperature  $T$ . The Thomson coefficient itself can be described using the differential of the Seebeck coefficient  $S$  over the temperature  $T$  multiplied by the temperature  $T$ .

$$\frac{d}{dx}Q = \tau \cdot I \cdot \frac{dT}{dx} \quad \text{with} \quad \tau = T \cdot \frac{dS}{dT} \quad (4)$$

When analyzing a thermoelectric generator, it must be taken into account that all the presented thermoelectric effects occur and overlap each other in the generator. Because of the dominance of the Joule heat the Thomson effect is generally neglected [1].

## 2.2 Thermoelectric Materials

The TEG technology has its roots in the early aerospace research. Many of the materials which are found today in commercial TEGs were initially developed for aerospace applications. As energy source for satellites, rovers and other aerospace applications they have been used since the 1960s in radio isotopic thermoelectric generators (RTGs). Environmental compatibility and economical aspects have hereby been of minor importance.

The direct transformation of thermal energy into electrical energy via the Seebeck takes place in the solid-state thermoelectric material of a TEG. In the literature, the thermoelectric

performance of a material is in general quantified by the dimensionless figure of merit  $ZT$ . This figure is composed of the Seebeck coefficient  $S$  ( $\text{VK}^{-1}$ ), the electrical conductivity  $\sigma$  ( $\text{Sm}^{-1}$ ), the thermal conductivity  $\kappa$  ( $\text{WK}^{-1}\text{m}^{-1}$ ) and the temperature  $\bar{T}$  (K) as follows:

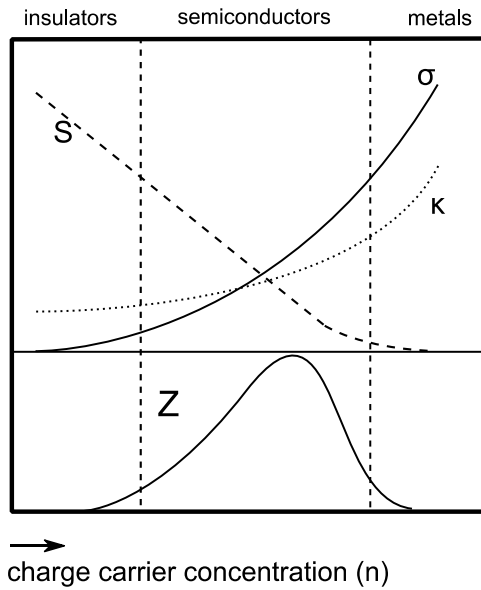
$$Z\bar{T} = \frac{S^2 \cdot \sigma}{\kappa} \cdot \bar{T} \text{ with } \bar{T} = \frac{(T_h + T_c)}{2} \quad (5)$$

The relation between the figure of merit and the conversion efficiency was described by Edmund Altenkirch:

$$\eta_{max} = \eta_{Carnot} \cdot \frac{\sqrt{1 + Z\bar{T}} - 1}{\sqrt{1 + Z\bar{T}} + \frac{T_c}{T_h}} \text{ with } \eta_{Carnot} = 1 - \frac{T_c}{T_h} \quad (6)$$

In this relation,  $Z\bar{T}$  is replaced by  $\overline{ZT}$  the integral mean value of  $ZT$  between the temperatures  $T_c$  and  $T_h$ . This allows a more accurate description for large temperature gradients since the temperature dependence of the material parameters ( $S(T)$ ,  $\sigma(T)$  and  $\kappa(T)$ ) are included. Note, whenever  $ZT$  is mentioned in this work it refers to the integral mean value  $\overline{ZT}$ .

Altenkirch's equation implies the larger  $ZT$  the greater is the transformed amount of thermal energy into electrical energy. Therefore an optimal thermoelectric material should have a high Seebeck coefficient  $S$ , a high electrical conductivity  $\sigma$  and a small thermal conductivity  $\kappa$ . **Figure 3** illustrates these properties of different material groups. According to this illustration the best thermoelectric materials are found in the group of semiconductors and metalloids. The search for new materials with improved thermoelectric properties of inorganic and organic nature represents a significant part of the current TEG research efforts.



**Figure 3 | Thermoelectrical parameters  $S$ ,  $\kappa$  and  $\sigma$**

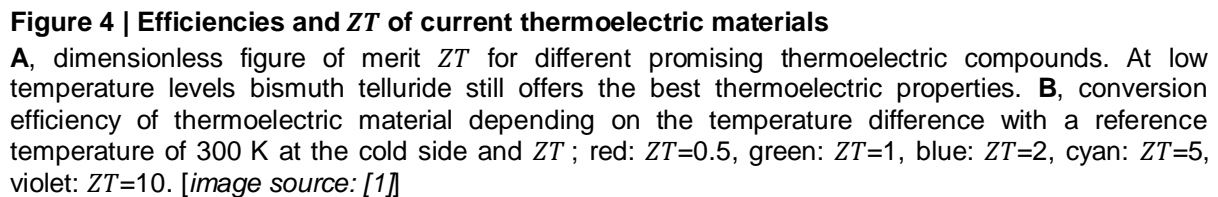
Comparison of Seebeck coefficient  $S$  ( $\text{VK}^{-1}$ ), the electrical conductivity  $\sigma$  ( $\text{Sm}^{-1}$ ) and the thermal conductivity  $\kappa$  ( $\text{WK}^{-1}\text{m}^{-1}$ ) for different groups of materials. Insulators and metals do not qualify for TEGs due to low electrical conductivity in the case of insulators and high thermal conductivity for metals. Semiconductors and semimetals show the best thermoelectric qualities. [image according to: [2]]

### 2.2.1 State of Technology: Thermoelectric Materials

In the literature, conventional TEGs are seen critical regarding their significance for the energy transition. For example, expresses Patyk strong doubts on the sustainability of classical TEGs and sees only very little influence on a sustainable future energy sector. This is mainly due to the usage of rare earth metals and heavy metals in the classical TEGs [3]. Vining goes even further with his criticism, saying that only a  $ZT$  value of 4 would justify a potential use for industrial waste heat [4]. In his paper though Vining simply compares the conversion efficiencies of different waste heat recovery systems and neglects the significance that TEG based systems offer: higher reliability, higher safety and their high tolerance dealing with fluctuating heat supply. An investigation of the sustainability of the new generation of TEGs considered in this work has not yet been done.

Since around 2005 there is an increasing interest for the use of TEGs in the field of commercial applications. This has led to the establishment of a couple young companies that specialize on TEGs and actively push forward research and development in this field. Emerged from these endeavors are new thermoelectric materials that are, ecologically compatible and can be produced in a sustainable and cost-efficient manner. The TEG technology currently finds itself on the brink of leaving their niche markets and becoming interesting for mass applications.

In the area of inorganic TEGs new modules based on magnesium silicate ( $\text{Mg}_2\text{Si}$ ), calcium and manganese oxides, Heusler alloys and scutterudites will become available in the near future. These modules will be more cost efficient and will have a lower environmental impact [5]. However, these materials will first be considered for high temperature applications, here they demonstrate high  $ZT$  values. For temperature below  $200\text{ }^\circ\text{C}$  bismuth telluride still shows the highest conversion efficiencies (**Figure 4, A**).

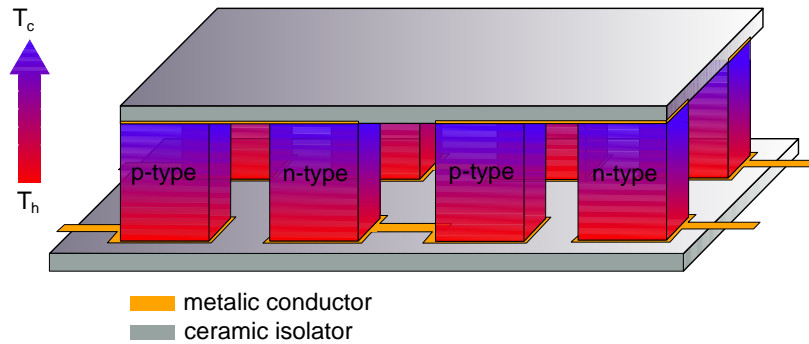


At the KIT, the enhancement to a  $ZT$  of 0.5 is seen as a realistic goal in the medium-term. This assumption is based on the recent progress in the field of material synthesis at the KIT and the otego GmbH. To achieve this, hybrid materials are synthesized in which printable organic and inorganic compounds are combined. Assuming an operation of an TEG with  $ZT = 0.5$  at a medium temperature level with  $T_h = 150\text{ °C}$  und  $T_c = 20\text{ °C}$  using the relationship **(6)** would give a conversion efficiency of 6.4 %.

In a thermoelectric generator, the individual thermocouples transform a proportion of the heat that is conducted through them into electricity. To scale up the output voltage, a TEG consists in general of a great number of thermocouples. Therefore, the individual thermocouples are electrically connected in series while the heat transfer happens in parallel. For a TEG in which the heat is evenly transferred over its surface the output voltage can be calculated by multiplying **(1)** with the number of thermocouples in the TEG:

$$U_{th} = n \cdot S_{AB} \cdot (T_h - T_c) \quad (7)$$

For the TEG to work properly next to the electrical management the heat management is of immense importance. As soon as the TEG reaches a thermal equilibrium the output voltage of the TEG drops to zero. It is therefore of great importance to design the heat source and the heat sinks in a manner to maintain the temperature difference at a maximum in the generator.

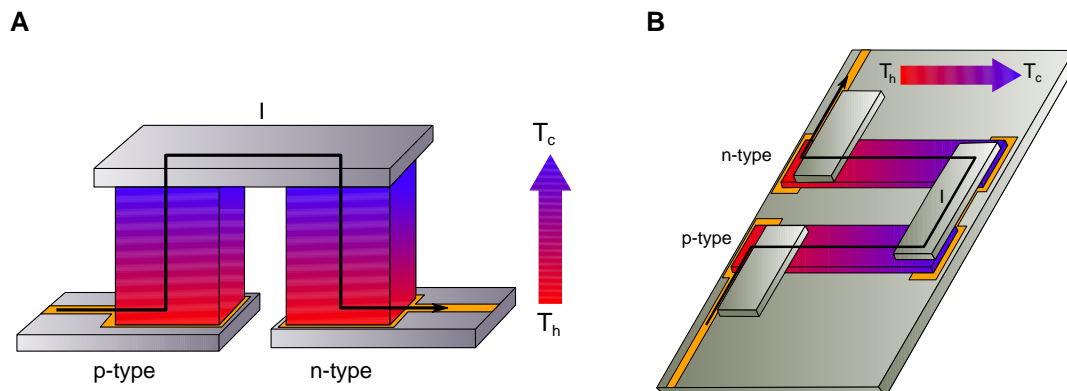


**Figure 5 | Classical TEG design**

To increase the output power of the TEG, the individual thermocouples are electrically connected in series and thermally in parallel.

### 2.3.1 Printed TEGs

A new generation of TEGs which build the foundation for this work are printed TEGs based on organic and inorganic materials. For this type of generator, the classical construction method with its bulk rods does not apply (**Figure 6**). An alternative concept is illustrated in **Figure 6, B**. On the contrary to the classical design the heat transfer is not perpendicular to the substrate but rather parallel to it. This alternative approach allows a layer based design suitable for printing processes.

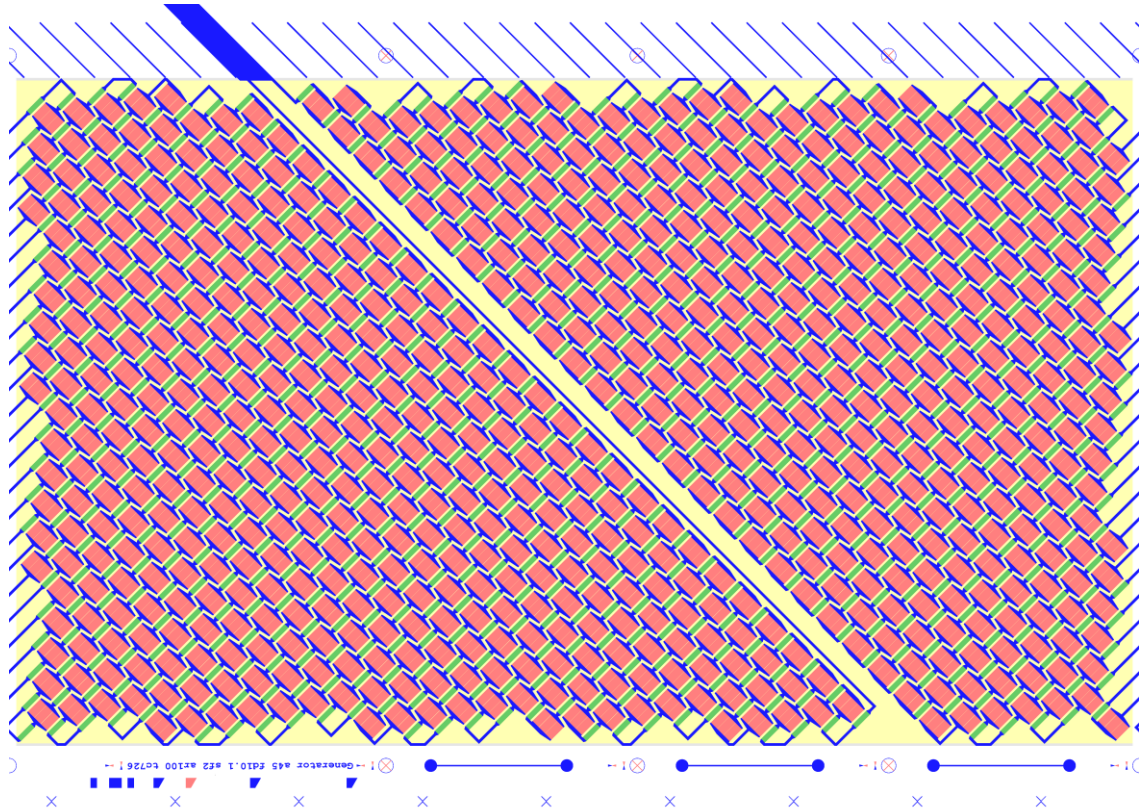


**Figure 6 | Thermocouple design**

**A**, classical setup of a thermocouple with two bulk feet of p- and n-type semiconductor material on a ceramic substrate. The temperature gradient is perpendicular to the substrate. **B**, alternative setup of a thermocouple, the flat design is better suited for an application of organic and solution based

materials. It allows an application using scalable printing technologies. *[image according to: [7]]*

At the Light Technology Institute (LTI) of the Karlsruhe Institute of Technology (KIT) an automated process to produce a new printed OTEG was elaborated and patented in 2013 [8]. In a role-to-role printing process, the thermocouples are applied on a thin foil and connected in series (**Figure 7**).

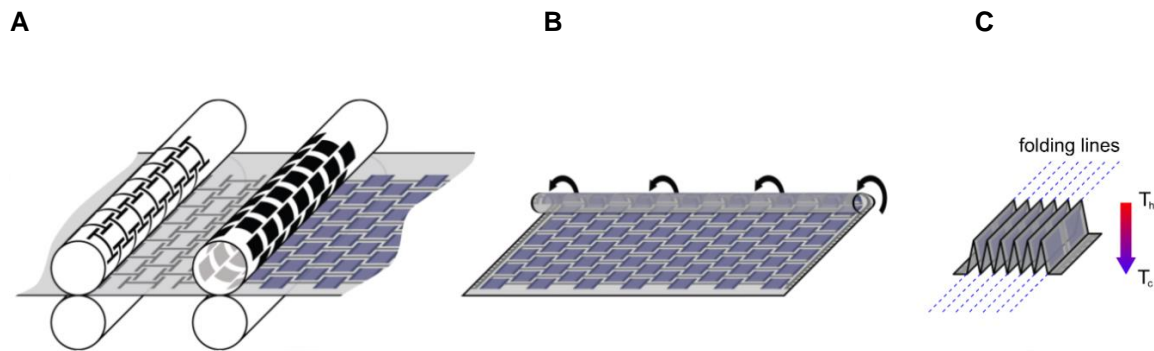


**Figure 7 | Printing layout**

Every thermocouple is made up of three p-type semiconductor legs (red), one n-type semiconductor (green) and connected with an electrical conductor (blue). The blue stripes on the edge mark the folding lines to arrange the thermocouple thermally in parallel. *[image source: [9]]*

In several post-processing steps, the foil is rolled and folded to produce the generators. The method combines cost efficient printable materials and scalable production processes and is therefore well suited to produce large surface TEGs.

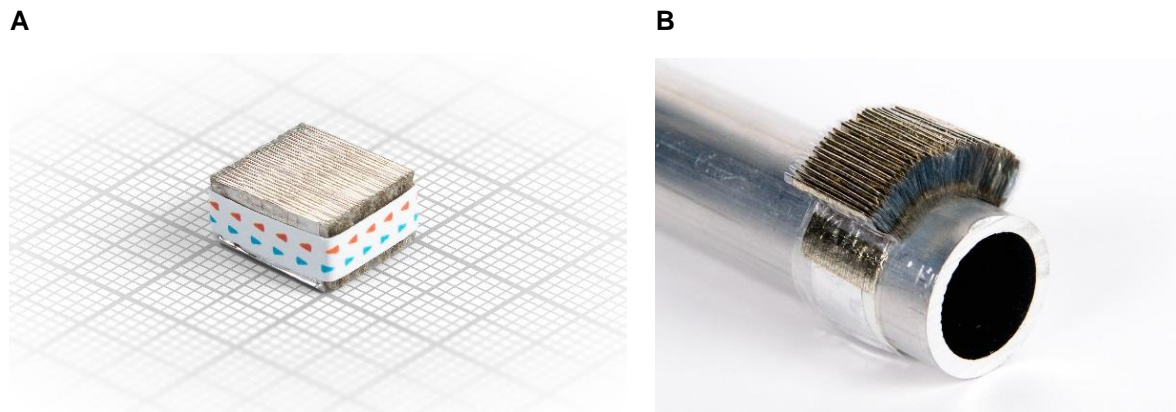




**Figure 8 | Production process of a printed OTEG (Patent holder KIT)**

**A**, From left-to-right: flexography (silver), screen printing (PEDOT:PSS), **B** rolling process, **C** folding process

Emerged from this development at the Light Technology Institute is a young spin-off company: the otego GmbH<sup>4</sup>. The start-up specializes on the development and production of printable TEGs that are non-toxic, cost efficient, bendable and can have various shapes and sizes. These aspects generally do not apply for classical TEGs.



**Figure 9 | OTEGs**

**A**, a ready-for-use OTEG for low power applications. The sugar cube sized generator can deliver a voltage of 1 V and will be the first TEG brought to market by the otego GmbH. **B**, on the contrary to the classical bulk TEG design, the printed TEGs are flexible and bendable. This ability opens an immense new spectrum for applications. [image source: <http://www.otego.de/en/>]

### 2.3.2 State of Technology: Power Generation

While there are already first commercial products based on classical TEGs for high temperature applications in the field of heat recovery and power generation soon to be brought to market, there is no industrial or academic development present that looks at the potential of printed TEGs for the recuperation of large amounts of waste heat in the medium and low temperature range. Until now printed OTEGs have only attracted interest for low power applications.

<sup>4</sup> <http://www.otego.de/en/>



Different scientific work and industrial studies conclude that TEGs in the medium and low temperature range offer two promising perspectives for the recuperation of industrial waste heat. On the one hand, when waste heat of a process is difficult to capture (radiation) and on the other hand, when by large surface TEGs, maintenance free and cost efficient electrical energy can be supplied [5]. In a previous work at the KIT interviews were conducted with experts in the field of waste heat recuperation. These interviews showed that there will be a growing demand for WHRS that need to work reliably under fluctuating operating conditions [10]. This demonstrates the clear necessity for the research conducted over the course of this thesis work.

A system which transforms radiation and convection heat of a metal casting process using  $\text{Bi}_2\text{Te}_3$ -TEGs was investigated by Børset et al. Based on an experimental setup, measurements under transient conditions were conducted and a mathematical model elaborated. The measurements demonstrated that the module delivered a peak power output of  $160 \text{ Wm}^{-2}$  with a temperature gradient of 100 K. From the results, it has been concluded that by changing the position of the system and by altering the TEG design a significant optimization up to  $900 \text{ Wm}^{-2}$  could be achieved [11].

Beyond that many papers investigate the techno-economical potential of WHRS based on TEGs compared to organic Rankine cycles (ORCs). Such a comparison elaborated Felgner et al. in 2012. Two exemplary models in which waste heat of biogas plant was transformed via an ORC system and via bismuth telluride TEGs were simulated over a life span of 25 years and compared. The finding of the paper was that the capital present value of the systems was very similar after the 25 years [12]. Regarding the cost reduction, which large surface printed OTEGs offer, a techno-economical comparison might turn out to be in favor of this new generation of TEGs.

## 2.4 Fundamentals of Heat Transfer

When working in the field of thermoelectricity an understanding of the fundamental heat transfer mechanisms is of immense importance. These mechanisms ultimately dictate the design of the TEG and the entire WHRS. For this reason, this section will give a brief overview of the two basic heat transfer mechanisms: radiation and conduction. In the part of conduction, the effect of convection will be included.

In general heat is the amount of energy transferred in-between two systems or the system and its surroundings by means of a temperature difference. The total amount of heat is composed of a superposition of the individual heat transfer mechanisms:

$$\dot{Q}_{tot} = \dot{Q}_{radiation} + \dot{Q}_{conduction \text{ or } convection} \quad (8)$$

### 2.4.1 Conduction and Convection

Conduction occurs between two bodies at rest by direct contact when a temperature gradient exists. According to Fourier the heat flow in a homogenous medium is proportional the temperature gradient. For many applications including this work the Fourier law in its one-dimensional form can be applied:

$$\dot{Q}_{cond} = -\lambda \cdot A \cdot \frac{dT}{dx} \quad (9)$$

The thermal conductivity  $\lambda$  has the dimension  $\text{Wm}^{-1}\text{K}^{-1}$  and  $A$  describes the surface of heat transfer in  $\text{m}^2$ . The minus indicates that the heat is always flowing in the direction of falling temperatures. The effect of conduction has a direct influence on this work; Fourier's law is used to describe the temperature changes in solids.

A special case of the conduction describes the convection. Different to conduction at least one medium involved in the heat transfer is not at rest. It occurs when liquid or gaseous mediums move from one region with a temperature to a region with a different temperature. Convection itself is split into natural or rather free convection and forced convection.

Free convection occurs when the flow is driven by internal forces resulting for example from density differences or gravitational forces. To forced convection is referred when external forces are applied by equipment such as fans, pumps, blower, etc.

In a convective heat transfer event, conduction is always present. When imagining a fluid passing a fixed wall due to adhesive friction between the fluid particles close to the wall surface, a small layer is formed where conduction dominates. The macroscopic heat transfer between a fluid and a fixed wall can be expressed via Newtons law of cooling:

$$\dot{Q}_{conv} = \alpha \cdot A \cdot (T_{surf} - T_{\infty}) \quad (10)$$

Newton's law basically represents a restatement of Fourier law. In it the heat transferred by convection is related with the exchanger surface  $A$  ( $\text{m}^2$ ), the temperature of surface  $T_{surf}$  (K), the temperature of the fluid at an infinite distance to the surface  $T_{\infty}$  (K) and the convective heat transfer coefficient  $\alpha$  ( $\text{WK}^{-1}\text{m}^{-2}$ ). The heat transfer coefficient  $\alpha$  depends besides on fluid parameters (density, kinetic Viscosity, heat conductivity) strongly on the type of flow (laminar, transition zone, turbulent) with which the fluid is passing the wall. It is common practice to

use dimensionless quantities such as Nusselt number ( $Nu$ ), Reynolds number ( $Re$ ) and the Prandtl number ( $Pr$ ) to describe this event with empirical equations. In heat exchangers, this relation between heat transfer and flow conditions is fundamental. The greater the disturbance of the flow, the smaller is the resistance by the convective heat transfer.

Forced convection plays a major role in this work. In the following section about similitude the individual dimensionless quantities:  $Nu$ ,  $Re$  and  $Pr$  that quantify the convective heat transfer coefficient are examined in greater detail.

### 2.4.2 Radiation

Each body with a temperature above 0 K emits heat in form of electromagnetic radiation depending on its temperature and the nature of its surface. The heat radiation can be described by:

$$\dot{Q}_{rad} = \sigma \cdot e_{surf} \cdot A_{surf} \cdot T_{surf}^4 \quad (11)$$

Here  $\sigma$  represents the Stefan-Boltzmann constant which is of the order of  $5.67 \times 10^{-8} \text{ Wm}^{-2}\text{K}^{-4}$ .  $e_{surf}$  describes the emissivity of the emitting surface,  $A$  describes the surface area and  $T$  the surface temperature. From this equation, it is directly apparent that the heat radiation strongly depends on the temperature of the surface. Since the scope of this work is limited to waste heat of medium to low temperature levels the radiated heat can be considered a minor effect and is neglected.

## 2.5 Computational Fluid Dynamics (CFD)

To solve a fluid dynamical problem means to determine the following unknowns:

1. density  $\rho$
2. velocities  $u_x, u_y, u_z$
3. pressure  $p$
4. temperature  $T$  or inner energy  $e$  or enthalpy  $h$

In a 3-dimensional case there are therefore six unknown variables that need to be solved using the following equations:

- I. mass conversation or continuity equation
- II. momentum conversation

III. energy conversation

IV. equation of state

The momentum conversation is a vector equation in 3-dimension with three individual momentum equations. With six equations for six unknowns a solution of the fluid dynamical problem is possible. In the field of fluid dynamics, the three partial differential momentum equations are the so called Navier-Stokes equations (NSE). To reduce the complexity of these equations simplifications for special assumptions are made and parts of the equations that only have little influence on the solution are neglected. In this study, the fluids are considered as single phase, incompressible ( $\rho=\text{const.}$ ), isothermal ( $T=\text{const.}$ ) with a constant molecular viscosity  $\mu$ . For these assumptions, the number of unknowns reduces to three velocities in x-, y-, and z-direction and a pressure term. The energy conversation and equation of state can therefore be disregarded. In addition, steady state conditions are assumed which eliminates the time dependent terms. This means that from the complex structure of the base equations:

$$\text{I.} \quad \underbrace{\frac{\partial \rho}{\partial t}}_{\text{temporal change in the control volume}} + \underbrace{\vec{\nabla}(\rho \vec{u})}_{\text{flow across control volume borders}} = 0 \quad (12)$$

$$\text{II.} \quad \underbrace{\frac{\partial(\rho \vec{u})}{\partial t}}_{\text{temporal change in the control volume}} + \underbrace{\vec{\nabla}(\rho[\vec{u} \otimes \vec{u}])}_{\text{flow across control volume borders}} = \underbrace{-\vec{\nabla} p}_{\text{pressure force}} + \underbrace{\vec{\nabla} \tau}_{\text{friction force}} + \underbrace{\rho \vec{f}}_{\text{external field forces}} \quad (13)$$

$$\text{III.} \quad \underbrace{\frac{\partial(\rho(e + \frac{1}{2}u^2))}{\partial t}}_{\text{temporal change in the control volume}} + \underbrace{\vec{\nabla}(\rho \vec{u}(e + \frac{1}{2}u^2))}_{\text{flow across control volume borders}} = \underbrace{-\vec{\nabla}(p\vec{u})}_{\text{pressure power}} + \underbrace{\vec{\nabla} \tau(\vec{u})}_{\text{friction power}} + \underbrace{\rho \vec{u} \vec{f}}_{\text{external field forces power}} - \underbrace{\vec{\nabla} \vec{q}}_{\text{heat transfer}} + \underbrace{\dot{Q}_s}_{\text{heat source}} \quad (14)$$

Only the following form of the equations are solved in this work:

$$\text{I.} \quad \nabla \vec{u} = 0 \quad (15)$$

$$\text{II.} \quad \rho \vec{u}[\vec{\nabla} \otimes \vec{u}] = \rho \vec{f} - \vec{\nabla} p + \mu \Delta \vec{u} \quad (16)$$

Additionally, to the equations boundary conditions need to be set on the region for which a solution shall be calculated. The differentiation of two types of boundary conditions is generally sufficient:

1. Dirichlet condition (determination of the value at the surface of the calculation region)

2. Neuman condition (determination of the gradient of the value at the surface of the calculation region)

In fluid dynamics, boundary conditions are generally defined for walls, inlets, outlets and symmetry. Usually the flow velocity is defined at the inlet directly (Dirichlet) while at the outlet most of the time the assumption is acceptable that the flow is fully developed which sets the gradients to zero (Neuman). At walls due to the non-slip condition the relative velocities are set to zero (Dirichlet). If the flow is symmetrical the computation time can be considerably reduced by using symmetry boundary conditions. It is then only necessary to simulate a part of the flow.

At a specific Reynolds number ( $Re$ ) there appear turbulent fluctuations in flows. Turbulent flows are generally specified by the following characteristics:

- unsteady
- 3-dimensional
- the turbulence enhances mixing by turbulent diffusion
- dissipation: in the smallest turbulent vortices, the kinetic energy is transformed irreversible into inner energy
- in turbulence phenomenon's there are both partially chaotic as well as statistical structures present

In fluid dynamics, there are three basic approaches to simulate the turbulence in a fluid: Direct numerical simulation (DNS), large eddy simulation (LES) and the Reynolds-averaged Navier-Stokes equations (RANS). Using the direct numerical simulation (DNS), the basic equations (12) and (13) are solved directly without further simplifications. This implies however a resolution in time and place that even considers the smallest turbulence event which makes extreme demands on memory and calculation time. The basic idea behind the large eddy simulation (LES) is to solve the large-scale turbulent events directly and model the rest. The most common methodology to simulate turbulence however is to solve the Reynolds-averaged Navier-Stokes equations using a turbulence model. Here the turbulence is modeled to 100 %. In the RANS equations, every flow parameter  $\phi$  is defined as its mean value  $\bar{\phi}$  and fluctuation term due to the turbulence  $\phi'$ . Since the RANS equations do not deliver a coherent solution on their own, different turbulence models are used to determine the unknown terms. Most turbulence models hereby define an extra turbulent viscosity  $\mu_t$ , which is different to the molecular viscosity not a material parameter but rather a flow parameter that may vary in time and place. The most widely used model, including this study is the k- $\epsilon$  model which implies that there is an equilibrium between the production and the

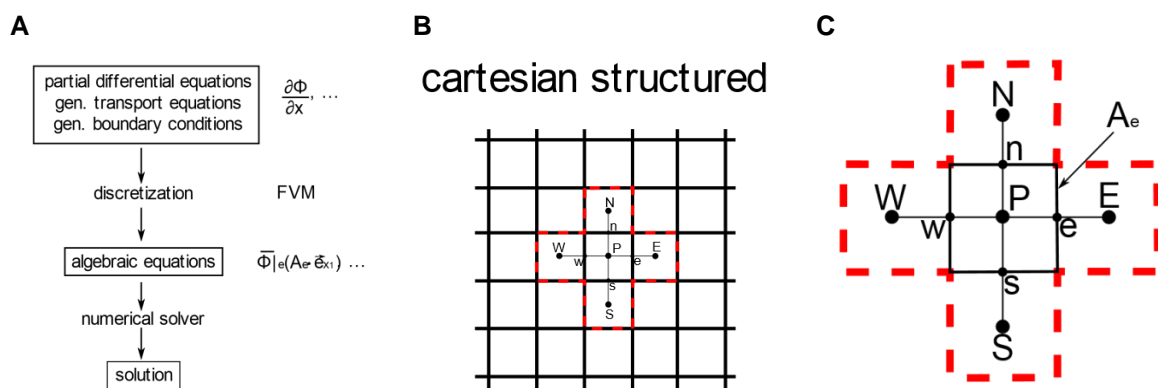
dissipation of the turbulent vortices. The characteristic parameters in this model are the kinetic turbulent energy  $k$  and its dissipation rate  $\varepsilon$ . For these two parameters boundary conditions must be declared additionally.

**Table 1 | Turbulence simulation**

	DNS	LES	RANS
<b>equations</b>	NS-equations	NS- and RANS equations	RANS equations
<b>degree of modeling</b>	0 %	0 % to 100 %	100 %
<b>calculation expenditure</b>	very high	high	low
<b>precision</b>	exact solution	between DNS and RANS	depending on the model used
<b>typical Re numbers</b>	low	high	high
<b>application</b>	research, validation and calibration of models	mainly research	industry

The presented equations and its boundary conditions only allow under special conditions an analytical solution. In most cases the solution must be determined computationally using a numerical method. This however requires that the differentials in the transport equations and the boundary conditions are replaced by algebraic terms (discretization). To do so prior to this step in the calculation region discrete points need to be defined where the information about the flow parameters of the cell is stored.

This is done by creating a mesh that specifies discretization points either on the cells edges or inside the cell. The most common discretization method in the field of fluid dynamics is the finite volume method (FVM). Its advantage lies in the conservativity of the equations when the method is applied correctly.

**Figure 10 | Computational fluid dynamics**

A, working principle of the numerical solving method. B, discretized mesh with a cartesian structured cell type. C, discretization of the finite volume method.

The basic principle behind FVM method is that conservation equations are integrated over the control volume of the cells. This is usually done using the gauss integral theorem with which the control volume can be expressed by its surfaces. The individual terms of the equations are then presented in an algebraic form, here exemplarily for a flow parameter  $\Phi$ :

$$\int_{A_e} (\rho \vec{u} \Phi) d\vec{A}_e = \underbrace{\overline{(\rho \vec{u} \Phi)}}_{\text{interpolation term}} |_e \left( \underbrace{A_e}_{\text{absolute value}} \vec{e}_x \right) \quad (17)$$

The equations for every cell defined in the flow region are then summarized in a linear system of equations in which  $A$  describes the coefficient matrix,  $\vec{X}$  the solution vector and  $\vec{b}$  the source terms:

$$A\vec{X} = \vec{b} \quad (18)$$

If an approximate solution  $\vec{X}^*$  is found the residual  $\vec{r}$  vector which allows a qualitative assumption of the convergence and the accuracy made by the solution can be determined:

$$\vec{r} = \vec{b} - A\vec{X}^* \quad (19)$$

Generally, the objective of a solution is to minimize the residual vector because this indicates that the mistake drops as well. While for simple flows where no fluctuations occur the monitoring of the residual is a good method to conclude the progress of the simulation. For unsteady flow conditions, it is more useful to monitor flow parameters directly and determine if they converge or show reasonable values.

## 2.6 Similitude

The operation behavior of a waste heat recovery system is defined by a large quantity of parameters depending on the working fluids, the system geometry, environmental influences and operation conditions. An engineering concept that aims to reduce this complexity by introducing (dimensionless) universal quantities is called similitude or similarity theory. Especially in the field of fluid dynamics and heat transfer this methodology is often applied. Over the course of this work several dimensionless parameters were used to describe different heat transfer or fluid dynamical events. These quantities are introduced in the following:

$$\text{Reynolds number} \quad Re = \frac{\bar{u} \cdot L_{char}}{\nu} \quad (20)$$

The Reynolds number can be interpreted as the ratio of friction and momentum forces, the lower the  $Re$  number the more dominant are the friction forces. In the field of fluid dynamics, the  $Re$  is generally used to determine the condition of a flow whether it is laminar or turbulent.



$$\text{Nusselt number} \quad Nu = \frac{\alpha \cdot L_{char}}{\lambda} \quad (21)$$

The Nusselt number quantifies the heat transfer between a wall and a fluid that is moving compared to a fluid that is at rest or simply the influence of convection compared to pure conduction. For common geometries such as pipes, ducts, etc. there exist empirical equations that determine  $Nu$ . In these equations  $Nu$  is often expressed as a function of  $Re$  and  $Pr$ .

$$\text{Prandtl number} \quad Pr = \frac{\nu}{\kappa} = \frac{\nu \cdot \rho \cdot c_p}{\lambda} \quad (22)$$

The Prandtl number is a dimensionless fluid parameter that relates the velocity field of a fluid with its temperature field. It is best understood when imagining the temperature development between a wall and fluid. The kinematic viscosity hereby determines the thickness of the velocity boundary layer while the heat conductivity, the density and the specific heat capacity are merged into a coefficient describing the thermal diffusivity of the fluid. The temperature conductivity determines the thickness of the temperature boundary layer between fluid and wall.

$$\text{Number of Transfer Units} \quad NTU = \frac{\alpha_{tot} \cdot A}{\dot{m} \cdot c_p} = \frac{A}{R_{tot} \cdot \dot{m} \cdot c_p} \quad (23)$$

The Number of Transfer Units is a dimensionless parameter used for dimensioning and calculation purposes of heat exchangers. It combines fluid parameters (specific heat capacity) with geometric parameters (heat exchanger surface, thermal resistivity) and the operational parameters (mass flow rate). This simplifies the equations of heat exchangers and brings it into a clear form (see (24), (25), (26) and (27)).

**Table 2 | Parameters of universal quantities**

$\bar{u}$	mean fluid velocity	$[\text{ms}^{-1}]$
$L_{char}$	characteristic length	$[\text{m}]$
$\nu$	kinematic viscosity	$[\text{m}^2\text{s}^{-1}]$
$\alpha$	heat transfer coefficient (convection)	$[\text{WK}^{-1}\text{m}^{-2}]$
$\lambda$	thermal conductivity	$[\text{Wkg}^{-1}\text{m}^{-1}]$
$A$	heat exchanger surface	$[\text{m}^2]$
$c_p$	specific heat capacity at constant pressure	$[\text{Jkg}^{-1}\text{K}^{-1}]$
$\rho$	density	$[\text{kgm}^{-3}]$
$k$	heat transmittance coefficient	$[\text{W}^{-1}\text{Km}^2]$
$\dot{m}$	mass flow rate	$[\text{kgs}^{-1}]$
$R_{tot}$	total thermal resistance	$[\text{Km}^2\text{W}^{-1}]$
$\kappa$	thermal diffusivity	$[\text{m}^2\text{s}^{-1}]$

## 2.7 Heat Exchangers

Before waste heat can be used at a different location, it must be made available. This task fulfil heat exchangers which by doing so represent the fundamental element of any WHRS. Heat exchanger can be classified in recuperators and regenerators. Relevant for this work are the recuperators in which the working fluids are spatially separated from each by a heat transferring surface. Another classification follows according to the flow directions of the working fluids: counter flow, parallel flow, hybrid flow and cross flow. For steady state operation, the temperature development of the heat exchanger can be derived in an analytical form from the 1<sup>st</sup> principle of thermodynamics. For parallel flow and counter flow conditions the temperature development over the streaming length (**Figure 11**) of the fluids can be described by:

$$T_{\infty} = \frac{T_1^0 - T_2^0}{\frac{NTU_2}{NTU_1} + 1} + T_2^0 \quad (24)$$

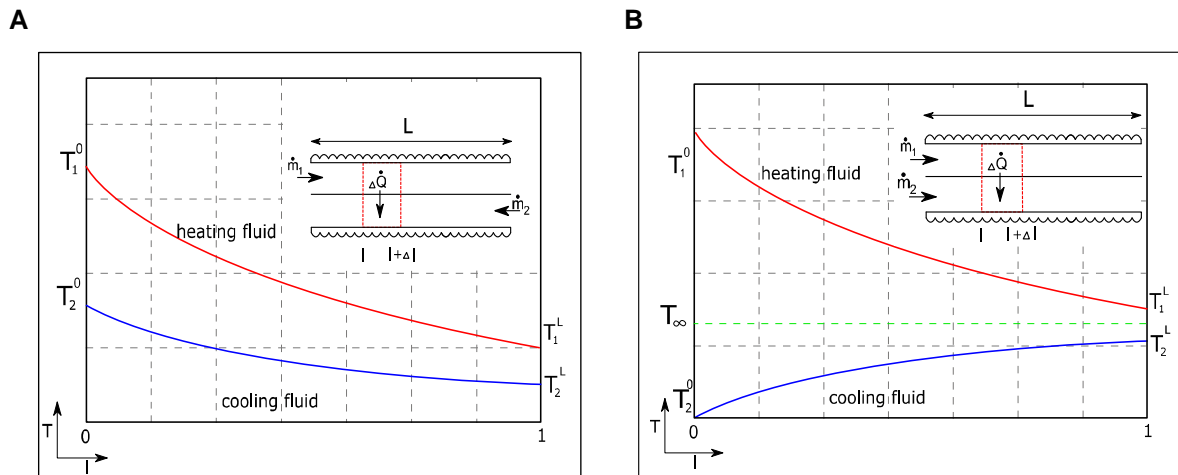
(parallel flow)

$$\frac{T_1(l) - T_{\infty}}{T_1^0 - T_{\infty}} = \frac{T_2(l) - T_{\infty}}{T_2^0 - T_{\infty}} = e^{-(NTU_1 - NTU_2)\frac{l}{L}} \quad (25)$$

$$\frac{T_1(l) - T_1^0}{T_2^L - T_1^0} = \frac{NTU_1}{NTU_2} \frac{1 - e^{-(NTU_1 - NTU_2)\frac{l}{L}}}{\frac{NTU_1}{NTU_2} - e^{-(NTU_1 - NTU_2)}} \quad (26)$$

(counter flow)

$$\frac{T_2^L - T_2(l)}{T_2^L - T_1^0} = \frac{e^{-(NTU_1 - NTU_2)\frac{l}{L}} - e^{-(NTU_1 - NTU_2)}}{\frac{NTU_1}{NTU_2} - e^{-(NTU_1 - NTU_2)}} \quad (27)$$



**Figure 11 | Temperature development in heat exchangers**

Temperature profile over the streaming length of a parallel (**B**) and counter (**A**) flow heat exchanger. The profiles of the temperature is described by the equations (24), (25), (26) and (27).

## 2.8 Software

### 2.8.1 Creo Parametric 3.0

*Creo Parametric 3.0*<sup>5</sup> is a commercial software for CAD modelling. In this sector, it delivers one of the most powerful and widely used platforms. For students and academical employees there exists a campus license agreement with PTC, the company that develops Creo. Models created with the academical license are not compatible with commercial versions of Creo Parametric. What's more is that the different versions are not necessarily upward and downward compatible.

### 2.8.2 Salome

*Salome*<sup>6</sup> is an open-source software that specializes on the pre- and post-processing of numerical simulations. For this work, it was found to offer powerful utilities that were useful prior to the meshing phase. While *Creo Parametric 3.0* was used to create the CAD model, Salome offered far more options to create a high-quality stereolithographic mesh from the CAD geometry (**Figure 25**). Salome can also be used as stand-alone software to generate CAD models.

### 2.8.3 Simulink

*Simulink*<sup>7</sup> is an add-on for *MATLAB* that allows a hierarchical modelling with graphical blocks. The direct connection with *MATLAB* was found to be the big advantage of this software, it allows to solve problems that cannot be solved by *Simulink* on its own. For this work *Simulink* was used to create a thermal model of a TEG based WHRS.

### 2.8.4 OpenFOAM

*OpenFOAM*<sup>8</sup> stands for *Open Source Field Operation and Manipulation*. In principle, it is a C++ library for solving partial differential equations. The objective of the software is the creation of executable applications particularly for the simulation of fluid dynamical problems. It provides a wide range of solvers, to solve the differential equations of physical problems

---

<sup>5</sup> <https://www.ptc.com/en/cad/creo/parametric>

<sup>6</sup> <http://www.salome-platform.org/>

<sup>7</sup> <https://de.mathworks.com/products/simulink.html>

<sup>8</sup> <http://www.openfoam.com/>; <https://cfdirect.com/>

including utilities for pre- and post-processing. For this work the release *OpenFOAM 4.1* was used.

### 2.8.5 gnuplot

*gnuplot*<sup>9</sup> is a script- and command line based program to generate plots of functions, data and data fits. It is a free software that is preinstalled on many Linux systems. Over the course of this work the program was used to generate plots for visualization purposes but also to generate polynomial function fits of velocity profiles to set initial conditions in upcoming simulations.

---

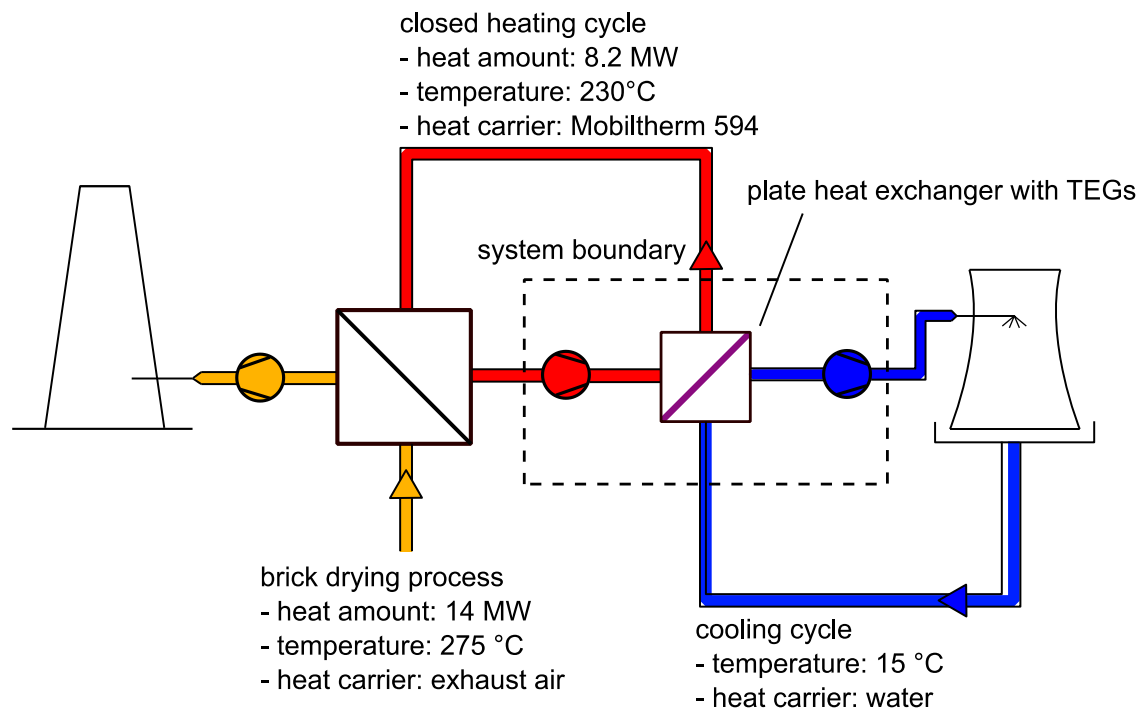
<sup>9</sup> <http://www.gnuplot.info/>

## 3 Methodology

### 3.1 Case Scenario

To demonstrate the applicability of the developed WHRS the operational conditions should be dictated by a real process. It was found that there are very few studies accessible addressing the waste heat potentials of industrial processes in a satisfactory manner. For this work a study from 2001, evaluating the waste heat potential of a brick drying process in a cement factory, was taken. It provides a detailed documentation of the heat source as well as an ORC implemented at the sight [13]. Unfortunately, the ORC is not any longer in operation. An optimization in the drying process led to a decrease in the supplied waste heat. To keep the ORC in operation additional heat would have been to be supplied; the project was therefore closed. This development demonstrates once more one of the critical problems of current WHR technology and demonstrates that new dynamic technologies in this field are urgently needed.

In the study, the drying process was extended by a closed thermal oil cycle (**Figure 12**) in which the heat from the hot flue gases was bundled and made accessible. For cooling water at 15 °C was assumed. With the heat from the thermal oil the ORC was operated. In this work, the energy used to bundle the heat in the thermal oil as well as energy used to transfer the heat to the environment is not further considered.



**Figure 12 | Case scenario**

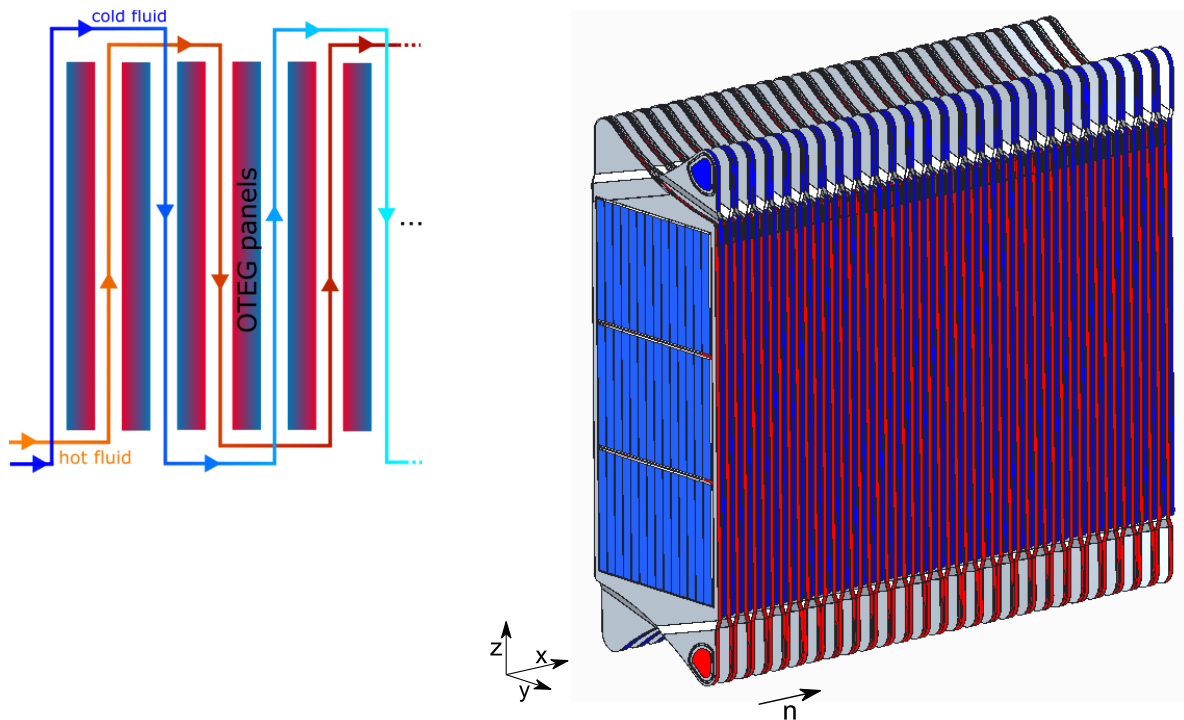
Waste heat potential of the cement plant in Lengfurth, Germany.

## 3.2 Constructive Concept

The purpose of a first concept was not to come up with a technically mature solution but rather to set the framework for upcoming simulation work. A plate heat exchanger setup was chosen and extended by an integration of the TEGs (**Figure 13**). Plate heat exchanger offer high heat transfer rates at compact dimensions and have a rather simple design. On the downside, sealing is an issue especially at high pressure (above 8 bar) and high temperature (above 150 °C) operating conditions. To extend the application range of plate heat exchangers on high-thermal and high-pressure conditions they are usually welded instead of screwed. Since plate heat exchanger are designed for liquid-to-liquid heat transfer, it was decided to use water as cooling medium in this project. The heat carrier fluid (Mobiltherm 594) was predetermined by the study case. Mobiltherm 594 is a thermal oil that is characterized by a high thermal stability, low viscosity over a large temperature range and a high heat conductivity coefficient.

**A**

**B**



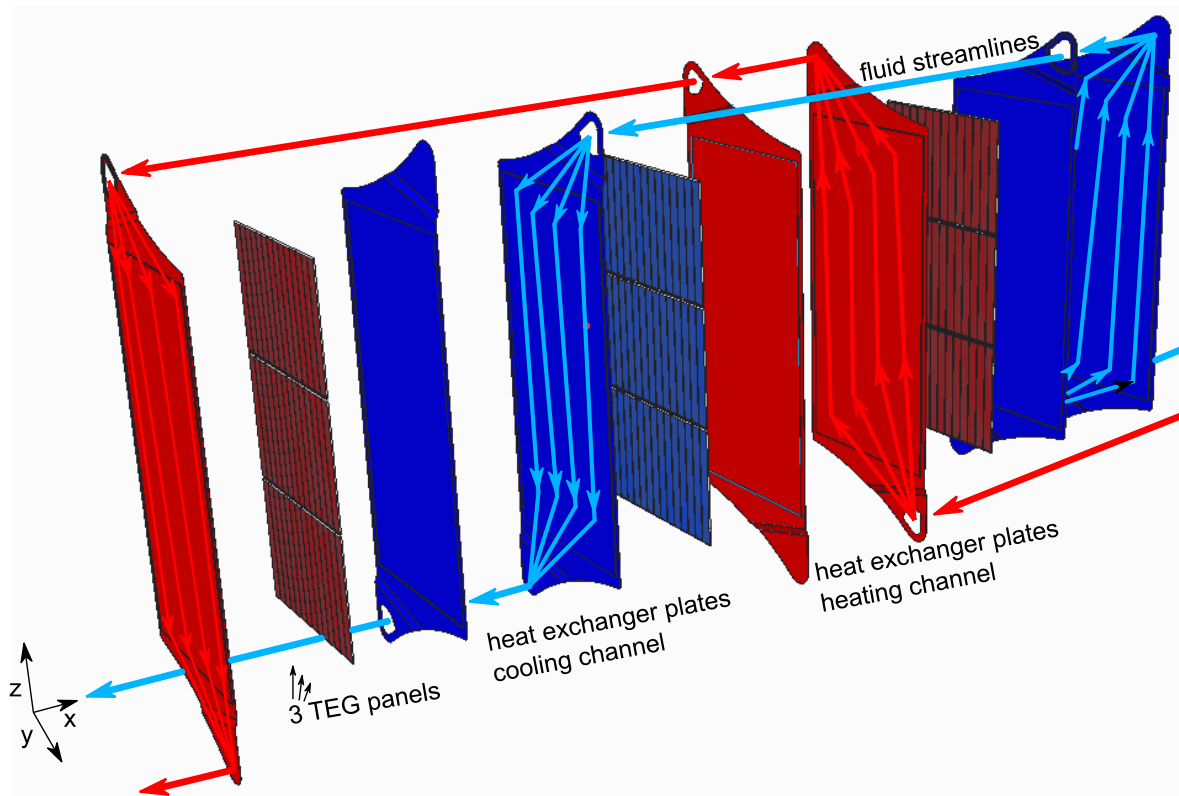
**Figure 13 | Heat exchanger stack**

**A**, schematic function principle of the heat exchanger stack. In-between every cooling and heating channel the surface is covered with TEGs working with the local temperature difference. In this setup, parallel flow conditions as well as counter flow conditions appear. **B**, illustrates a possible constructive stack design with 3 TEG panels in between every hot and cold channel.

Aside from the heat exchanger design the technical layout of the TEGs must be altered to fit the system design. Different to the current TEG layout which eventually produces a generator of the size of a sugar cube, a large surface must be covered by the generators. While the foils can already be produced at a large scale via printing technology, it is imaginable that the following production steps (**Figure 8**) can be customized to produce instead of a single cube generator a long cord made up of the individual TEGs. The cord is then cut into strings and arranged next each other. After sealing the TEGs in a polymer mantle a possible generator could look like a flexible panel of the proportions of DIN A4. These panels might be realized without the necessity to develop an entirely new sophisticated production process.

For simulation and visualization purpose, the constructive concept was implemented in CAD. An exploded view of the stack and the flow schemata of the operating fluids is illustrated in **Figure 14**.

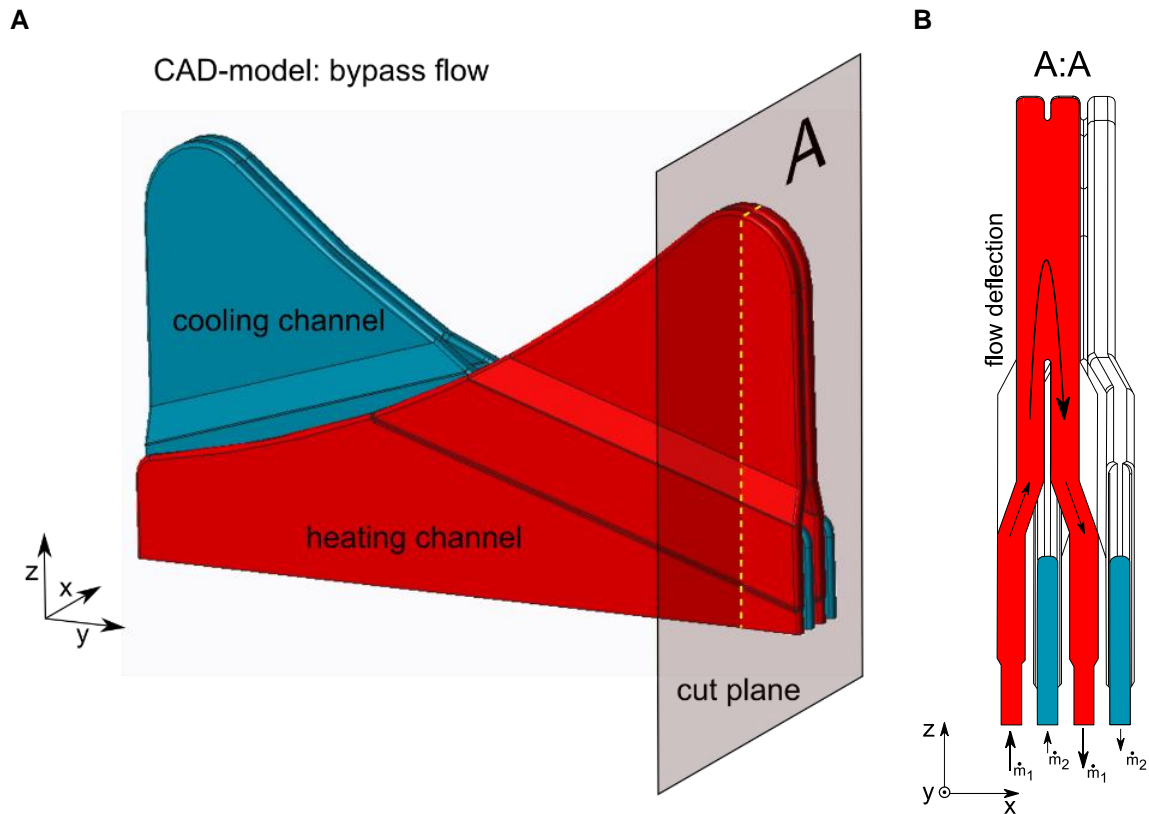
Considering the fluids, dynamical simulations of the flow in the channel and the bypass into the next channel was modeled and designed in CAD. Through the strong flow deflection (**Figure 15, B**) in the narrow bypass at the top and bottom of every exchanger channel the pressure drops is believed to be significantly high in this section of the heat exchanger.



**Figure 14 | Stream lines and stacking order of the heat exchanger stack**

Explosion of the stack assembly of a heat exchanger equipped in each case with 3 TEG panels. The arrows represent the streamlines of the working fluids furthermore the temperature in the figure is color-coded.





**Figure 15 | Fluid transfer to the next channel**

Cut through the bypass flow of the heat exchanger channel. To reach the next channel the flow needs to bypass the channel height and two times the TEG thickness. The channel height is 5 mm.

### 3.3 Thermal Model: Methodology

Simulink/MATLAB was used to simulate the thermal behavior of a WHRS based on a simple plate heat exchanger design like the one presented above. The fundamental working principle of the Simulink model is based on analytical and empirical equations from the fields of heat transfer, fluid dynamics and thermodynamics. Due to the simplicity, its symmetry and repeating design of the setup only a small section needs to be mathematically described and simulated. Subsequently the obtained specific values can be scaled up to conclude the absolute values of the entire WHRS. The foundation and the working principle of the Simulink model is best understood by following this breakdown:

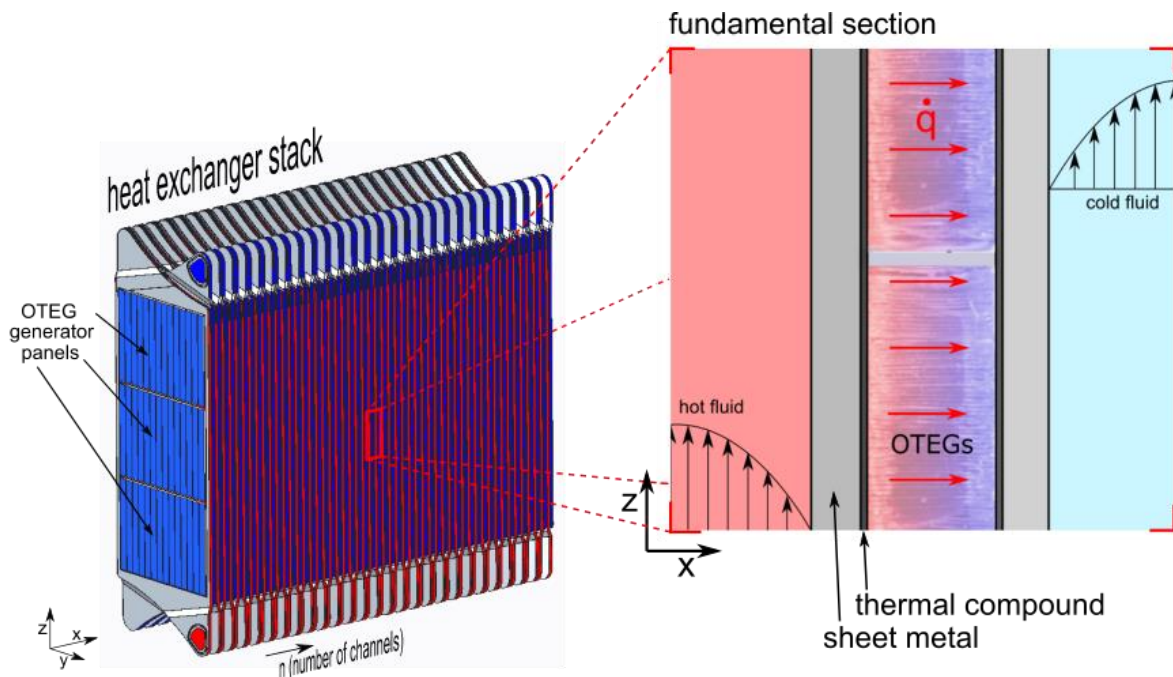
- Reducing Complexity: From the complex to the simple model.
- Thermal oil & Water: Simulating the working fluids.
- Newton's Law of Cooling: Solving the base equation.
- $R_{tot}$ : Determining the total thermal resistance.
- TEG Thickness: Optimizing the generator thickness via MPPT.
- Global Temperature Development: Heat transfer along the streaming length.

- Local Temperature Development: Heat transfer through the TEG.
- Power Output: Putting it all together.

### 3.3.1 Reducing Complexity

Part of the initial phase of every virtual modelling task is the reduction of the complexity of the real physical model. In this work, the constructive design of the heat exchanger equipped with TEGs was broken down to the elemental section that is sufficiently describing the working principle of the WHRS. Besides this simplification, additional assumptions are made to reduce complexity including:

- steady state operation
- adiabatic operation, heat is only entering and leaving the system via its mass flows
- evenly distributed heat transfer across the heat exchanger surface
- negligence of heat transfer in the z- and y-direction and at the side
- heat is only conducted through the TEG
- influence of heat radiation is not considered



**Figure 16 | Reduction of complexity**

Due to the symmetric and repeatable structure of the heat exchanger stack. The mathematical modelling can be done on the fundamental section representing a cut through the heat exchanger wall. Driving force of the heat transfer is the temperature gradient between the two working fluids. The total thermal resistance is composed of the individual resistances wall layers.

### 3.3.2 Thermal oil and Water

The fundamental factors of influence for the thermal simulation are the fluid properties of the working fluids. Specific fluid values such as: heat capacity  $c$   $\text{Jkg}^{-1}\text{K}^{-1}$ , density  $\rho$   $\text{kgm}^{-3}$ , heat conductivity  $\lambda$   $\text{Wkg}^{-1}\text{m}^{-1}$  and kinetic viscosity  $\nu$   $\text{m}^2\text{s}^{-1}$  but also the mass flow rates  $\dot{m}$   $\text{kgs}^{-1}$  are important influential factors. While the influence of the temperature is taken into account, the pressure dependence of the fluid properties is neglected. This assumption for non-compressible working fluids is acceptable. During the simulation, the fluid values are taken either from functions describing the values over the temperature or from interpolation of lookup table values. The values of both fluids, the thermal oil (Mobiltherm 594) water are taken from the VDI Heat Atlas [14]. The datasheets for the two fluids are attached to the appendix.

### 3.3.3 Newton's Law on Cooling

The base equation solved by the Simulink model is the one-dimensional Newtonian law of cooling, which describes a modification of Fourier's law of heat transfer. In it the thermal conductivity  $\lambda$  is replaced by the reciprocal of the thermal resistance  $R_{tot}$   $\text{WK}^{-1}\text{m}^{-2}$  or the heat transmittance coefficient  $\alpha$   $\text{W}^{-1}\text{Km}^2$ . For every time step in the model the equation is solved for the local values of  $R_{tot}$  and  $\frac{dT}{dx}$ . It is therefore necessary to determine these values in advance of every simulation step.

$$\dot{q} = \frac{\dot{Q}}{A} = \alpha \left( \frac{dT}{dL} \right) = \frac{1}{R_{tot}} \left( \frac{dT}{dL} \right) \quad (28)$$

### 3.3.4 Total Thermal Resistance $R_{tot}$

The total thermal resistance  $R_{tot}$  is composed of the two thermal resistances due to conduction and convection:

$$R_{tot} = R_{cond} + R_{conv} \quad (29)$$

It describes the local temperature development from the hot fluid across the TEG to the cold fluid at one point along the streaming length (**Figure 17**). The effective temperature difference  $\Delta T_{eff}$  affects hereby the efficiency of the generator.

The thermal resistance due to conduction is calculated from the thicknesses  $\delta$  and the thermal conductivities  $\lambda$  of the individual layers of the wall separating the two working fluids.

$$R_{cond} = \frac{\delta_1}{\lambda_1} + \frac{\delta_2}{\lambda_2} \dots \frac{\delta_{n-1}}{\lambda_{n-1}} + \frac{\delta_n}{\lambda_n} \quad (30)$$

For the given setup, the stacking order of the sheet metal, heat transfer paste and the TEG is composed as follows:

**Table 3 | Thickness, heat conductivity and thermal resistance of the stacking order**

Description	$\delta$ [mm]	$\lambda$ [WK <sup>-1</sup> m <sup>-1</sup> ]	$R_t$ [WK <sup>-1</sup> m <sup>-2</sup> ]
Sheet metal wall	0.5	20	0.000025
Heat transfer compound	0.5	5	0.0001
OTEGs	3.45	0.3	0.01153
Heat transfer compound	0.5	5	0.0001
Sheet metal wall	0.5	20	0.000025

$$R_{cond} [\text{WK}^{-1}\text{m}^{-2}] = 0.011555$$

**Table 4 | Thermal resistance due to convection for ambient pressure and stream velocities of  $v_1=0.18 \text{ ms}^{-1}$  and  $v_2=0.27 \text{ ms}^{-1}$**

Description	$\alpha$ [WK <sup>-1</sup> m <sup>-2</sup> ]
Mobiltherm 594 (227°C)	0.002843
Water (43°C)	0.006477

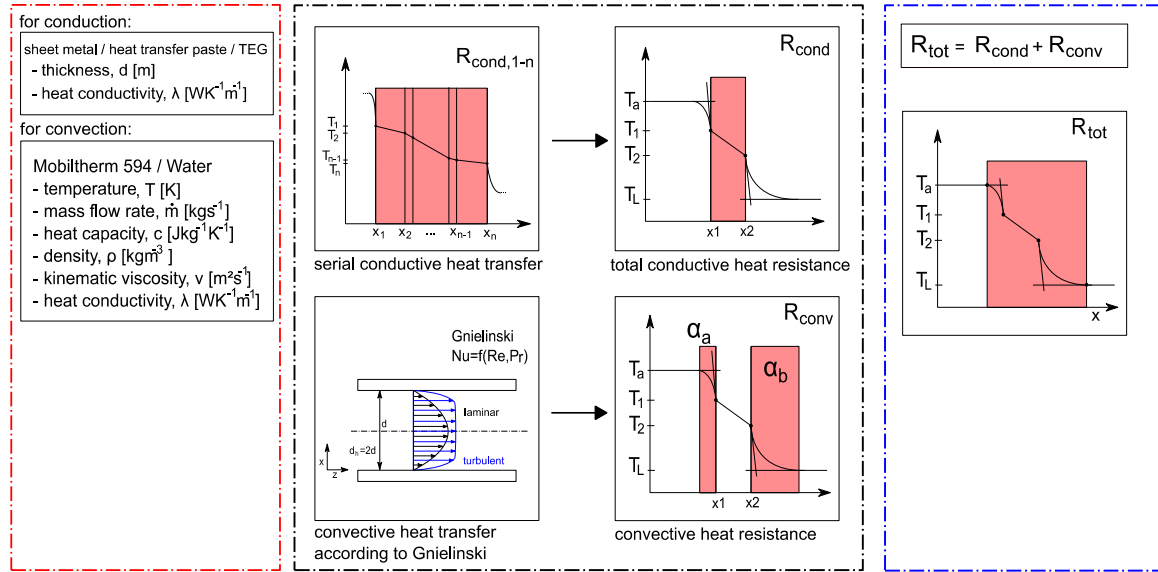
$$R_{conv} [\text{WK}^{-1}\text{m}^{-2}] = 0.00932$$

One of the significant parameters in the dimensioning of the systems represent the mass flows entering and leaving the heat exchanger. Most of the thermal energy leaves the system via the cooling flow while just a small portion is transformed into electricity. The convective heat transfer between the working fluids and the sheet metal walls plays hereby an important role. It thereby depends significantly on the flow conditions, quantified by the Nusselt number.

$$R_{conv} = \alpha_a + \alpha_b \text{ with } \alpha = \frac{\lambda \cdot Nu}{L_{char}} \quad (31)$$

With increasing flow velocities, the thermal resistance  $R_{conv}$  decreases. On the other hand, higher flow velocities lead to higher pressure drops in the channels. To optimize the relationship between the heat transformed in the TEGs and the dissipated energy in the channels is the main motive of this thesis work. To determine the amount of transformed heat into electricity the thermal model provides a solution. According to the empirical equations which Gnielinski derived for the heat transfer in parallel plate ducts, the three analytical equations for the Nusselt number for laminar [15], transition zone and turbulent [16] flow were implemented in Simulink.

## Input

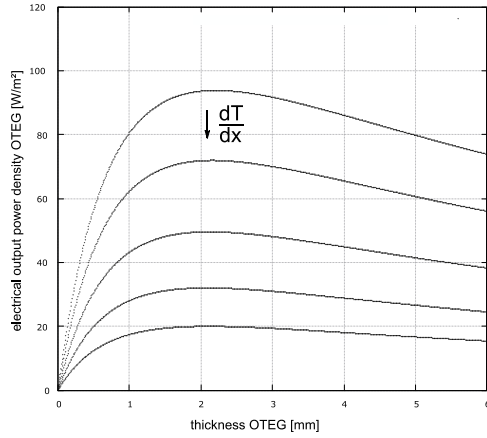


**Figure 17 | Calculation of the total thermal resistance**

$R_{tot}$  is split up into  $R_{cond}$  and  $R_{conv}$  and calculated individually during every time step of the simulation. The thermal resistance due to convection (fluid to wall) depends on the flow conditions while the conductive thermal resistance depends on the stacking order of the separation wall. The convective heat transfer is described after Gnielinski. Gnielinski supposes three empirical equations for the Nusselt number for laminar, transition zone and turbulent flow.

### 3.3.5 TEG Thickness

An important dimensioning factor of the TEG is its thickness  $d_{TEG}$ . When imagining a very thick TEG the conversion efficiency is large due to a large temperature difference. However, the heat flux becomes very small due to the high thermal resistivity, resulting in a small output power. The other extreme case describes a very thin TEG. Here the thermal resistance is low as well as the temperature gradient, which leads to a small conversion efficiency. These two extreme cases show that there must be an optimum for the thickness of the TEG. By implementing a maximum power point tracking approach, the optimum shall be determined in every simulation step. This happens at the beginning of every timestep of the simulation since the thermal resistance due to the OTEG ( $R_{OTEG}$ ) represents the mayor contribution to the total thermal resistance  $R_{tot}$ .



**Figure 18 | Maximum Power Point Tracking**

The optimal thickness of the TEG changes slightly depending on the local temperature difference. The thermal model includes a subroutine that does a maximum power point tracking to determine the optimal  $d_{TEG}$ .

### 3.3.6 Global Temperature Development

The other variable in Newton's law of cooling is the temperature gradient  $\frac{dT}{dl}$ . Over the streaming length of a heat exchanger, the temperatures of the working fluids are asymptotically converging towards each other. For the assumptions of steady state operation conditions and adiabatic conditions temperature profiles for the two working fluids are calculated by (24), (25), (26) and (27).

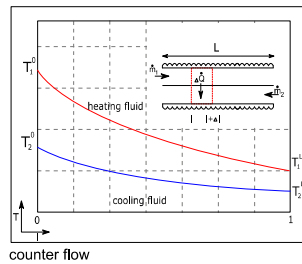
While Simulink works with time steps the simulation model is time independent. Therefore, the time steps are used to move through the heat exchanger along its streaming length. During every simulation step, the program is moving to a new position in the heat exchanger. The step size is determined by  $\frac{dl}{dt}$  and can be altered in the model. The smaller the step size is chosen the more accurate results delivers the simulation.

#### Input

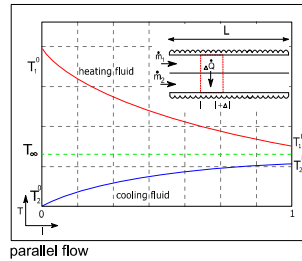
Mobiltherm 594 / Water  
- temperature,  $T$  [K]  
- mass flow rate,  $\dot{m}$  [kg s<sup>-1</sup>]  
- heat capacity,  $c$  [J kg<sup>-1</sup> K<sup>-1</sup>]  
- density,  $\rho$  [kg m<sup>-3</sup>]  
- kinematic viscosity,  $\nu$  [m<sup>2</sup> s<sup>-1</sup>]  
- heat conductivity,  $\lambda$  [W K<sup>-1</sup> m<sup>-1</sup>]

$$R_{tot} = R_{cond} + R_{conv}$$

- NTU1, cooling fluid  
- NTU2, heating fluid  
number of transfer units (NTU):  
dimensionless parameter  
to determine the technical  
layout of heat  
exchanger



counter flow



parallel flow

#### Output

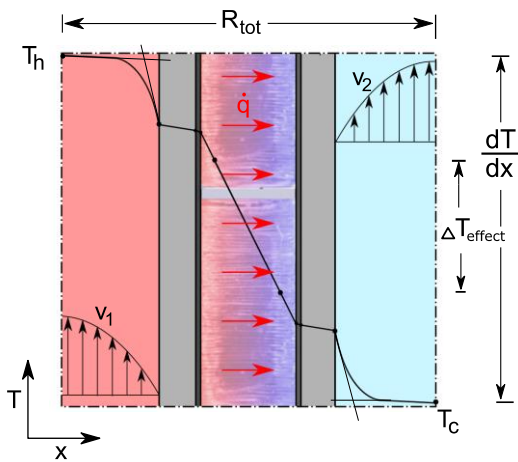
$$\frac{dT_{global}}{dl} = f(NTU1, NTU2, T_{01}^1, T_{01}^2)$$

**Figure 19 | Temperature development in the WHRS**

For steady state condition, the temperatures of the working fluids over the streaming length depend on the mass flow rates, the total thermal resistance and fluid parameters. In the literature, these parameters are usually merged in the dimensionless Number of transfer units ( $NTU$ ).

### 3.3.7 Local Temperature Development

At this point in the simulation, the base equation can be solved. This means that the heat flux density at the current position along the streaming length of the heat exchanger is known. With the heat flux density and the local temperature difference between the working fluids it is now possible to determine the local temperature distribution via Newton's law on cooling for the convective heat transfer and Fourier's law for the conductive heat transfer. Instead of solving the equation for the heat flux density, it is now solved for the unknown temperatures.



**Figure 20 | Effective temperature difference**

The temperature gradient used by the thermoelectric materials in the TEG ( $\Delta T_{eff}$ ) is lower than the temperature difference applied to the TEGs surface ( $\Delta T_{OTEG}$ ). For the Carnot efficiency  $\eta_{Car}$  and the figure of merit  $ZT$  the effective temperature difference is of importance.

With the acquired local temperatures distribution in the elemental section it is possible to determine the output power of the TEGs.

### 3.3.8 Putting it all together

In a final step, the results are combined to determine the overall performance of the system. Therefore, at every time step the transferred heat quantity is determined from the local heat flux density  $\dot{q}$ , the channel height  $b$  and the step size  $\frac{dl}{dt}$  following the relation:

$$\dot{Q} = \dot{q} \cdot A_{surf} \text{ with } A_{surf} = b \cdot \frac{dl}{dt} \quad (32)$$

To use this relation, isothermal heat transfer conditions from the fluid is assumed, although this is contradictory to the fact that the temperature is changing along the streaming length. For a small step size, the temperature stays approximately constant. The error made by this assumption is believed to be acceptable. By reducing the step size this error can be further minimized. The total heat transferred over the entire system results from the integrations of the individual heat quantities at each simulation step.

The quantity of the generated electricity by the TEGs is determined by the conversion efficiency  $\eta$  and the heat  $\dot{Q}$  transferred at each time step. Using Altenkirch's relation for a given  $ZT$  the conversion efficiency  $\eta$  is determined:

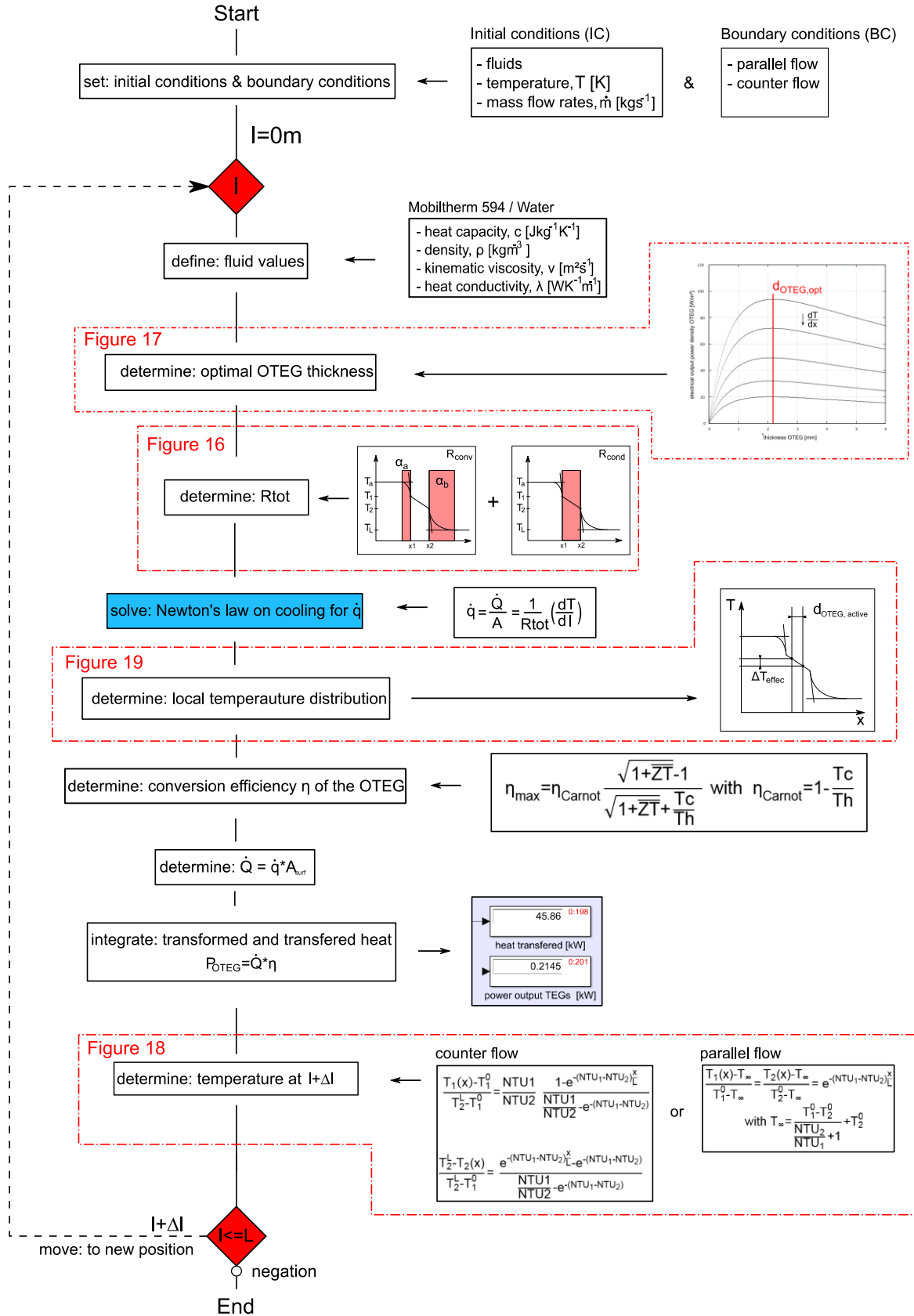
$$\eta_{max} = \eta_{Carnot} \cdot \frac{\sqrt{1 + \overline{ZT}} - 1}{\sqrt{1 + \overline{ZT}} + \frac{T_c}{T_h}} \text{ with } \eta_{Carnot} = 1 - \frac{T_c}{T_h} \quad (33)$$

The value  $Z$  will be kept constant over the simulation and multiplied by the mean temperature  $\bar{T}$ . The temperature dependence of the individual thermoelectric properties ( $S(T)$ ,  $\sigma(T)$  and  $\kappa(T)$ ) is therefore not further considered in this work.

As for the heat quantity, an integrator block is used to sum up the delivered output power  $P_{OTEG}$  over the streaming length ( $l$ ). To preserve consistency, the transformed energy must be subtracted from the total heat transferred.

$$P_{OTEG} = \int_0^L \dot{Q} \cdot \eta_{OTEG} dl \quad (34)$$





**Figure 21 | Operation chart of the thermal model**

Schematic working principle of the thermal model. The temperature differences along the streaming length as well as the temperature profiles through the heat exchanger wall are determined, which allows the calculation of the TEG power.

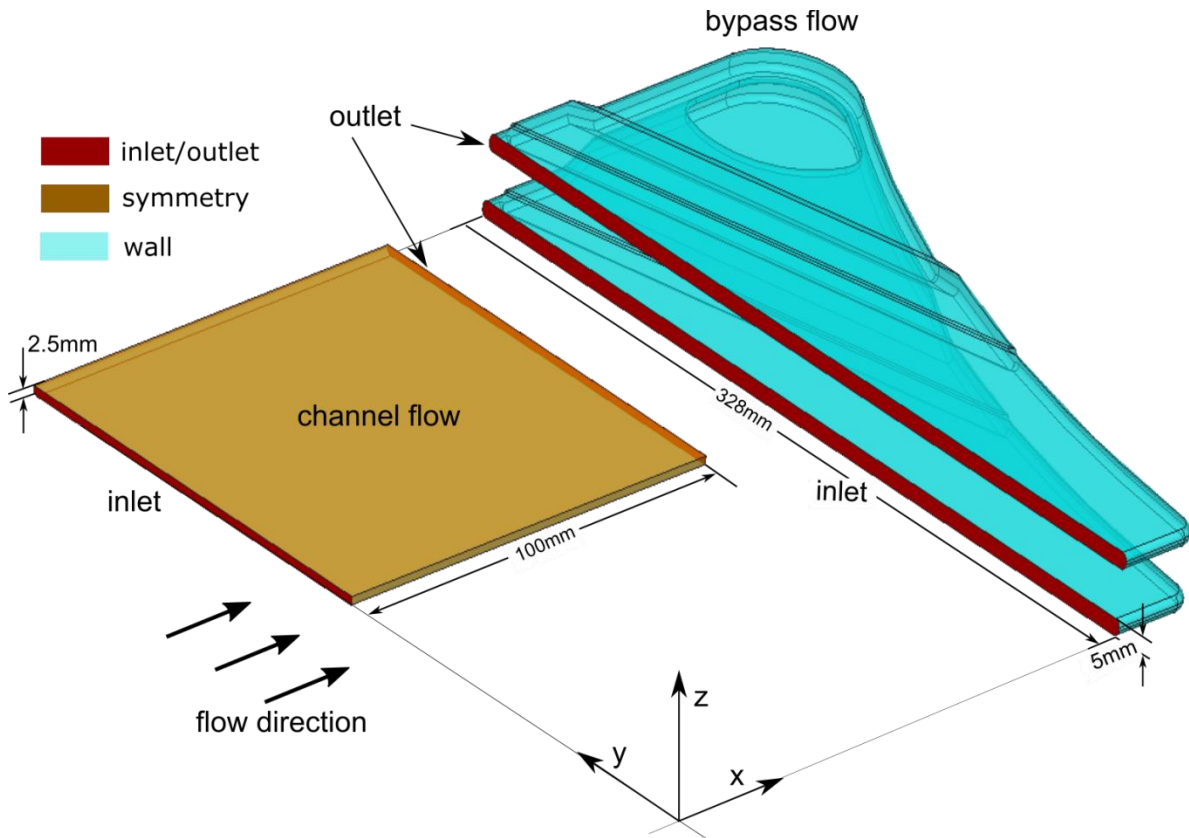
### 3.4 Fluid Dynamical Model: Methodology

In a closed thermodynamic system in which a fluid is in movement, the dissipated energy due to friction describes the main contribution to the exergy loss of the system. Setting up Bernoulli's law for such a system shows that this dissipative exergy loss manifests itself in form of a pressure drop. A circulation pump needs to supply in best case at least a pressure equal to the pressure drop to allow a circulation of the working fluids.

In the proposed system, the energy supplied to the circulation pump must be considered to allow a conclusion of the net power output of the entire system. Therefore, the flow in the heat exchanger was simulated numerically using OpenFOAM. Due to the simple, repeating and symmetric design of the system two sections of the heat exchanger were found to be sufficient to characterize the systems pressure drop. While for the parallel plane duct (channel flow) there is an analytical solution for a steady, incompressible flow by solving the Navier-Stokes equation, more complex flows need to be solved numerically. The CFD work can generally be split into three phases: pre-processing, simulation and post-processing. Each of these phases can be themselves be split into several sub phases. With the help of this structuration the methodology of the CFD simulation will be demonstrated in the following.

#### 3.4.1 Geometrical Model

Like the methodology applied for the thermal model the first task lies in a reduction of the complexity of the system. The two channels which describe the surface of the flow of the system were derived from the CAD model. The flow geometry was idealized by restricting the models to the essential fluid dynamical aspects, by filling small gaps and smoothening of flarings, steps and edges as well as elimination of details. In **Figure 22** the two simplified channel sections are illustrated. The flow is split into the flow along the wall where the heat is transferred (channel flow) and the bypass that guides the fluid into the next channel (bypass flow). For the rest of the thesis to these channels is referred to as channel flow and bypass flow. While for the channel flow the symmetry of the flow can be mirrored along the symmetry planes in y- and z-direction there are no symmetry conditions which can be used for the bypass flow.



**Figure 22 | Characteristic channel flows**

Quarter piece of the flow (referred to as channel flow). The symmetry of the channel is used to reduce calculation time of the CFD solver. Bypass flow describes the U-turn (see **Figure 15**) into the next heat exchanger channel due to the higher complexity and the disturbed flow the pressure drop in this channel is expected to be much larger.

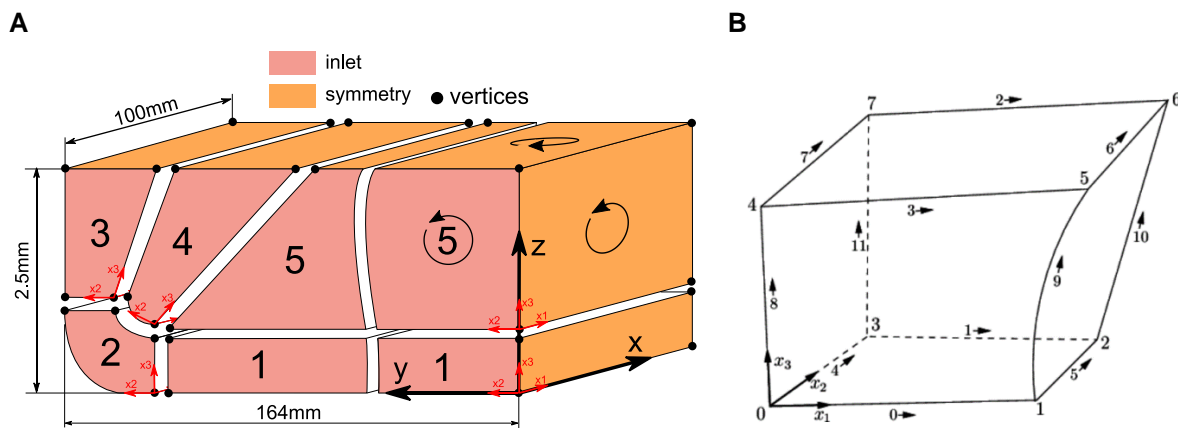
### 3.4.2 Meshing

The numerical method that uses OpenFOAM to solve the fluid dynamical problem is the finite volume method (FVM). It is therefore necessary to divide the flow geometry into small volumes in whose centers the values are saved describing the flow. In this work two different methods were applied to create the calculation mesh. For the mesh generation of the channel flow the OpenFOAM utility *blockMesh* was used while for the more complex geometry of the bypass flow the utility *snappyHexMesh* was chosen. The following describes the characteristics and functional principles behind both utilities.

#### **blockMesh**

The working principle behind *blockMesh* is the manual creation of the flow geometry. In the first instance, the domain geometry is split into one or several hexahedral blocks. Therefore the position of the vertices (corners) are defined in a global coordinate system (x, y, z). According to the order of implementation the vertices are numbered. OpenFOAM generally follows very closely the conventions of the C++ programming language, the first vertex is

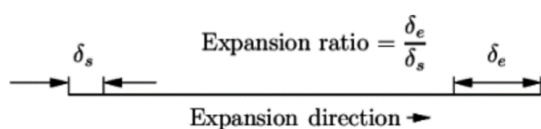
therefore numbering of elements begins with a zero. In the next step, the block is created by connecting the vertices with each other. Acceptable shapes for the connecting edges are straight lines, arcs or splines. For block two and four (**Figure 23, A**) the round edges were defined as arcs. The utility follows a special convention which must be followed very closely to run *blockMesh* correctly. Each block receives its own local coordinate system with the origin at its first vertex. The axes orientate itself according to the **Figure 23, B**. The  $x_1$ -direction points always in the direction of the edge “0”,  $x_2$ -direction along the edge “5” while  $x_3$ -direction follows edge “8”. Best practice guidelines for *blockMesh* recommend, a detailed sketching of the geometry prior to writing the *blockMesh* script.



**Figure 23 | OpenFOAM utility *blockMesh***

**A**, breakdown of the channel flow in the individual blocks (1...5) for the *blockMesh* utility. Every block consists of 8 vertices, the positioning of the local coordinate systems depends on the order in which the vertices are arranged in the script. **B**, shows the numbering convention of the blocks edges and vertices [image source: <https://cfd.direct/openfoam/user-guide/blockmesh/>].

After the blocks have been created they are further divided into the finite volumes. This is done by defining a number of cells in each direction of the local coordinates. When defining the number of cells of each individual block it must be kept in mind that the number must match the number of cells of the neighboring blocks. In CFD simulation it is common practice to refine the net in regions where gradients of the streaming parameters are expected. For the channel flow, this is the case close to the walls. Therefore, the *blockMesh* utility offers simple grading, edge grading and multi-grading for the blocks. For the simple geometry of the channel flow simple grading (**Figure 24**) was used to refine the mesh towards the walls.



**Figure 24 | Simple grading of *blockMesh***

Simple grading specifies uniform expansion along the local directions  $x_1$ ,  $x_2$  and  $x_3$ . [image source: <https://cfd.direct/openfoam/user-guide/blockmesh/>]

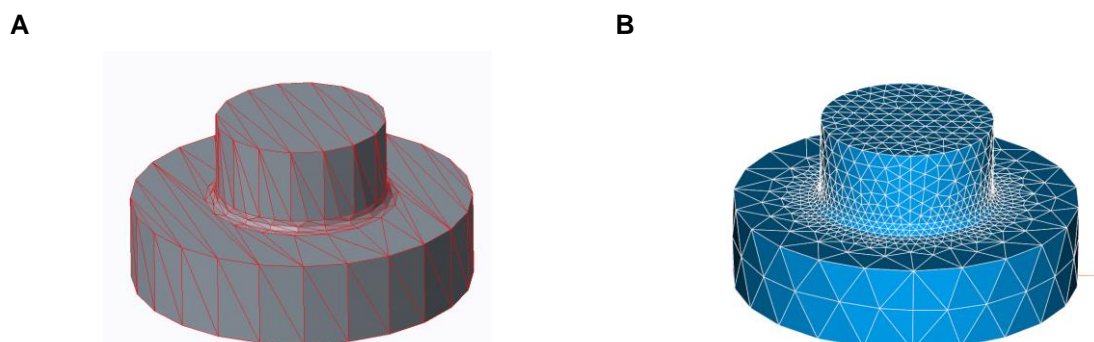
Parametrization for mesh generation can be useful especially regarding automation and large meshes (e.g. refinement, change of several vertices). The *m4* macro language offers

the possibility to use math operation when creating the *blockMesh* script. This language is hereby not particularly a part of OpenFOAM but on most Unix-like systems available. Using *m4* preprocessor language the channel flow was defined with its basic geometry parameters: length, width, height, arc radius.

The last entry of the *blockMesh* script describes the boundaries of the mesh. For the channel flow, there are four boundaries that need to be defined: input, output, wall and symmetry (**Figure 22**). In CFD the individual boundary surfaces are referred to as patches. Like the previous entries to the script there is a convention for the correct declaration of the boundary patches. Each block face that describes a boundary is defined by a list of four vertices. While the first vertex can be chosen randomly the order of the following vertices must follow the rule: when standing inside the block and looking on the surfaces the vertices are listed along the clockwise direction. In **Figure 23, A** this is indicated for block five. If the blocks surfaces within the geometry share the same vertices with its neighbors the patches are automatically merged. If this is not the case it has to be merged separately. A more detailed description of the *blockMesh* utility offers the official OpenFOAM website<sup>10</sup>.

### snappyHexMesh

Another OpenFOAM utility for mesh generation that was used for the bypass flow is *snappyHexMesh*. Like *blockMesh* it creates a mesh consisting of hexahedral cells, however the working principle is another. Instead of creating the geometry in the utility directly it needs to be defined first in a CAD-Software and exported. Therefor the geometry of the bypass flow was exported from Creo Parametric 3.0 in a neutral CAD format such as STEP or IGES and further modified in the open-source software Salome. This was done since Creo Parametric 3.0 only offers very limited adjustment options for the file format STL (STereoLithography) with which *snappyHexMesh* works.



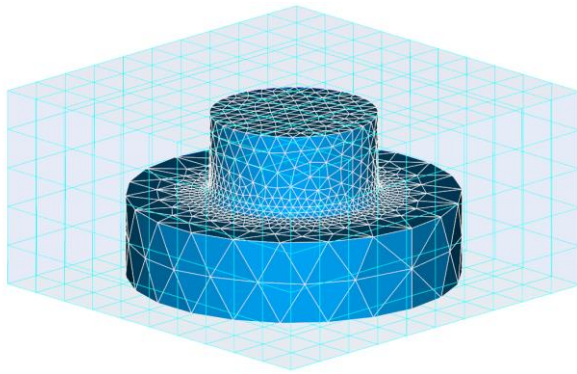
**Figure 25 | File format STL**

**A**, STL mesh from Creo Parametric 3.0. **B**, advanced STL mesh of the surface geometry in Salome.

The STL file format defines the geometry by covering the geometric surface with a net of triangles (**Figure 25**). When exporting the STL files from Salome it should be noted to export

<sup>10</sup> <https://cfd.direct/openfoam/user-guide/blockmesh/>

the meshes separately. Meaning that prior to the export, the mesh is split up in several sub-meshes. In **Figure 25, B** the mesh is made up of two meshes that are merged around the step curve. This procedure ensures that in *snappyHexMesh* the individual meshes can be altered apart from each other. An exemplary separation could be according to the boundary affiliation. Meaning that the meshes that define the input are together in one file but separately from the meshes defining the output. Also, the character encoding of the file should be in ASCII to allow manual manipulation of the file. By adding an indicative name at the beginning and at the end of the STL files we can later use these names for some sub-utilities in *snappyHexMesh*. In the *snappyHexMesh* script we define the geometry by listing the individual STL files. The program will automatically navigate to the correct folder and load the geometry. The tool requires an additional mesh surrounding the geometry (**Figure 29**). A commented *snappyHexMesh* script used for the bypass flow is added to the appendix.

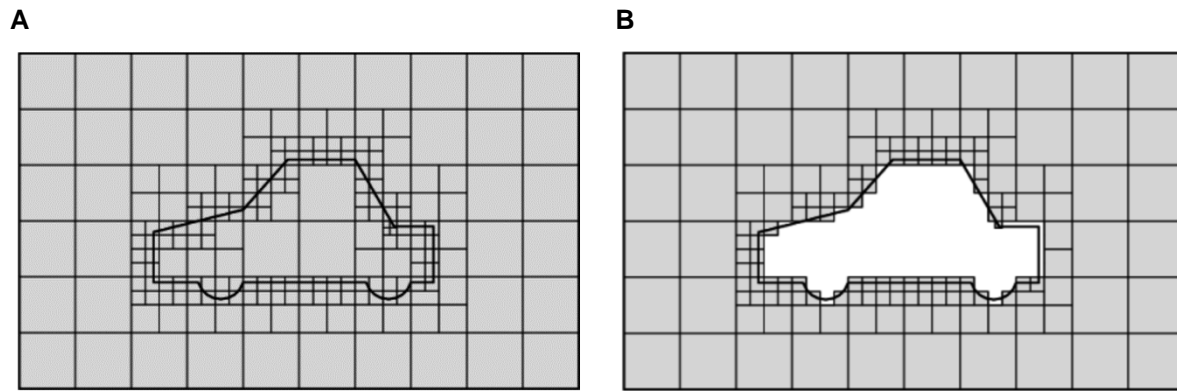


**Figure 26 | Bounding box for *snappyHexMesh***

When creating the boundary box, it must be kept in mind that the initial cell size (level 0) is dictated by the bounding box. An unfavorable initial cell size can lead to too much cells and might overload the RAM of the working station.

Now that *snappyHexMesh* has the geometric information it proceeds in three steps that are consecutively executed by the program: *castellatedMesh*, *snap* and *addLayers*. In the first step *snappyHexMesh* finds the intersections of the cells of the bounding box with the geometric triangular surface of the STL files. The cells of the bounding box sharing an intersection with the geometry are refined to a level that is specified in the *snappyHexMesh* script. The cells of the bounding box hereby dictate the “level 0”; the next level consist of four equidistant hexahedral cells that split the previous cell. This method creates a first rough approximation of the surface. Following the refinement step the unneeded cells are deleted. For the program to decide if either the interior or exterior cells of the geometry are kept a point must be specified in the region, where the mesh shall be conserved. In **Figure 27** the *castellatedMesh* step for an outer flow around a car profile is illustrated.



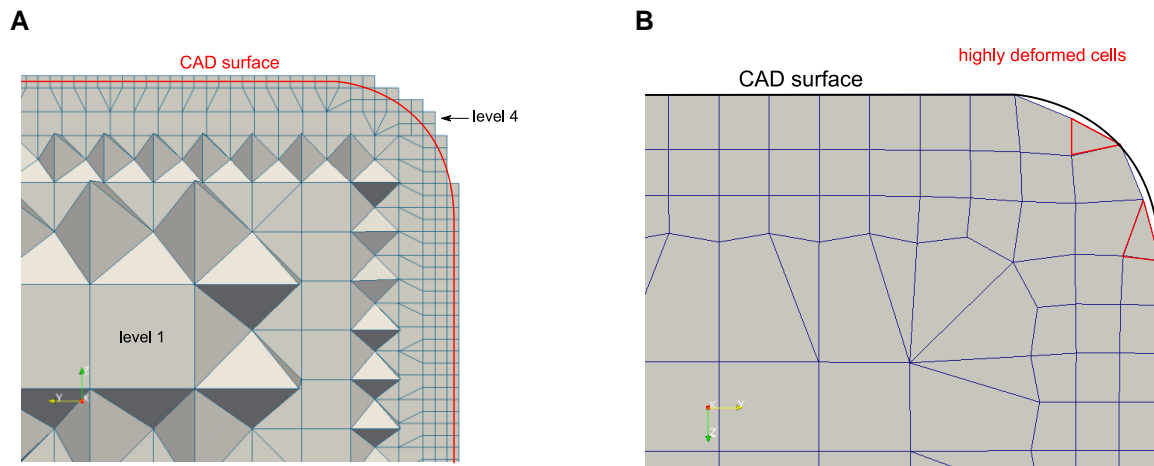


**Figure 27 | castellatedMesh**

**A**, refinement of the cells near to the geometry surface. **B**, deletion of the interior cells.

[image source: <https://cfd.direct/openfoam/user-guide/snappyHexMesh/>]

In the snapping phase, the cells near the surface are altered. Therefore, their edges are snapped on to the surface of the STL surface. While in this manner the mesh adopts the final shape of the geometry the snapped cells are deformed and skewed (**Figure 28, B**). Yet to preserve a certain quality of the mesh the edges of the deformed cells are iteratively moved and compared with quality values during the snapping phase. If an acceptable quality is reached or the maximum number of iteration steps has been exceeded, *snappyHexMesh* moves to the final phase *addLayers*.

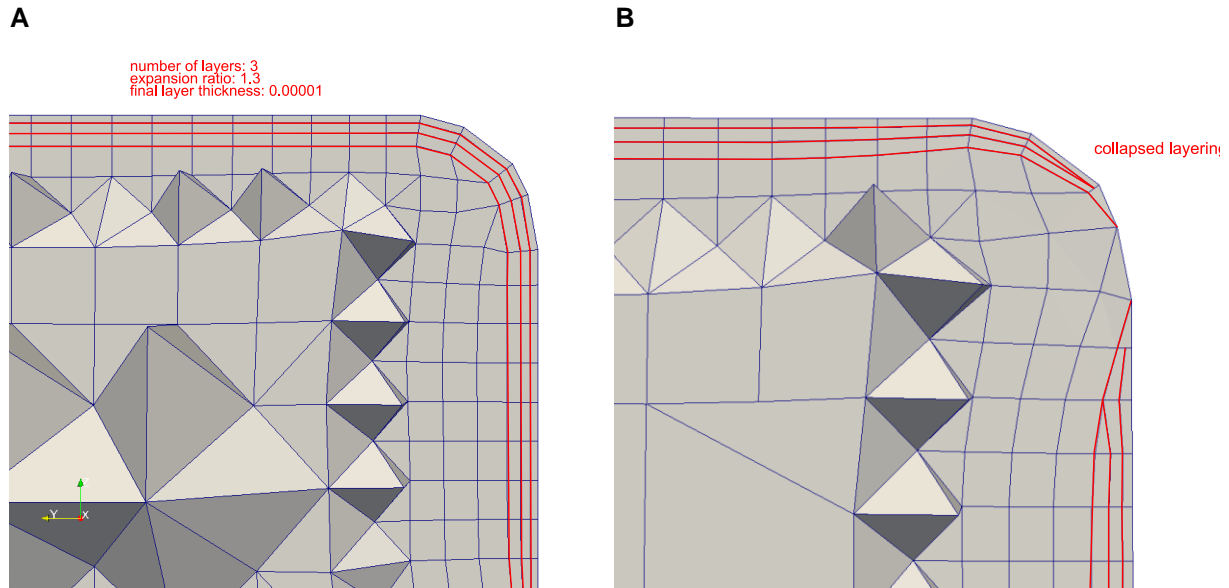


**Figure 28 | castellatedMesh and snapping**

**A**, result of the *snappyHexMesh* utility after the sub routine *castellatedMesh* for a section of the bypass flow. **B**, result after snapping the overlapping edges to the on to the STL surface.

During the layering, additional layers are inserted between the specified surface and a distance to it. In the *blockMesh* utility the layering was done by defining a simple grading towards the wall. Here the *snappyHexMesh* must be told the number of layers to be inserted, the expansion ratio and the final layers thickness. Based on this input *snappyHexMesh* tries to create the layers. For undisturbed geometries, the layering works well while at sharp edges and where different levels come together the layering regularly collapses. Although

the layering phase executes iterative optimization steps similar to the ones executed during the snapping phase, good results for *addLayers* are difficult to achieve.



**Figure 29 | addLayers**

**A**, successful layering around a corner. **B**, collapse of the layering due to unfavorable snapping in the routine anterior.

Summarizing the mesh generation, *snappyHexMesh* offers an automated method to create a hexahedral mesh for a given geometry while *blockMesh* follows a manual approach. The automation approach of *snappyHexMesh* does not necessarily implies that it delivers better results or is less time consuming than *blockMesh*. From the experience gathered on the two utilities during this work it is doubtful that *snappyHexMesh* will ever deliver a better mesh but for complex geometries the time saving can be significant. The difficulty when using *snappyHexMesh* is to get a high success rate from the three steps and still have cells with acceptable quality values. To achieve this, it is essential to simply run *snappyHexMesh* several times and try to find the optimal configurations.

### 3.4.3 Solver

On the computed meshes for the two channels the CFD calculations were executed. The solver used is one of the standard solvers that come with OpenFOAM; the *simpleFoam* solver works with the incompressible form of the Navier-Stoke equations (NSE) for steady, viscous and single-phase flows. Besides a laminar flow a turbulence model based on the Reynolds-averaged Navier-Stoke equations (RANS) was implemented for high flow velocity. By means of this the pressure drops for both working fluids (Mobiltherm 594 and water), flowing in the two channels was determined for laminar and turbulent flow conditions. The methodology behind this procedure is explained in the following.



### Fundamental Equations “simpleFoam”

Due to the simplification of the flow (incompressible, steady state, single-phase, Newtonian fluid, viscous) the *simpleFoam* solver does not solve the general form of the NSE, but rather a further simplification of (15) and (16) in the form:

$$\text{(mass conversation)} \quad \nabla \cdot u = 0 \quad (35)$$

$$\text{(momentum conversation)} \quad \nabla \cdot (uu) + \nabla \cdot \nu_{eff}(\nabla u) = -\nabla p \quad (36)$$

In this form  $u$  describes the velocity field,  $\nu_{eff}$  the effective kinematic viscosity and  $p$  the relative pressure field divided by  $\rho$ . During post processing, it is important to multiply the relative pressure with the density to get the real pressure. The equations are used to solve the laminar as well as the turbulent flow problem. The use of the RANS model implies that turbulence effects are completely described in a turbulence model. As long as local turbulent effects are not of interest the RANS modelling represents an adequate method. For basic turbulent viscosity models the effective viscosity  $\nu_{eff}$  is composed of the molecular (fluid specific) viscosity  $\nu$  and a turbulent part  $\nu_\tau$  described by the turbulence model.

### The SIMPLE and SIMPLEC Algorithm

The SIMPLE (Semi-implicit Method for Pressure Linked Equations) algorithm represents the numerical schemata that solves the fluid dynamical problem on the created mesh. Next to the SIMPLE algorithm an alternative version, the SIMPLEC algorithm was used. Without going to deep into the details about the fundamentals of the algorithms, the difference lies in an additional approximation of the velocity corrector term which is neglected for the SIMPLE algorithm. In theory, the SIMPLEC algorithm is therefore more consistent and tends to converge faster [17].

#### 3.4.4 Boundary Conditions and Initial Conditions

In order to be able to solve the fluid dynamical problem boundary and initial conditions have to be defined. For the walls, the no-slip first-type boundary condition (Dirichlet) is defined as a fixed zero value for the velocity field  $u$ . For the outputs of two channels the pressure is set as well with a Dirichlet boundary condition to a fixed zero value this can be done since we are only interested in the pressure difference. At the walls surface a Neumann or second-type boundary condition is defined for the pressure. For the laminar flow, these boundary conditions are sufficient to run the simulation. Depending on the turbulence model used for

the turbulence flow further boundary conditions for the turbulent parameters need to be set. In this case a Launder & Sharma  $k$ - $\varepsilon$  model delivers the turbulent viscosity  $\nu_\tau$  via the introduction of the turbulent kinematic energy  $k$  and the dissipation rate of the turbulent kinematic energy  $\varepsilon$ . At the wall, the  $k$  and  $\varepsilon$  are set with a fixed value close to zero but not exactly zero. This is done out of numerical reasons and does not have any physical reason. At the outlets  $k$  and  $\varepsilon$  the gradient is set to zero.

**Table 5 | Boundary conditions for the characteristic channel flows (wall, outlet and symmetry)**

Description	wall	outlet	symmetry
Velocity field, $(u_1 \ u_2 \ u_3)$	fixedValue (0 0 0)	zeroGradient	symmetry
Pressure field, $p$	zeroGradient	fixedValue 0	
Turbulent kinematic energy, $k$	fixedValue 1e-10	zeroGradient	
Dissipations rate of $k$ , $\varepsilon$	fixedValue 1e-10	zeroGradient	
Turbulent viscosity, $\nu_\tau$	nutLowReWallFunction	calculated	

While the boundary conditions for the walls, outlet and symmetry patches can be defined relatively straight forward, the definition of the correct initial conditions of the inlet is more sophisticated. To set the wanted initial conditions at the inlet the three methods used will be explained in the following: *mapping*, *swak4Foam* and *lookup2DTable*.

## Mapping

The objective of the mapping utility is to change the inlet conditions in a manner that the values at the outlet from the previous simulation step are reused to define the new inlet conditions. Using the mapping method, it is possible to simulate a fully developed channel flow on a relatively small mesh.

## swak4Foam

*swak4Foam* stands for SWiss Army Knife for Foam.

*“Like that knife it rarely is the best tool for any given task, but sometimes it is more convenient to get it out of your pocket than going to the tool-shed to get the chain-saw.”*<sup>11</sup>

The library *swak4Foam* which is not part of the official OpenFOAM release offers several utilities to create custom initial conditions and other objects that would otherwise require the user to do C++ programming within the OpenFOAM environment. Over the course of this work the sub-utility *groovyBC* was used to define parabolic initial conditions for the velocity fields at the inlet of the bypass flow. The objective of this procedure was to pass the outlet

<sup>11</sup> <https://openfoamwiki.net/index.php/Contrib/swak4Foam>

velocity field from the channel flow as inlet conditions to the bypass flow. Therefore, the velocity profiles were fitted using gnuplot and saved. Prior to running the simulation of the bypass flow, the program would navigate to the folder of the corresponding channel flow and pass the fitting information to the current simulation case.

### lookup2DTable

Despite the summary of several different utilities in *swak4Foam*, the library was found to be unsuitable to define the initial conditions of the turbulent parameters  $k$  and  $\varepsilon$  since it only offers a 1-dimensional lookup table. To still be able to supply custom initial field conditions the OpenFOAM C++ class *interpolation2DTable* was used to define the cells at the inlet depending on their y- and z-coordinates. Prior to this a script had to be written that brings the VTK output file into the correct form required for the *interpolation2DTable* to work with.

**Table 6 | Boundary conditions of the channel flow for the inlet (laminar, turbulent)**

Description	laminar	turbulent
Velocity field, $(u_1 \ u_2 \ u_3)$	mappedPatch	mappedPatch
Pressure field, $p$	zeroGradient	zeroGradient
Turbulent kinetic energy, $k$	/	mappedPatch
Dissipations rate of $k$ , $\varepsilon$	/	mappedPatch
Turbulent viscosity, $\nu_\tau$	/	calculated

**Table 7 | Boundary conditions of the bypass flow for the inlet (laminar, turbulent)**

Description	laminar	turbulent
Velocity field, $(u_1 \ u_2 \ u_3)$	swak4Foam	swak4Foam
Pressure field, $p$	zeroGradient	zeroGradient
Turbulent kinematic energy, $k$	/	interpolation2DTable
Dissipations rate of $k$ , $\varepsilon$	/	Interpolation2DTable
Turbulent viscosity, $\nu_\tau$	/	calculated

### 3.4.5 Post Processing

The value of interest for this work is the pressure drop  $\Delta p$  over the streaming length of the channel flow and the bypass flow. To monitor the evolution of this value after every simulation step the average pressure difference between the inlet and the outlet was saved.

### Monitoring of the Simulation

To monitor the progress of the simulation two parameters were scoped during the simulation: the residuals and the pressure drop. Since the turbulence was modeled to 100% the residual

development for both flow conditions (laminar and turbulent) theoretically gives evidence of the progress of a simulation. In practice, this is not always the case; a plausible example where the residual monitoring fails is the Kármán vortex street. Although the stream might be laminar disturbances in the flow conditions generally prevent the solution to converge. For this reason, the pressure drop was monitored to allow an estimation of the solution when the residuals would not allow a prediction.

### 3.4.6 OpenFOAM

The following description shall give a brief overview on the most relevant aspects of the working principle of OpenFOAM. Since it does not come with a build-in general user interface (GUI) its function principle might seem abstract at the beginning especially for users that are not used to command line interface (CLI) based programs. Apart from that the structure of OpenFOAM is build up in a very logical and consequent manner. The case folder always contains the fundamental subfolders: *0*, *constant* and *system*. These three folders contain the minimal information for OpenFOAM to solve a problem.

For a steady state solver, OpenFOAM solves the equations of a physical model in a pseudo-time. For every solution that is specified to be written, a folder with the corresponding timestep will be created and added to the case structure, e.g. folder *0.002* for the time 0.002s. Also present in the case folder there are often found scripts, that execute different jobs such as: reset the original case structure, change parameters in the files, fit solutions, etc. The dimensions with OpenFOAM works are the physical base units: mass, length, time, temperature, amount of substance, amperage and luminous intensity [kg m s K mol A cd]. These are specified in the given order by its exponents e.g. velocity [0 1 -1 0 0 0 0].

#### Folder 0

According to the convention the folder *0* contains the solution after 0s. This is the case since in this folder the boundary and initial conditions are specified. For every flow parameter that is changing over time a file must be created. A laminar flow contains therefore a file for the pressure ( $p$ ) and the velocity ( $U$ ) while for a turbulent flow files for the turbulent kinetic energy ( $k$ ), its dissipation rate (*epsilon*) and turbulent viscosity (*nut*) need to be appended. Exemplary files can be taken from tutorials or from the official OpenFOAM websites.

#### Folder constant

How the name implies in the folder *constant* usually contains information about the parameters that do not change over the course of the simulation. In *transportProperties* the material values are defined in this case the fluid specific viscosity  $\nu$ ; the *turbulenceProperties*

defines the turbulence model used. The folder *polyMesh* contains the information about the mesh. For this purpose, the utilities *blockMesh* and *snappyHexMesh* create their files that clearly define the mesh within this folder: *boundary*, *faces*, *neighbour*, *points*, *owner*.

### Folder system

The folder *system* summarizes the files that give instructions for the solver to solve the CFD problem. Most important are the files *controlDict*, *fvSchemes* and *fvSolution*. The *controlDict* delivers instructions for the time control, solutions, output, post processing, etc. while *fvSchemes* and *fvSolution* define numerical methods and schemes such as: discretization methods for the operators and the fields, solver used, convergence criteria's, under relaxation, etc. For the targeted use of *fvSchemes* and *fvSolution* a deeper understanding of the mathematical principles in the field of numerics is necessary.

**Table 8 | Additional OpenFOAM files in the folder *system***

File name	description
meshQualityDict	Defines quality values of the mesh cells. This file is used by several different utilities e.g. <i>snappyHexMesh</i> .
snappyHexMesh	The <i>snappyHexMesh</i> script must be saved in this folder to run the utility.
decomposeParDict	Complex geometries are generally run on several cores in parallel. The file defines the method how the mesh is divided on the individual cores.
pressureDifferencePatch	A post processing tool that logs the pressure difference between two patches (here inlet and outlet). The last entry into the file is used to determine the pressure drop of the flows. Additionally, the convergence of this parameter was monitored to make an assumption about the progress of the simulation.
singleGraph	A post processing tool that delivers flow values over along a coordinate. This tool is used to determine the velocity profiles of the channel flows.
sampleDict	An outdated post processing tool. New versions of OpenFOAM can still run this utility with an extra command.

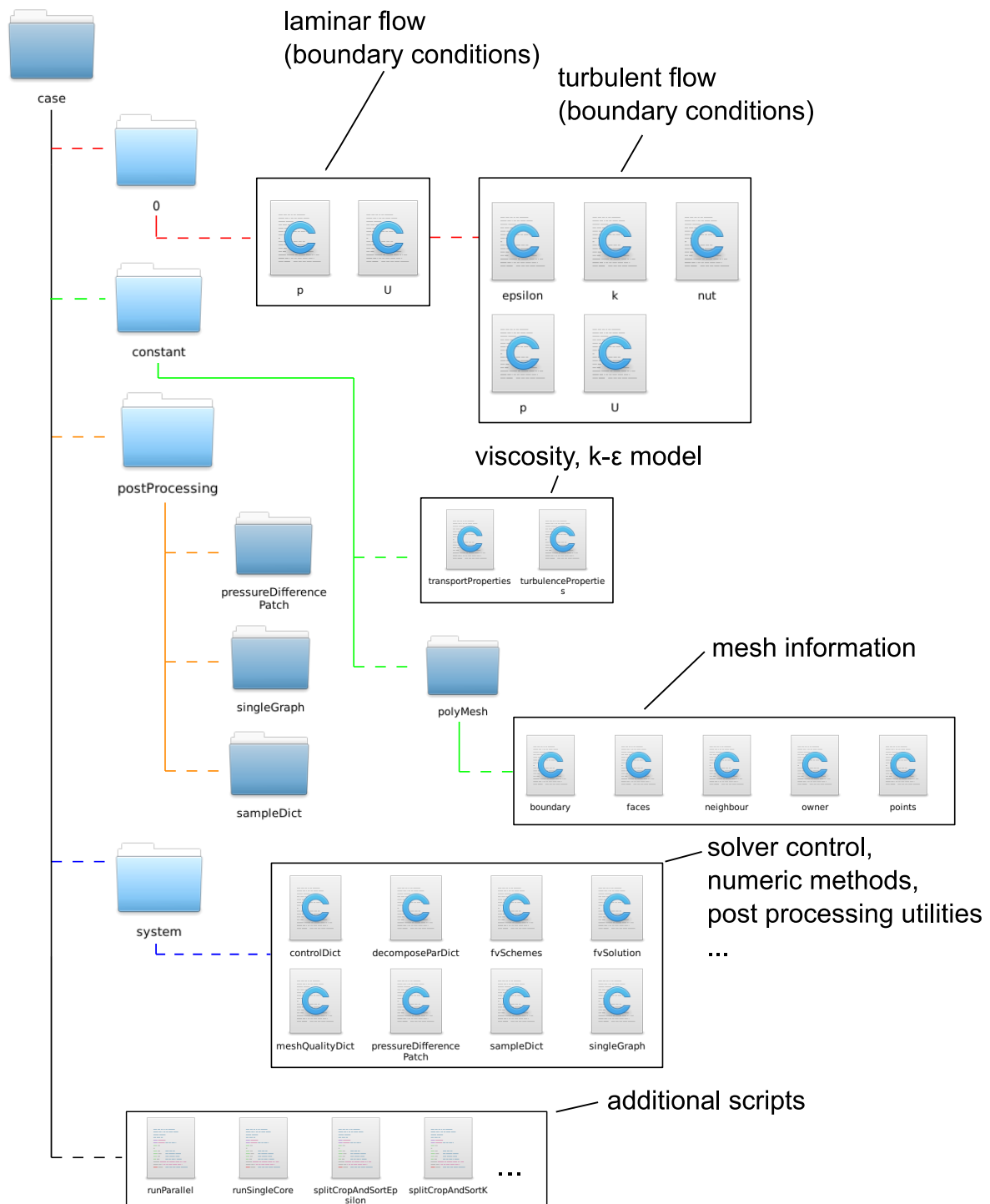


Figure 30 | Exemplary OpenFOAM case folder

### 3.5 Shell Scripting

Paradox may seem that although a flow through a heat exchanger is simulated, the solver does not solve the energy equation. This means that for the simulated system, isothermal conditions are assumed. Obviously, an isothermal system contradicts with the working principle of a heat exchanger system at least as long as no phase changes occur. However, for a short segment along the channel flow the temperature can be assumed to stay constant. To still be able to make an estimation for the pressure drop including the temperature influence with the *simpleFoam* solver, several simulations are conducted in a pre-set temperature and velocity range.

Here the relatively simple automatization by running scripts is one of the strengths of command line interface programs (CLIs) and comes in handy. For this work OpenFOAM was executed via the BASH (Bourne-again shell) which is the default login shell for most Linux distributions. Several shell scripts were written that would automatically create the OpenFOAM case structure, manipulate data, run the case, fit data curves and write the results to a file. More precisely the script would create the case structure for the channel flow or the bypass flow, define the temperature and the corresponding fluid parameters, write those parameters to the correct files in the case structure and set initial boundary conditions.

For the bypass flow the profiles for  $U$ ,  $k$  and  $\varepsilon$  at the outlet of the channel flow are identified and set as initial conditions in the bypass simulation for the turbulent and laminar simulations. Due to a good approximation of the velocity profile  $U$  by a polynomial function, *swak4Foam* was used to describe the initial velocity field. Since  $k$  and  $\varepsilon$  have a rather complex profile that cannot any longer be approximated by a single polynomial function, the method of using a *lookup2DTable* was favored. By this manner an upstream geometry that would create a fully developed inlet flow is redundant. When the OpenFOAM case is completely set-up the script runs OpenFOAM on itself and saves the pressure drops in a file. A flow chart that describes the general principle of the shell script is illustrated in **Figure 31**. In the chart, an inner loop and an outer loop are implemented the outer loop changes the average velocity while the inner loop alters the temperatures of the fluid.

Besides the script that manages the simulation, other scripts had been created that would take over tasks such as fitting of flow parameters, manipulation of lookup table data, manipulation of STL files, etc. A commented version of one script that would set up the case structure and run the simulation can be found in the appendix, as well as a list of the most important shell commands.

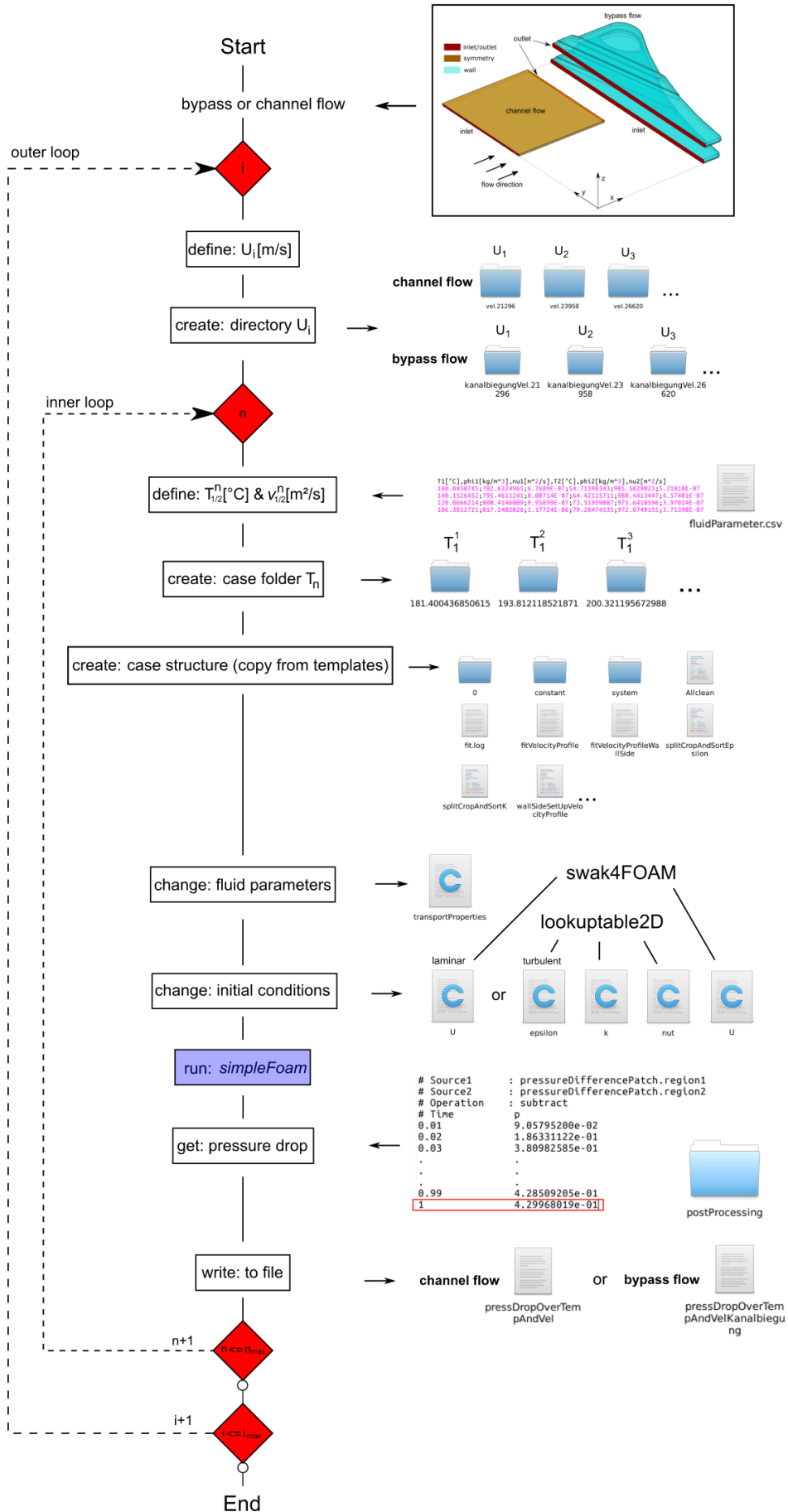
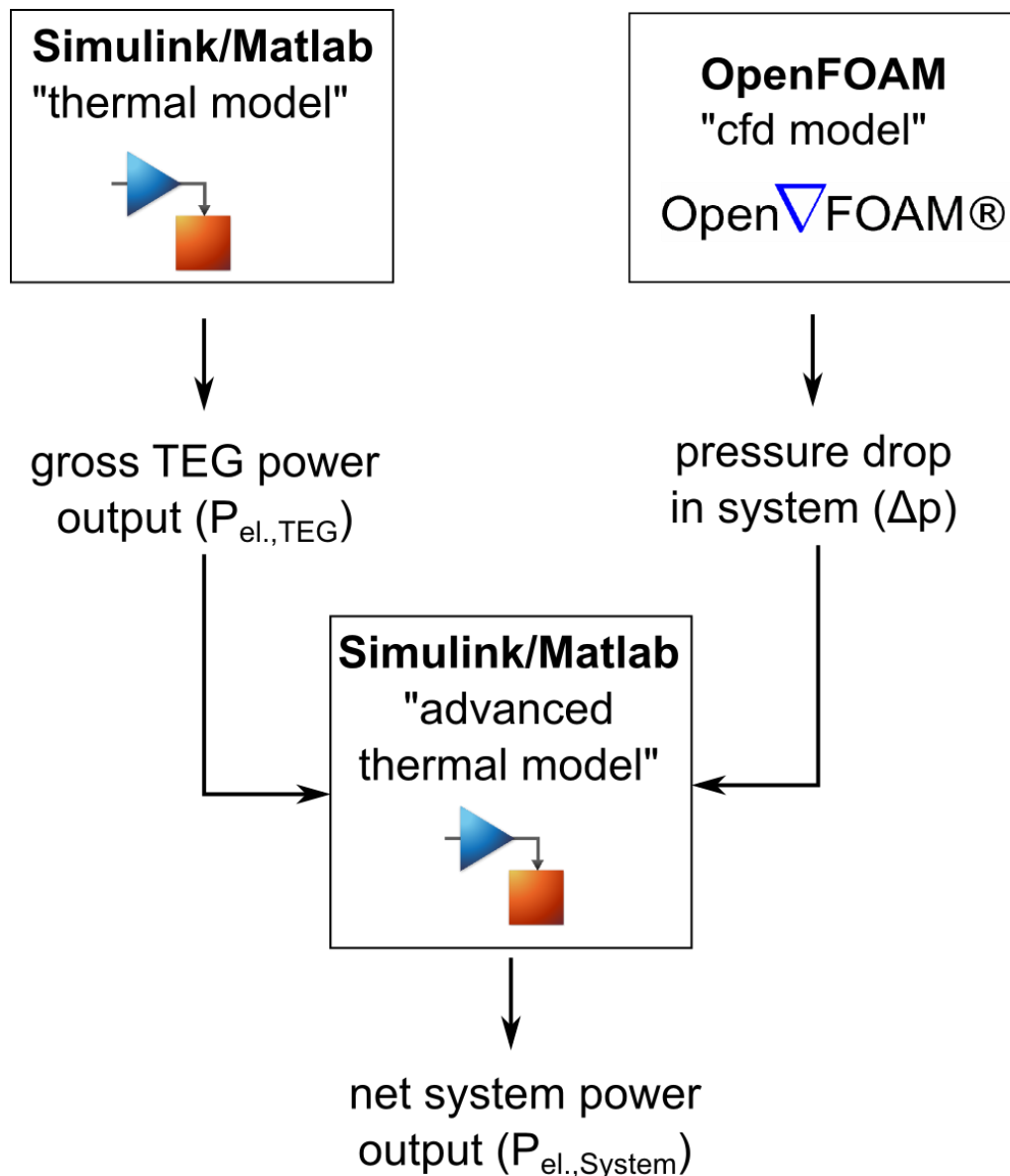


Figure 31 | Working principle of the shell script for simulation automation purpose



### 3.6 Merging the Models

In the final step of the work the results of the two models were brought together. The pressure drops were inserted in the thermal model in form of lookup tables. This then allowed the live monitoring of the systems operation parameters. With the help of the model it was possible to determine the optimal operation conditions and eventually the net power output and net efficiency of the system. Beyond that, the model offered the possibility to determine technological goals such as a target  $ZT$  value for the material development or lay the fundamentals for economic goals such as production costs. It shall be mentioned once more that the energy needed to bundle the heating fluid and make it available is not further considered in this work.



**Figure 32 | Simulation methodology**

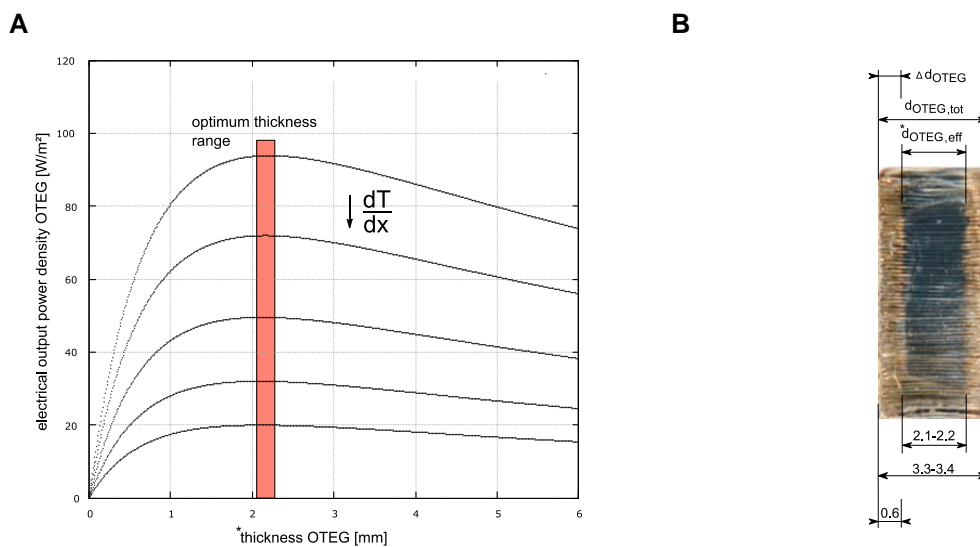
Since the waste heat is considered as waste the relevant parameter for which the system must be optimized is the net power output and not the net efficiency of the system.

## 4 Results

### 4.1 Thermal Model

#### 4.1.1 TEG Thickness

Resulting from the underlying optimization cycle the TEG thickness was determined. Note that the values from the thermal simulation do not represent the overall thickness of the TEG but only the active layer ( $d_{OTEG, eff}$ ). It is therefore necessary to add the inactive layer ( $\Delta d_{OTEG}$ ) on each side (**Figure 33, B**). **Figure 33, A** illustrates the optimal thickness of the OTEG ( $d_{OTEG, eff}$ ) that delivers the maximal TEG output power for every step. For this particular operation conditions, the optimal thickness of the TEG was found to stay within narrow bounds (2.13 mm and 2.21 mm) over a wide range of temperature gradients  $\frac{dT}{dx}$ . However, it was also found that the assumption that the maximum power point of the TEG would also represent the maximum power point of the system is misleading. At constant mass flow rates  $0.29 \text{ kgs}^{-1}$  (Mobiltherm 594),  $0.28 \text{ kgs}^{-1}$  (water) and a  $ZT$  value of ca. 0.52 the difference between the maximum power point of the TEG determined by the optimization cycle was found to be approximately 10 % below of the maximum net power output of the system at a constant thickness of 4 mm. This behavior might be explained as follows: A thinner TEG that works within its optimum in the front section of the system might alter the conditions for the following TEGs to come in the system in a way that leads to lower power output. The findings demonstrate that a simple MPP-tracking for the TEG does not represent a sufficient method to determine the optimal thickness of the TEG.



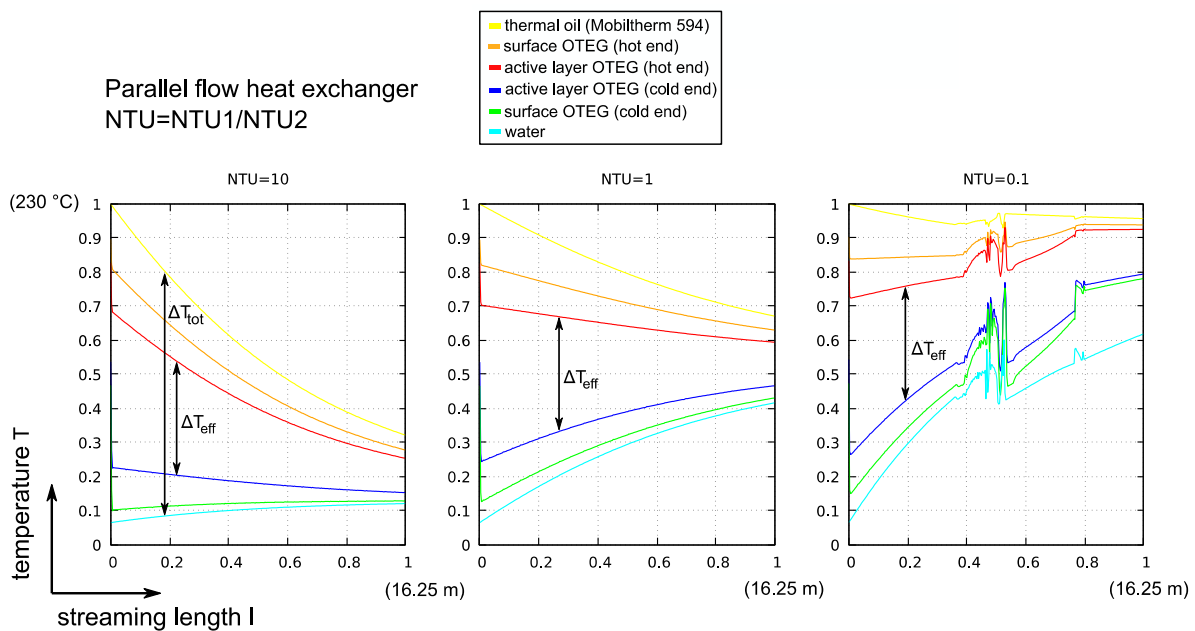
**Figure 33 | Optimum thickness of OTEG**

**A**, the optimal TEG thickness stays within tight borders, over a wide range of temperature gradients.

B, layer proportions of the TEG.

#### 4.1.2 Global Temperature Development

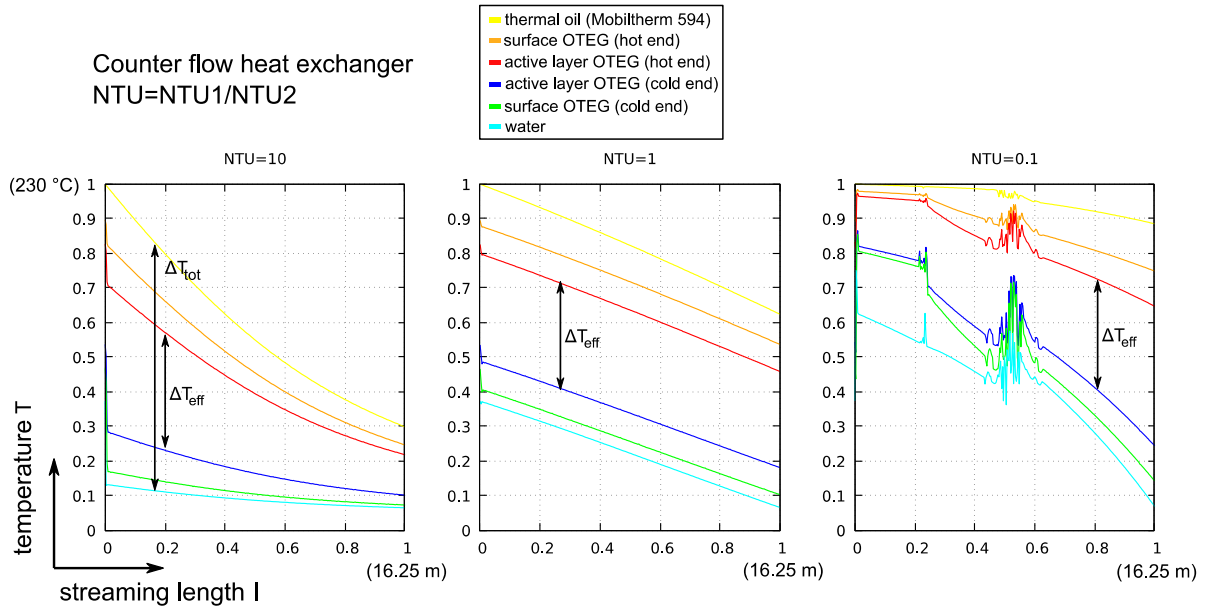
Equations (24), (25), (26) and (27) were implemented in Simulink to determine the temperature gradients over the streaming length for a parallel and a counter flow heat exchanger under steady state conditions. For different operation conditions, the temperature development was scoped over the course of the simulation. The results that the model delivers ( $NTU$  above 1) are thereby in good accordance with the expected behavior in both cases parallel flow (**Figure 34**) as well as counter flow (**Figure 35**) conditions. For low values of  $NTU$  the simulation collapses in the region, where a phase change of the cooling fluid medium occurs. In this case the inconsistencies occur in the cooling fluid around 100 °C at the vaporization temperature for water under ambient pressure. Since operation conditions under which the fluids undertake a phase transformation are not considered in the model this instability is acceptable.



**Figure 34 | Temperature development (parallel flow)**

Relative temperature development of a parallel flow heat exchanger for different Numbers of Transfer Units. For the parallel flow heat exchanger, higher temperature gradients in the front section of the system are expected than for counter flow conditions. This might lead to a higher power density of the system.

Also in accordance with the expectations, is the development of the effective temperature gradient. A rule of thumb for operation conditions of a TEG states, that at the active thermoelectric material approximately half of the total temperature difference between the heat source and sink applies, both **Figure 34** and **Figure 35** approximately illustrate this relation.

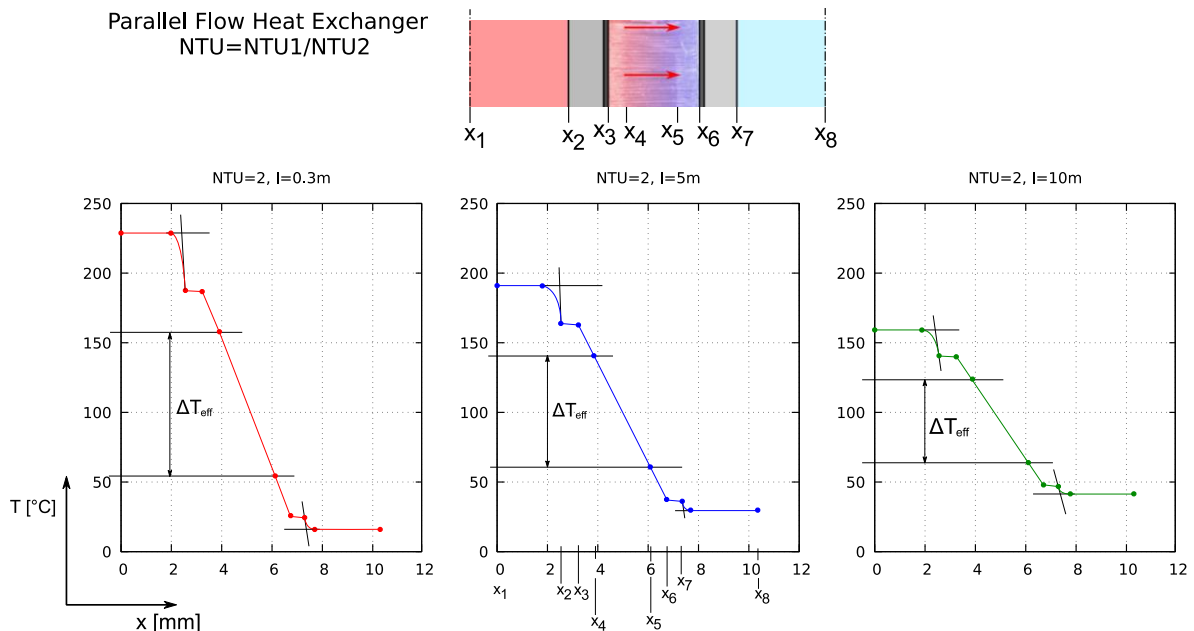


**Figure 35 | Temperature development (counter flow)**

Relative temperature development of a counter flow heat exchanger for different Numbers of Transfer Units. The disturbance visible for  $NTU=0.1$  is due to the phase change of water around 100 °C. Experience as well as the results from the simulation demonstrate that at the TEGs approximately half of the total temperature difference applies.

### 4.1.3 Local Temperature Development

**Figure 36** shows a high resolution of the temperature profiles through the heat exchanger wall for three locations along the streaming length.



**Figure 36 | Local temperature development (parallel flow)**

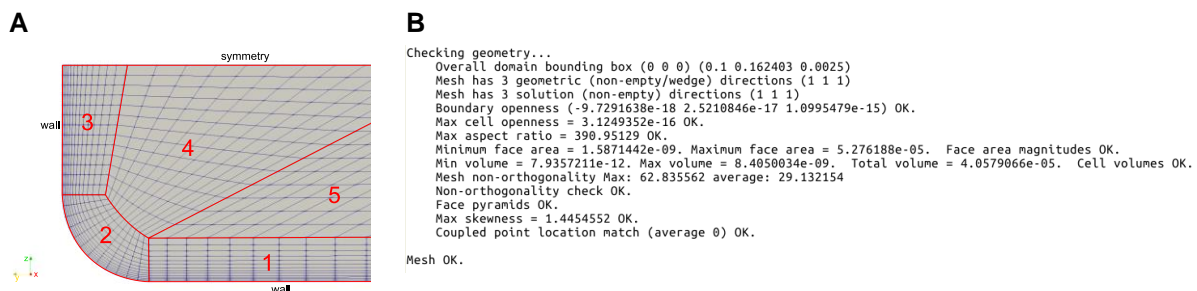
Local temperature development through the individual layers of the heat exchanger wall at 0.3 m, 5 m and 10 m streaming length.

## 4.2 CFD Model

### 4.2.1 Meshing

#### blockMesh

A section of the mesh created by the utility *blockMesh* is shown in **Figure 37, A**. The different blocks are framed with red lines. In flow direction (x-direction), there are only few cells defined since a high resolution in this direction is not necessary. To determine mesh quality parameters the utility *checkMesh* was executed. The output of the utility is illustrated at the **Figure 37, B**. Of special interest are thereby the values for Mesh non-orthogonality, Max skewness and Max aspects ratios. For all three parameters, the objective is to lower the values as far as possible. Although *checkMesh* states that the Mesh is OK this does not necessarily mean that the mesh will deliver good results. For the simulation of the channel flow the mesh quality was found to be sufficient.



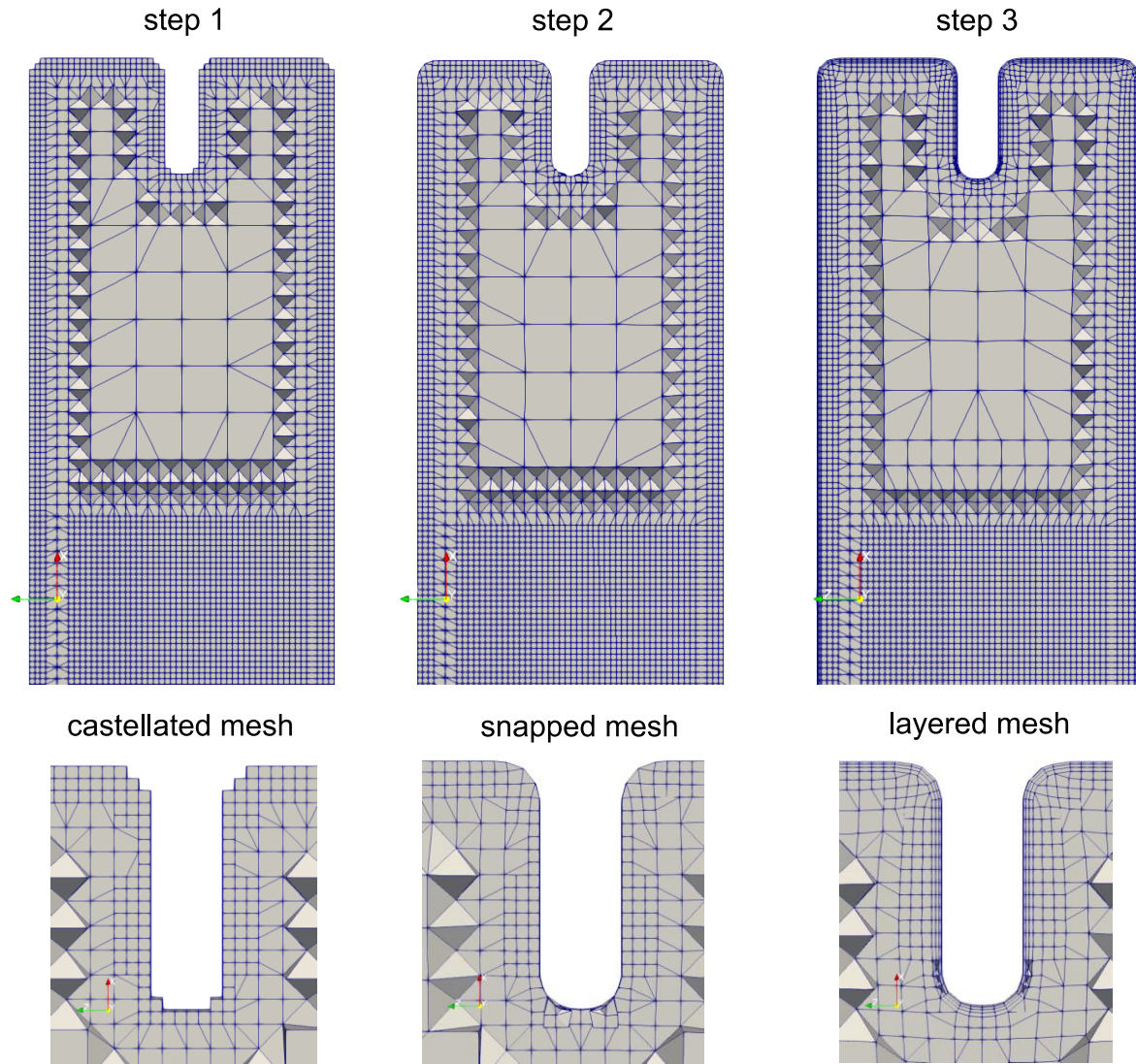
**Figure 37 | Results *blockMesh* and *checkMesh***

**A**, front view on the characteristic section of the channel flow inlet. For block 1, 2 and 3 a simple grading towards the walls was chosen to have a higher resolution in this region. **B**, results from the *checkMesh* utility.

#### snappyHexMesh

The *snappyHexMesh* utility was found to be very efficient for creating the mesh from the bypass flow geometry. It enables the creation of large computational meshes on conventional workstations (the specifications of the used workstation are found in the appendix). Although *snappyHexMesh* introduces a different approach of creating meshes through automation, it was found that the familiarization phase for the utility took much time. Due to various adjustment possibilities and the lack of an interactive user interface, the operation of the utility eventually comes down to try-and-error. Problematic geometries for the application are sharp edges and corners. Here *snappyHexMesh* often fails to follow the model geometry. To still deliver acceptable results, the problematic regions were smoothened in the CAD model and afterwards refined in the *snappyHexMesh* utility. In the U-turn where the highest gradients were expected the region was further refined by placing a

cylinder in this region with a high resolution (level 4). In the end, the mesh consisted of 8,236,312 cells. Running *snappyHexMesh* in parallel on four cores using the *scotch* method to distribute the geometrical model in four parts ended up in a total simulation time of 1777.97 s (ca. 30 min). **Figure 38** illustrates the results of the individual *snappyHexMesh* phases on a characteristic section.



**Figure 38 | Meshing results *snappyHexMesh***

Illustration of a section of the bypass flow after every sub routine of the *snappyHexMesh* utility. All in all, the bypass flow was meshed around 30 times to find acceptable configurations for the utility.

To determine the overall quality of the mesh, the *checkMesh* routine was applied. In **Table 9** three important quality parameters and their allowed thresholds are summarized for the individual steps.

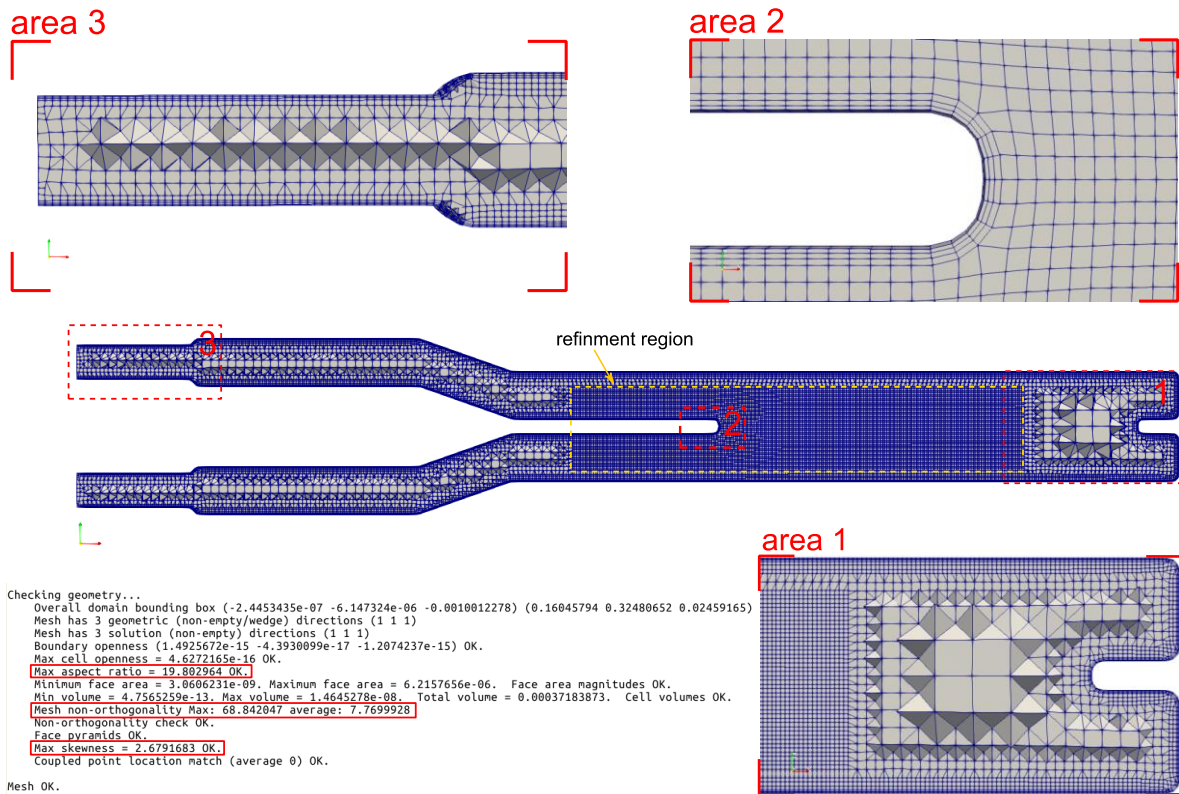


**Table 9 | Results of checkMesh for the bypass flow created with *snappyHexMesh***

Routine	cells	time	max. aspect ratio	max. non-orthogonality	max. skewness
castellatedMesh	4,565,437	489.87s	1.0361704	25.897983 average: 9.0379446	1.0000005
snap	4,565,437	493.78s	5.3076481	53.602623 average: 9.3937756	2.6791683
addLayers	8,236,312	784.7s	19.802964	68.842047 average: 7.7699928	2.6791683
thresholds	/	/	20	75	15

**Table 10 | Results for the *addLayers* step of *snappyHexMesh***

patches	faces	layers	overall thickness	Success rate
wallPipe	5,763	2.25	0.000175 m	74.3 %
wallInside	556,707	2.96	0.000232 m	98.5 %
wallOutside	620,889	2.96	0.000233 m	98.6 %
wallSide	74,917	2.28	0.000175 m	74.2 %
set-up	/	3	0.0001 (expansion ratio 1.3)	

**Figure 39 | Results *snappyHexMesh***

Meshing results after *snappyHexMesh* for the bypass flow. Presented is a slice of the bypass flow

(see **Figure 15**). To allow a higher resolution around the flow deflection a refinement region was defined in the U-turn.

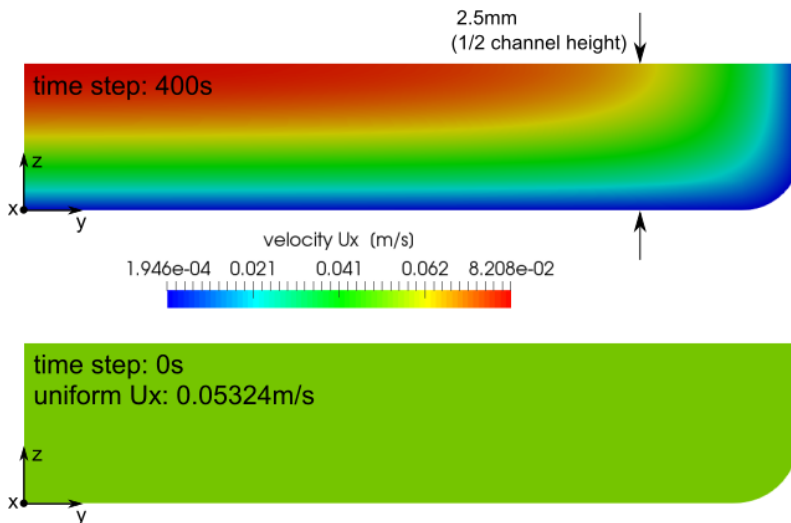
An additional indicator that delivers information about the overall mesh quality is the success of the third step (*addLayers*). If this step has low success rates, the reason can often be traced back either to an insufficient mesh from the previous two steps, sharp geometries or quality thresholds that only allow small tolerance (*meshQualityDict*). **Table 10** shows the results from the layering step. There is still potential for improvement of the layering especially at the side surface of the bypass channel as well as for the surface connecting the two channels. Still most of the surface faces are layered to a sufficient degree.

#### 4.2.2 Boundary Conditions and Initial Conditions

Of special interest during the work was the correct definition of the initial conditions at the inlet of the two flows. The results of the three methods: *mapping*, *swak4Foam* and *interpolation2DTable* for the inlet are presented in the following.

##### Mapping

The mapping utility was found to be the most convenient method regarding time invest. Since this method is pre-defined in OpenFOAM the implementation was simple. Using this routine, a fully developed flow could be simulated on relatively short segment (100 mm) of the channel flow. However, it was only applicable for the simple channel flow.



**Figure 40 | *mapping***

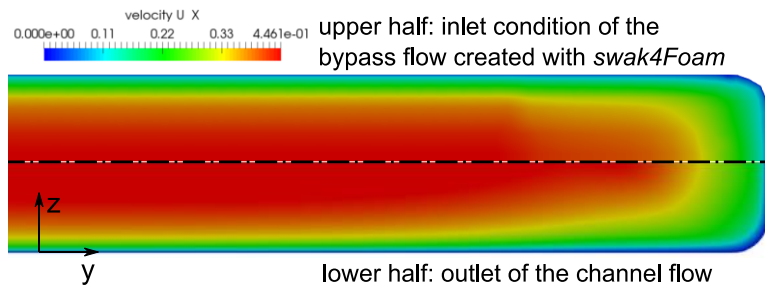
Result of the mapping method for the inlet of the channel flow. After defining initially, a uniform velocity field with  $u_x = 0.05324 \text{ ms}^{-1}$  after 400 s there is a fully developed flow at the inlet.

##### swak4Foam

To determine the velocity profiles of the channel flow for turbulent and laminar flow conditions the velocity component  $u_x$  was detected over the channel height near the outlet of



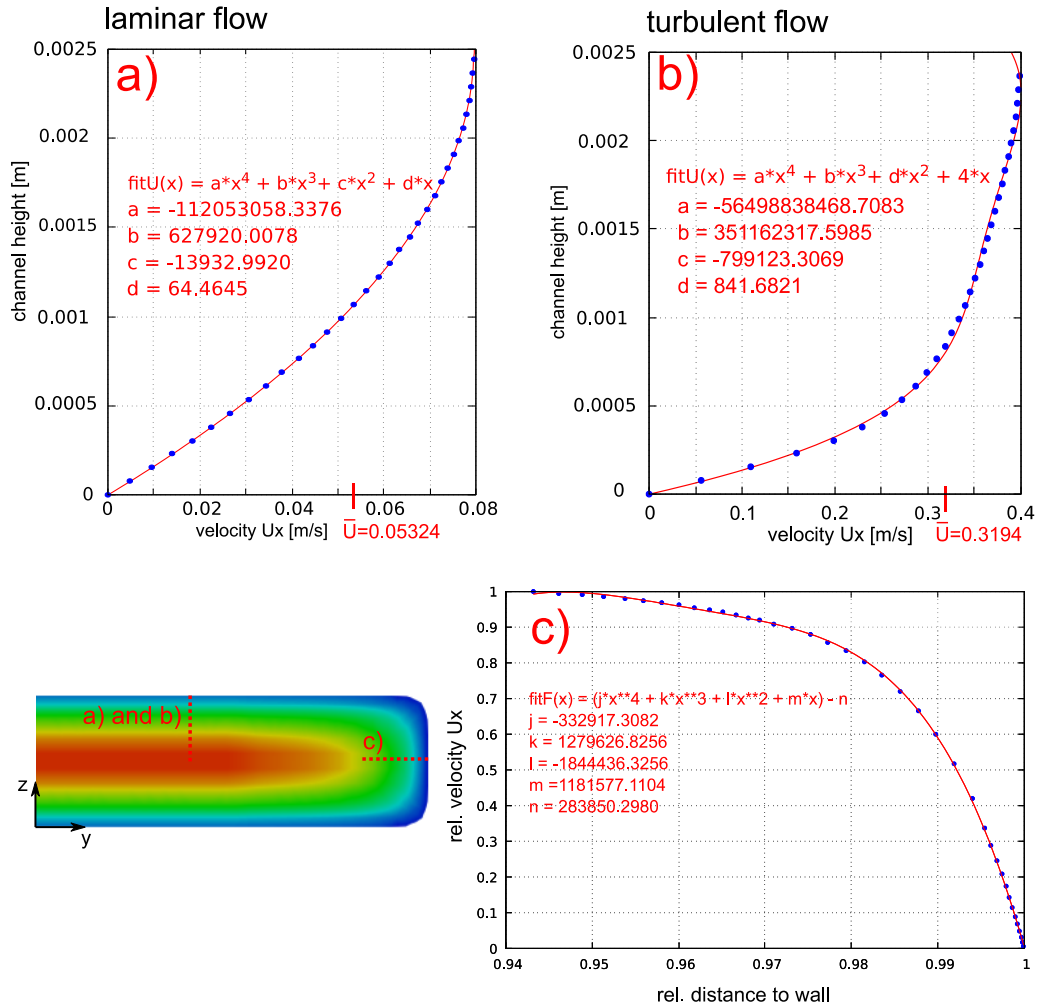
the channel flow. With gnuplot the velocity profile in flow directions ( $u_x$ ) was fitted with a polynomial function of the 4<sup>th</sup> order. The result of the fit was then used by the utility *swak4Foam* to setup the inlet conditions for the next bypass flow simulation. **Figure 42** illustrates this procedure; for the central part of the channel the velocity is pre-set according to the profile  $u_x(z)$ . To also respect the velocity distribution at the edges the relative velocity development near the side walls was detected. The produced fits from gnuplot have shown good agreement with the measured values. All in all, the *swak4Foam* utility *groovyBC* delivered good results for setting the outlet conditions of the velocity from the channel flow as inlet conditions for the bypass flow (**Figure 41**).



**Figure 41 | *swak4Foam* and *groovyBC***

Modeling of the inlet velocity with *groovyBC*. The modelled flow field (upper half) reflects closely the velocity field leaving the outlet of the channel flow.

While *swak4Foam* delivered good results for the velocity components, creating the initial conditions for the turbulent flow components ( $k$  and  $\varepsilon$ ) wasn't done with this utility. In contrary to the velocity the development of the turbulence parameters could not be fitted with enough accuracy using a single function. Besides that, *swak4Foam* does not offer a 2-dimensional lookup table to define the values depending on their  $y$ - and  $z$ -coordinate.



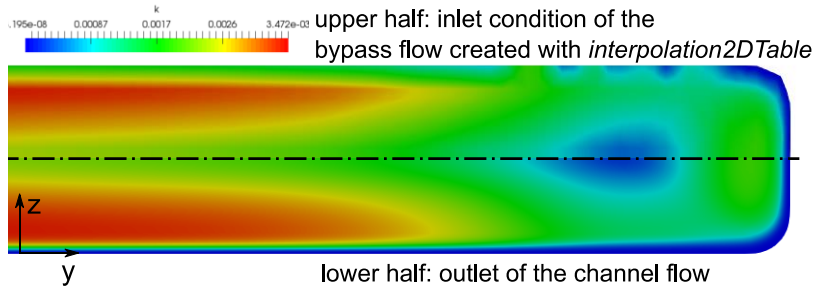
**Figure 42 | Velocity profiles**

Fits of the velocity field along different sections of the channel flow outlet. The shell script automatically takes these results and sets up the initial conditions for the bypass flow inlet for the following simulation.

## Lookup2DTables

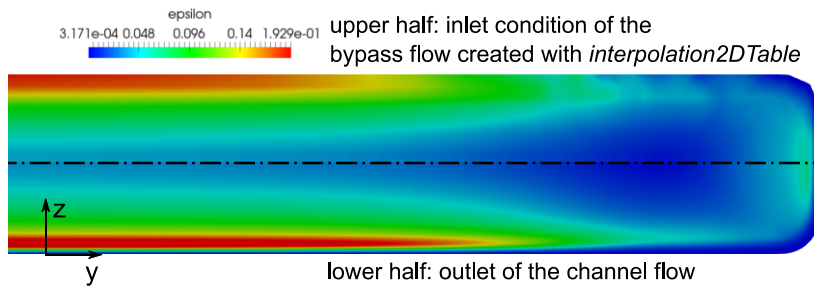
Due to the open source character of OpenFOAM, a lookup table for the turbulent flow parameters could be created by the utilization of the C++ class *interpolation2DTable* that is predefined in OpenFOAM. As the name indicates the class would create a 2-dimensional lookup table with interpolation in-between its values. Prior to its utilization a shell script was written that would sort the values and order them in the required manner dictated by the *interpolation2DTable* class. Following every simulation of the channel flow with the turbulence model the lookup tables for  $k$  and  $\varepsilon$  were created for later use in the bypass flow simulation. An exemplary figure that demonstrates the problem of this method is demonstrated in **Figure 43**. There are clear inconsistencies visible between the implementation (upper half) and the desired state (lower half). Especially near the walls the value should converge to zero which was not achieved. For the dissipative rate of the turbulent kinetic energy  $\varepsilon$  the results behave similar (**Figure 44**) while for the center of the

channel the method delivers good results the areas with high gradients are problematic. Despite this inconsistency, the lookup tables were used to set up the inlet conditions since it is believed that this difference would only have minor influence on the pressure drop and be corrected by the boundary conditions itself in the first section of the bypass flow.



**Figure 43 | interpolation2DTable k**

There are clear inconsistencies for initial setup of the turbulent kinetic energy  $k$  (upper half) compared to the desired state (lower half). A higher resolution in the lookup table might improve the quality of the method.



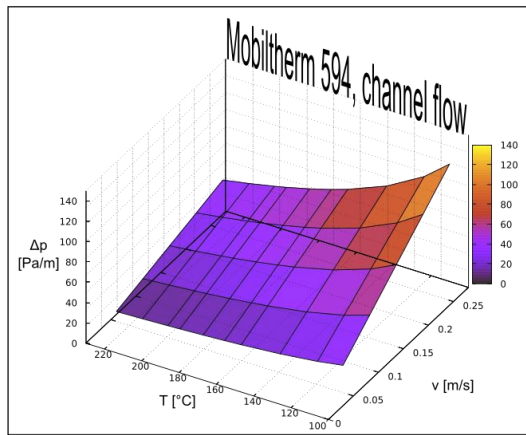
**Figure 44 | interpolation2DTable epsilon**

In areas of high gradients, the method using the *interpolation2DTable* fails. For the dissipation rate of the turbulent kinetic energy  $\epsilon$  it is even more obvious, like the turbulent kinetic energy  $k$  the values at the wall should drop to zero.

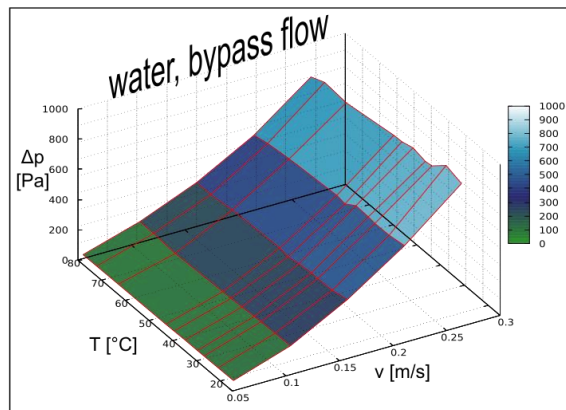
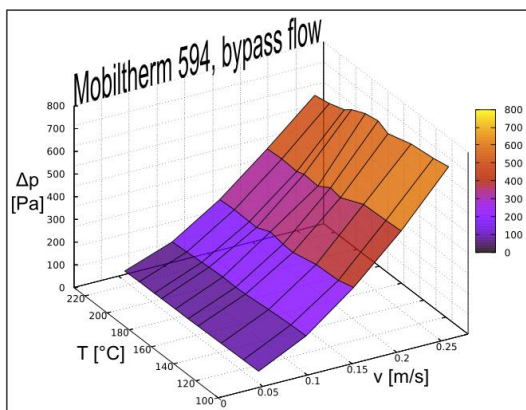
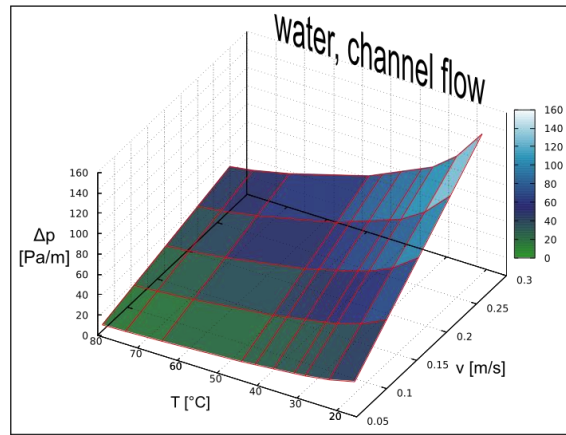
### 4.2.3 Pressure Drop

The objective of the fluid dynamical simulation eventually was to determine the dissipative energy loss in the heat exchanger by measuring the pressure drop over the streaming length. **Figure 45** and **Figure 46** show the results acquired from the two channels. Here the pressure difference  $\Delta p$  ( $\text{Pa m}^{-1}$  and Pa) between the inlet and the outlet is described by the flow velocity  $u$  ( $\text{ms}^{-1}$ ) and temperature  $T$  ( $^{\circ}\text{C}$ ) of the two fluids. As expected the pressure drop in the bypass flow is significantly higher as in the simple channel flow. Also, the temperature dependence of the pressure drop is small compared to the influence of the flow velocity.

A



B

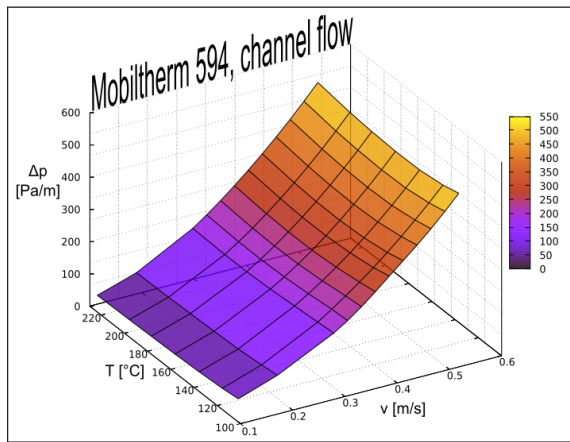


**Figure 45 | Pressure drop ( $\Delta p$ ) for laminar flow conditions**

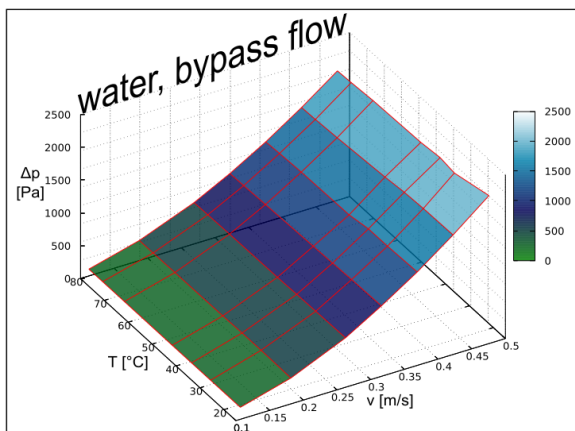
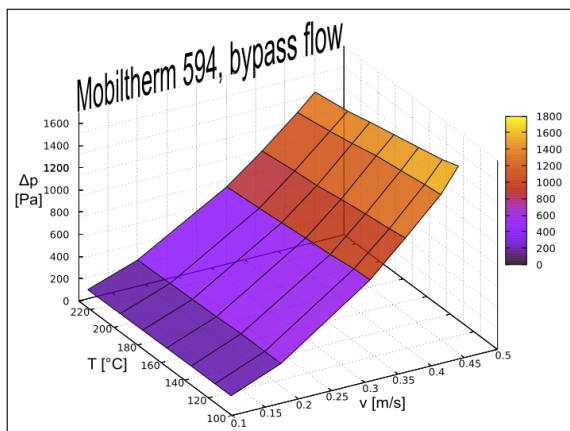
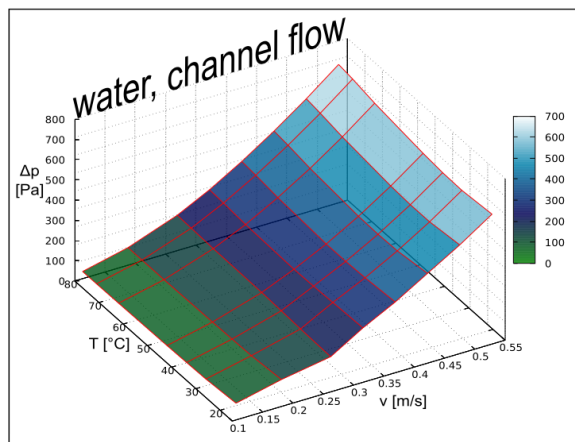
Pressure difference between inlet and outlet for the channel and bypass flow for the working fluids Mobiltherm594 (A) and water (B). The temperatures range from 106 °C to 227 °C for thermal oil and 15 °C to 80 °C for water. The flow conditions are assumed laminar for velocities from 0.05 m/s to 0.2662 m/s.

Different to the laminar channel flow in **Figure 45**, the results show a parabolic increase  $\sim U^2$  of the pressure drop. In the literature, this relation between pressure drop in a hydraulic component and the flow velocity is often mentioned. For the simulations implemented with a turbulence model (**Figure 46**) higher pressure drops were expected owing to the larger effective viscosity introduced by  $\nu_t$ . For small flow velocities where the flow would be in the transitions zone in between laminar and turbulent conditions, the RANS modulation do not necessarily deliver valid results. Originally the used turbulence model was developed for pure turbulent conditions. A transition zone model could hereby deliver more precise values but since the objective laid in a first approximation the results were used anyway.

A



B

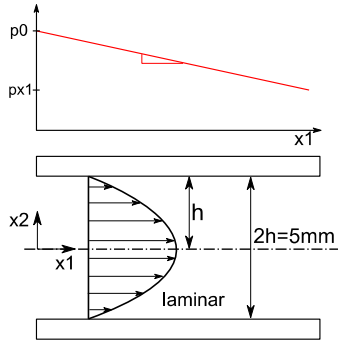


**Figure 46 | Pressure drop ( $\Delta p$ ) for turbulent flow conditions**

Pressure difference between inlet and outlet for the channel and bypass flow for the working fluids Mobiltherm 594 (A) and water (B). The temperatures range from 106 °C to 227 °C for thermal oil and 15 °C to 80 °C for water. The flow conditions are turbulent for velocities from 0.011 ms<sup>-1</sup> to 0.4873 ms<sup>-1</sup>.

#### 4.2.4 Validation of the Fluid Dynamical Model

To eventually validate the results from the CFD model it is common practice to compare the acquired data with measured results from an experimental setup. Due to the lack of an experimental setup this was not possible, however for the simple case of a laminar, incompressible, steady parallel plate duct flow that is fully developed in flow direction there exists an analytical solution.



**Figure 47 | Effective Temperature Difference**

Schematic drawing of the velocity development of an infinite parallel plate duct. For this simplified scenario, the velocity profile and the pressure difference along flow direction can be determined analytically.

It can be derived from the general form of the incompressible NSE in cartesian coordinates and the mass continuity equation by using the simplifications:

- 2D:  $\frac{\partial(\dots)}{\partial x_3} = 0, u_3 = 0$
- steady:  $\frac{\partial(\dots)}{\partial t} = 0$
- horizontal channel:  $\vec{f} = 0$
- fully developed in flow direction:  $\frac{\partial(\vec{u})}{\partial x_1} = 0$

By applying these simplifications on the general form of the equations:

$$\text{(mass conservation)} \quad \frac{\partial f}{\partial t} + \nabla(\rho \vec{u}) = 0 \quad (15)$$

$$\text{(momentum conservation)} \quad \rho \left( \frac{\partial \vec{u}}{\partial t} + \vec{u}(\nabla \vec{u}) \right) = \rho \vec{f} - \Delta p + \mu \nabla^2 \vec{u} \quad (16)$$

reduces to:

$$\text{(mass conservation)} \quad \frac{\partial u_2}{\partial x_2} = 0 \quad (37)$$

$$\text{(momentum conservation)} \quad \vec{u}(\nabla \vec{u}) = \frac{-\Delta p}{\rho} + \eta \nabla^2 \vec{u} \quad (38)$$

Solving the mass conversation equation for the given setup with the boundary condition  $u_2(h)=0$  it follows that  $u_2=0$ . Using this in the momentum conversion further simplifies the three equations in  $x_1$ -,  $x_2$ - and  $x_3$ -direction to:

$$\text{x1-direction} \quad 0 = -\frac{\partial p}{\rho \partial x_1} + \vartheta \frac{\partial^2 u_1}{\partial x_2^2} \quad (39)$$

$$\text{x2-direction} \quad 0 = -\frac{\partial p}{\partial x_2} \quad (40)$$

$$\text{x3-direction} \quad 0 = 0 \quad (41)$$

The pressure drop only evolves in  $x_1$  and is assumed constant in flow direction ( $\frac{\partial p}{\partial x_1} = \text{const. } \frac{dp}{dx_1}$ ). By integrating equation (39) the pressure difference can be described by:

$$\Delta p(x_1, x_2) = -\frac{2\rho v u_1(x_2)}{h^2 - x_2^2} \cdot \int_0^{x_1} dx_1 \quad (42)$$

The linear slope of the solution is visible in **Figure 45** for the channel flows. Also comparing the two solutions directly shows good conformance of the numerical and the analytical results (**Table 11**).

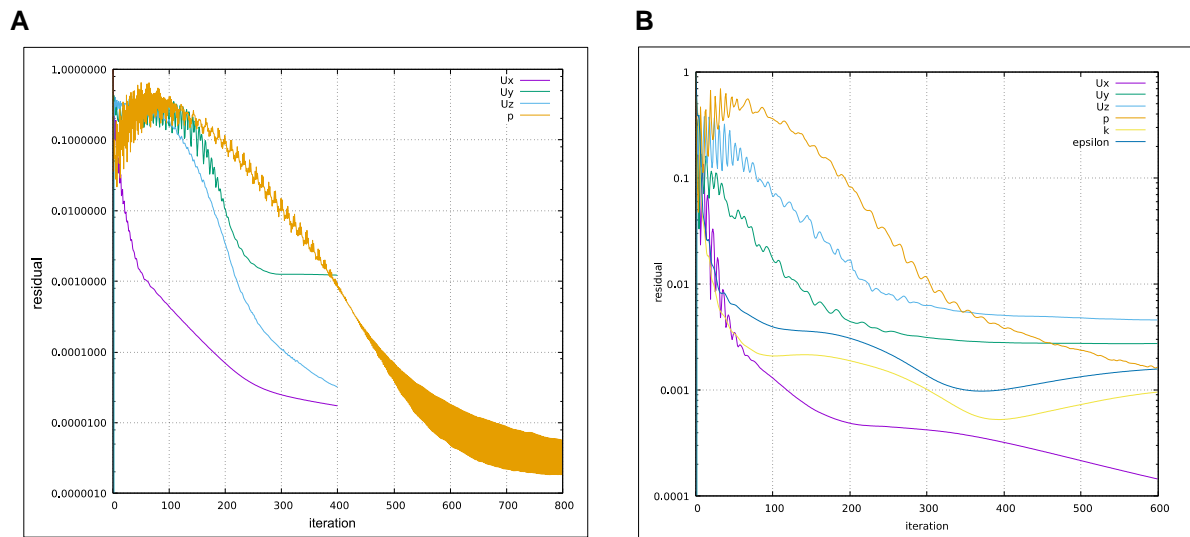
**Table 11 | Comparison numerical and analytical solutions (channel flow)**  
pressure drop [Pam<sup>-1</sup>]

velocity [ms <sup>-1</sup> ]	temperature [°C]	numerical	analytical
.05324	64.42125711	11.5783877390	11.9874921258
.05324	73.51959887	10.0007345320	10.3524396050
.05324	79.28474535	9.32098587330	9.6479192223
.10648	64.42125711	23.1548244010	23.9520113042
.10648	73.51959887	19.9994475370	20.6843853114
.10648	79.28474535	18.6399622730	19.2767616737
.15972	64.42125711	34.7296825520	35.9080437569
.15972	73.51959887	29.9973390510	31.0094563173
.15972	79.28474535	27.9585105710	28.8997306156
.21296	64.42125711	46.3058270420	47.8554839657
.21296	73.51959887	39.9978579670	41.3345273231
.21296	79.28474535	37.2808198280	38.5201511336
.26620	64.42125711	57.8867315746	59.8092553016
.26620	73.51959887	50.0067000770	51.6476196843
.26620	79.28474535	46.6129499570	48.1308633712



### Monitoring of the CFD Solutions

To allow an estimation of the solution of the CFD simulation different parameters were monitored. For time, independent simulations the residual of the simulation can give evidence of the progress of the solution **Figure 48**. However, in practice the residual is a difficult parameter to interpret. It allows an estimation about the numerical schemes rather than on the physical correctness. For this reason, besides the residual other flow parameters were scoped, here the pressure difference between the inlet and outlet was monitored over the iteration time. When the values would converge close enough on a value the simulation was stopped, the allowed discrepancy was estimated roughly. For the simple channel flow the monitoring of the residual delivered good to acceptable results for the laminar (**Figure 48, A**) and turbulent (**Figure 48, B**) conditions.

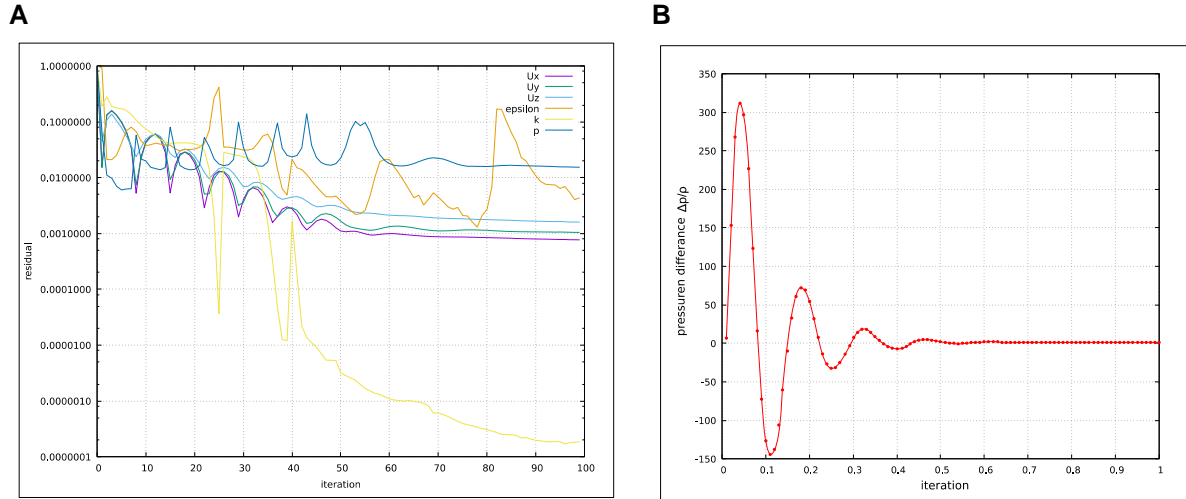


**Figure 48 | Simulation monitoring of the residuals (channel flow)**

Residuals for the channel flow for the laminar conditions (**A**) and turbulent flow conditions (**B**).

For the bypass flow, monitoring the residuals of the solution was not successful (**Figure 49, A**), therefore the relative pressure difference was observed instead (**Figure 49, B**). The convergence of the pressure drop is obvious and was reached for all the bypass flow simulation. Although the solution still fluctuates on the third decimal, the values were used. The solutions were judged as good enough for a first evaluation. In the field of flow simulation there is still seen room for improvement.





**Figure 49 | Simulation monitoring (bypass flow)**

**A**, residual plot for the turbulent flow in the bypass flow. Since the fluctuation of residuals do not allow a direct assumption about the progress of the simulation, additionally direct monitoring of the pressure difference of the inlet of outlet was conducted, **B**. The simulation was stopped when the results would only vary on the 3<sup>rd</sup> decimal place.

## 4.3 Merging the Models

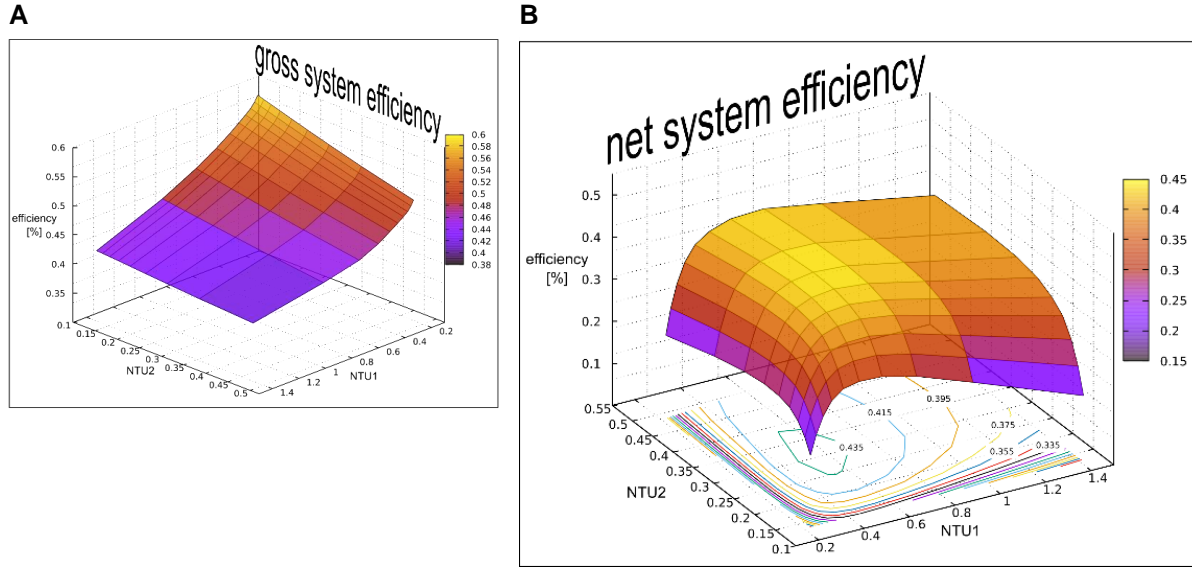
In the end, the real achievement of the thesis work was found to be rather than the development of a mature WHRS based on OTEGs, the development of a methodology that allows the estimation of the technical potential of such a system. By including the results from the OpenFOAM simulation (**Figure 45** and **Figure 46**) into the Simulink model live monitoring of the net efficiency and net power generation was possible.

Rather than the efficiency the net power output is the key factor determining the potential of the system since the heat is generally considered as waste which is present anyway. While there are many parameters that are dictated by the case scenario such as heat exchanger surface, fluids, temperature levels, etc. The system has some operational parameters which need to be optimized:

- mass flow rate (heating)  $\longrightarrow NTU_1$
- mass flow rate (cooling)  $\longrightarrow NTU_2$
- TEG thickness  $\longrightarrow d_{eff}$
- ZT value  $\longrightarrow \overline{ZT}$

Now with the help of the extension of the thermal model it is possible to determine the optimal operation conditions at which the net power output of the system will become maximal.

First, it was possible to demonstrate that for a  $ZT$  value of 0.1 which is a value that is currently achievable and a constant effective TEG thickness of 2.5 mm surplus electrical energy could be generated by the system. **Figure 50** shows the results for the efficiencies depending on the values of  $NTU_1$  (heating channel) and  $NTU_2$  (cooling channel). While **Figure 50, A** demonstrates that efficiency of the generator steadily increases with  $NTU_1$  and  $NTU_2$  becoming minimal, **Figure 50, B** on the contrary demonstrates that there is an optimal operation point at which the net efficiency of the WHRS becomes maximal.

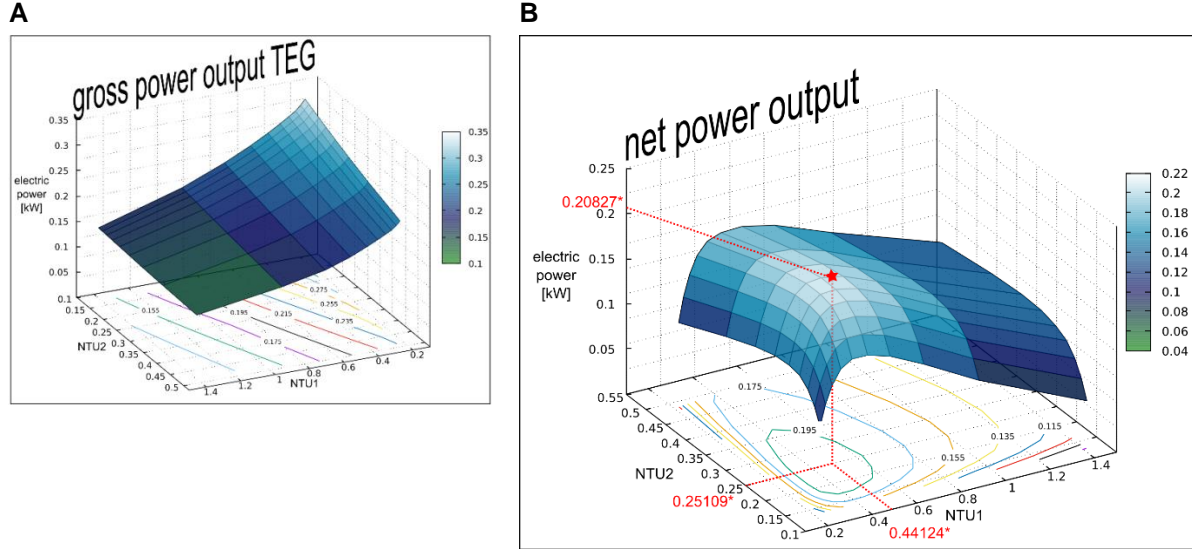


**Figure 50 | Efficiencies of the WHRS ( $ZT \approx 0.1$ )**

Total TEG efficiency depending on the NTU of the heating and cooling channel (**A**). **B**, demonstrates the net system efficiency of the proposed WHRS for a  $ZT \approx 0.1$  and constant  $d_{eff}$  of 2.5 mm.

That the generator efficiency increases with  $NTU_1$  and  $NTU_2$  dropping can be explained by the fact that the mass flow rate finds itself in the denominator of NTU (see (21)). An increase in mass flow rate which eventually leads to an increase of the fluid velocity, results in fewer time that the fluid is in contact with the heat exchanger surface and therefor a higher exiting temperature for the heating fluid and lower exiting temperature for the cooling fluid. These conditions signify optimal operation conditions for the TEG. However, with increasing fluid velocities the dissipated energy in the heat exchanger increases; an optimal operation point can be found. The optimal operation conditions for a system using generators with  $ZT$  of 0.095 is illustrated in **Figure 50**. The contour lines illustrate the optimal operation conditions for a maximum net efficiency is found near  $NTU_1=0.55$  and  $NTU_2=0.3$  with a maximal efficiency around 0.435 %. The pump efficiency was set to 30 % and the effective TEG thickness kept constant at 2.5 mm. Even more relevant than the net efficiencies are the results for the net power output of the system. When power fuel consumption is not an influential factor, as it usually is not when one is talking about waste heat, the maximum net power output represents the key value that needs to be found. From monitoring the system

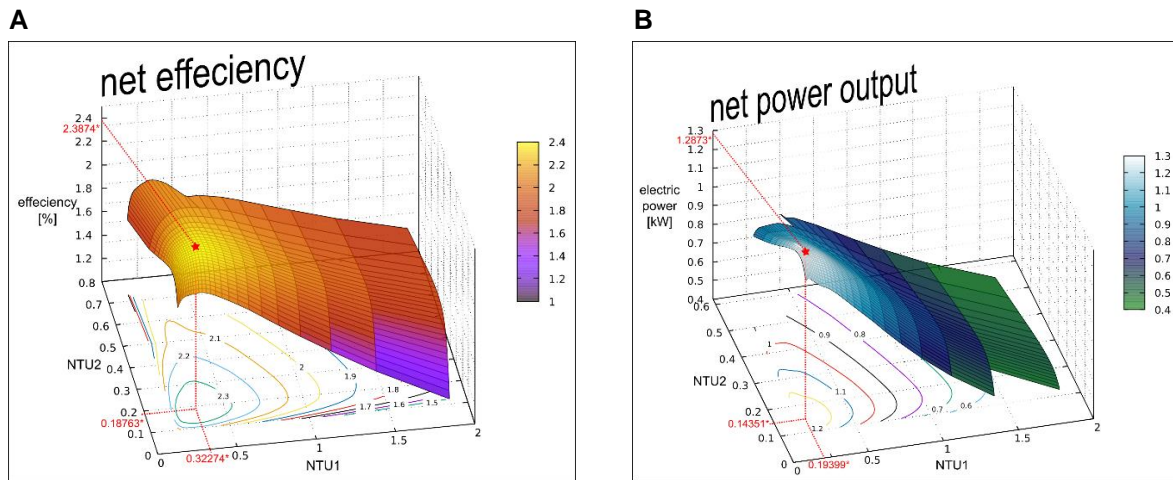
power output (**Figure 51**) there is a similar behavior visible as for the efficiencies with maximal values up to 0.2 kW. The contour lines also illustrate that operation conditions for maximum net efficiency do not match the operation conditions for the maximum net power output.



**Figure 51 | Power output of the WHRS ( $ZT \approx 0.1$ )**

Gross electric output power depending on  $NTU$  (**A**). **B**, shows maximal net electric output power of the system for a  $ZT \approx 0.1$  and constant  $d_{eff}$  of 2.5 mm.

With an increase in TEG efficiency we observe the movement of the optimal operation point towards lower  $NTU$  values. This behavior is seen in **Figure 52**. Here the WHRS was simulated with an  $ZT \approx 0.5$  and effective TEG thickness of 2.5 mm. It was found that the elaborated model delivers valid results up to a  $ZT$  value of 0.8. This limitation is due to the fact that beyond  $ZT \approx 0.8$  the fluid velocities are leaving the range for which data from the CFD simulation is available.

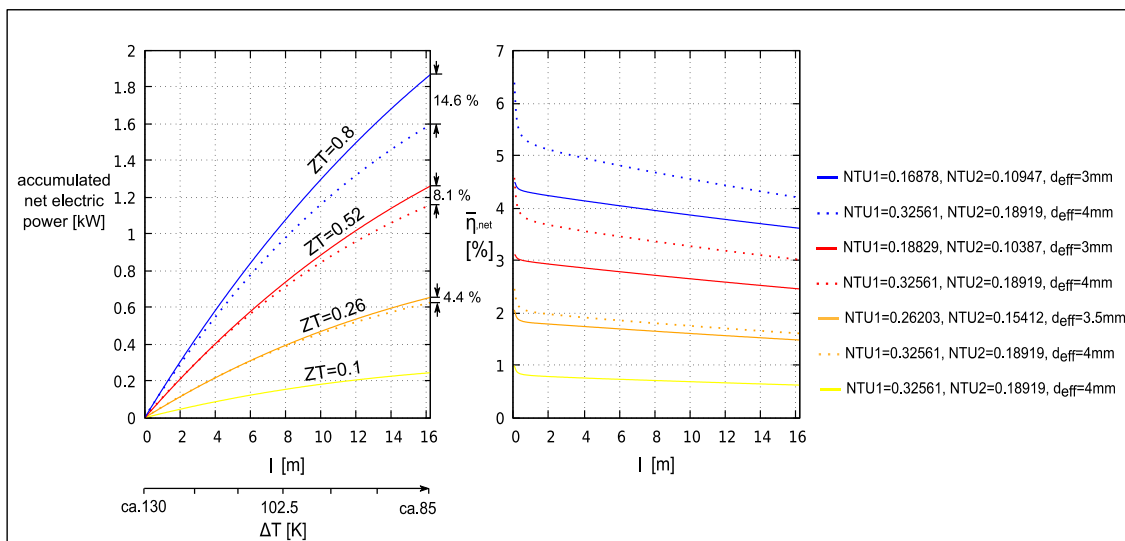


**Figure 52 | Power output and efficiency of the WHRS ( $ZT \approx 0.5$ )**

Performance data of the WHRS for a  $ZT \approx 0.5$  and constant  $d_{eff}$  of 2.5 mm. **A**, hereby shows the net efficiency and **B**, the maximal net electric output power.

Until this point the TEG thickness was kept constant and optimization was only done for the two values of  $NTU_1$  and  $NTU_2$ . The simulation results revealed that the optimal ratio between these two parameters are found in a relatively narrow area of 1.5 to 1.8. This meant a relatively important restriction of the optimization area with which the simulation time was considerably shortened.

During the further procedure, the TEG thickness as a variable parameter was also included in the final optimization process. The net efficiency of the WHRS would hereby steadily increase with a growth of the TEG thickness which is logical since a thickening of the TEG would lead to a higher temperature gradient recoverable by the TEG. While the net efficiency showed this behavior, it was found that the net power output would form a maximum at a certain TEG thickness depending strongly on the conversion efficiency of the generator. With increasing conversion efficiency, the TEG would become thinner. **Figure 53** illustrates the final results of this work. It shows the achievable net electric power delivered by the WHRS for given setup introduced over the course of this work. For presently achievable values of  $ZT$  of 0.1 the simulation came to the result that at optimal operation conditions ( $NTU_1=0.32561$ ,  $NTU_2=0.18919$ ) and an ideal effective TEG thickness of 4 mm, the maximal achievable output power would be of the value of 246 Watts. Furthermore, it was identified that operating the WHRS in the optimum for  $ZT$  of 0.1 using enhanced TEGs with higher conversion efficiencies would lead to results that lie clearly below the achievable. These differences are indicated and quantified by the dotted lines in **Figure 53**. When determining the technological potential of a WHRS this must be taken into account, otherwise the potential might be underestimated.



**Figure 53 | Potential of the WHRS for different  $ZT$  values**

Net electric power, net conversion efficiency for different  $ZT$  values over the streaming length of the WHRS. A summary of the specifications and the explicit values of the WHRS operating under ideal conditions can be found in **Table 12** and **Table 13**.

**Table 12 | Specifications of the WHRS I**

Constant parameters	description
Heat carriers	Liquid water (15 °C), liquid Mobiltherm 594 (230 °C)
Dimensions ( $L \times W \times H$ )	1.2 m x 0.4 m x 1.2 m
Streaming length $L$	16.25 m
Exchanger surface	4.875 m <sup>2</sup>
Channel ( $L \times W \times H$ )	0.6 m x 0.3 m x 0.005 m
Pump efficiency	30%

**Table 13 | Specifications of the WHRS II**

\*Note that changing values along the streaming length are taken from the end of the WHRS ( $L=16.25\text{m}$ ).

<b>ZT value</b>	<b>0.1053</b>	<b>0.2637</b>	<b>0.5275</b>	<b>0.798</b>
<b>Effective TEG thickness (optimized)</b>	<b>4 mm</b>	<b>3.5 mm</b>	<b>3 mm</b>	<b>3 mm</b>
<b>*NTU<sub>1</sub> (optimized)</b>	<b>0.3526</b>	<b>0.2622</b>	<b>0.1883</b>	<b>0.1688</b>
<b>*NTU<sub>2</sub> (optimized)</b>	<b>0.18992</b>	<b>0.1541</b>	<b>0.1039</b>	<b>0.1095</b>
$NTU_1/NTU_2$	1.721	1.701	1.813	1.542
Net power output	0.2463 kW	0.6559 kW	1.264 kW	1.872 kW
Total heat flux	39.44 kW	44.24 kW	51.46 kW	51.83 kW
Net power density	50.5 Wm <sup>-2</sup>	134.5 Wm <sup>-2</sup>	259.3 Wm <sup>-2</sup>	384.2 Wm <sup>-2</sup>
Net conversion efficiency	0.6245 %	1.482 %	2.457 %	3.612 %
Average carnot efficiency	24.861 %	24.71 %	24.9 %	25.2 %
*Volume flow (Mobiltherm 594)	1.398 m <sup>3</sup> h <sup>-1</sup>	1.876 m <sup>3</sup> h <sup>-1</sup>	2.841 m <sup>3</sup> h <sup>-1</sup>	3.206 m <sup>3</sup> h <sup>-1</sup>
*Volume flow (water)	1.092 m <sup>3</sup> h <sup>-1</sup>	1.453 m <sup>3</sup> h <sup>-1</sup>	2.355 m <sup>3</sup> h <sup>-1</sup>	2.265 m <sup>3</sup> h <sup>-1</sup>
Pumping losses	0.04354 kW	0.09472 kW	0.327 kW	0.3761 kW
Exiting temperature (Mobiltherm 594)	175.3 °C	183.9 °C	194.9 °C	198.2 °C
Exiting temperature (water)	46.77 °C	42 °C	34.6 °C	35.64 °C

## 4.4 Evaluation of the WHRS Design

From the results of the simulation there are several conclusions that can be made about the design of the WHRS that should be considered in a future WHRS development especially in respect to the recuperation of large amounts of waste heat. While in commercial plate-heat exchangers the individual channels are connected in parallel in the proposed design the channels are connected in series. This design however leads to considerably smaller volume flow than in a parallel design. In parallel the higher volume flows of the heat carriers allow for higher heat amounts to be transferred due to higher exiting temperatures. Furthermore, in a parallel flow connection, lower pressure drops are expected due to the lower flow deflection at the entrance and the end of the heat exchanger channel. Instead of U-turn the flow only

follows a 90° deflection. This would then also result in a clear flow direction (counter flow or parallel flow) and not in a mixed flow direction (counter flow and parallel flow) as presented in the study. Due to considerably lower pressure drop in the channel flow compared to the bypass flow a system should be built with larger dimensions. Besides these improvements pumps for higher volume flow rates generally work with higher efficiencies. With hindsight to the proposed constructive design, it should be replaced by a design that consider channels connected in parallel. Since the focus was on the methodological approach and the simulation work, these constructive improvements were not further included in this work.

## 5 Conclusion

The study was set out to evaluate a waste heat recovery system based on a new generation of printed and large surface scale TEGs for low to medium temperature levels. In future, there will be an increasing demand for waste heat recovery systems that must work reliably under fluctuating operating conditions. Due to the direct conversion from heat to electricity, TEGs offer explicit advantages compared to classical technologies with a mechanical intermediate stage. This study sought a methodological approach that would allow to give evidence of the potential of such a system from a technological point of view. Therefore, based on a first exemplary concept a thermal simulation model as well as fluid dynamical simulation models were developed. Using this methodology it was accomplished to answer the following research questions:

- “How can a TEG based waste heat recovery system be conceptionally realized?”
- “How can it be simulated?”
- “What maximal output power can be delivered by the WHRS?”

There are many ways to realize a constructive concept. For this study, a simple plate heat exchanger design was adjusted to allow the integration of TEGs between the heating and the cooling channels. Due to the simple and repeatable structure, the simulation work could be conducted on relatively small characteristic sections of the systems and subsequently scaled up to reproduce the behavior of the entire system. By creating a thermal model for the specific design of the WHRS in the Simulink modeling environment the thermal behavior could be simulated. This allowed the monitoring of a variety of absolute and relative parameters related to operation conditions, material parameters and the geometry of the system such as: temperature development along the streaming length as well as through recuperators wall, heat flux through the TEGs, TEG power output, volume flows, fluid velocities, fluid parameters, optimized TEG thicknesses, different universal quantities of heat transfer and fluid dynamics theory ( $Nu$ ,  $Re$ ,  $Pr$ ,  $NTU$ ).

Simultaneously, computational fluid dynamics (CFD) simulations on two characteristic segments of the channels using OpenFOAM were performed to determine the dissipated energy caused by the two fluids moving through the two segments. The influence of the flow velocity and the temperature was included. By implementing the results of the CFD simulation into the Simulink model it was possible to monitor the net efficiencies and most notably the net power output of the system and determine the optimal operation conditions for the mass flows and the optimal TEG thickness.

By analyzing the results produced by the simulation one can generally conclude:

1. That already small conversion efficiencies ( $ZT = 0.1$ ) which are achievable at this moment for the present TEGs can deliver not only the auxiliary energy for powering the pumps but also surplus electrical energy.
2. That the MPPT of the TEG does not necessarily coincides with the MMPP of the entire WHRS.
3. That when determining the technological potential by the net power output depending on the  $ZT$  value, an optimization of the operation conditions in this case ( $NTU_1$ ,  $NTU_2$  and TEG thickness) must be included. Otherwise the potential might be underestimated.
4. That the optimal ratio  $NTU_1/NTU_2$  for maximum net output power was found between relatively narrow bounds of 1.5 and 1.8.
5. That with increasing  $ZT$  values the ideal thickness of the generator tends to decrease.
6. That an increase in  $ZT$  value nearly linearly increases the net electrical power of the WHRS.

In addition to these general conclusions the simulation demonstrated that already a relatively compact WHRS unit like the one proposed in this study and currently available printed OTEGs with a  $ZT$  values of ca. 0.1 can deliver a surplus electrical power of 246.3 W. Due to the linear relationship between  $ZT$  value and net power output one can conclude that a  $ZT$  value of 0.4, which is believed to be realistic achievement in the near future would allow the WHRS to deliver 1 kW of electricity.

However, simulation raises always the question of validation. Over the course of this work it was not possible to compare the simulation with experimental data. Therefore, the delivered results shall be interpreted with caution. Moreover, the simulation work does not qualify for high fluid velocities since the data from the CFD simulation was only conducted to a maximum average inlet velocity of  $0.6 \text{ ms}^{-1}$ . While the methodological approach of this work is transferable on other types of heat exchanger this is not the case for the presented WHRS power values. Both models were explicitly elaborated for the plate heat exchanger design presented in this work. An additional point that is not considered in this study, is the energy needed to make the waste heat available and the energy needed to release the transferred heat into the environment. For a complete life cycle assessment of the WHRS this should be included.



It is obvious that the field of waste heat recovery based on this new generation of printed TEGs still offers much room for improvements. Since the focus of this work lay on the development of the thermal and fluid dynamical models, the technical design of the examined WHRS was kept simple. Commercial heat exchangers usually have a rippled heat exchanger surface to improve the heat transfer mechanism and have a different flow guidance to support larger volume flows and reduce pressure drops due to high flow deflection. Interesting would be how such improvements would change the performance of the WHRS. The elaborated models could be adjusted to simulate these alterations. The characteristic flexibility of the TEGs might be an interesting approach that would allow the TEGs to be incorporated into various heat exchanger designs in which the heat is transferred over curved surfaces. An interesting approach especially from the view of application would be the development of a system that would allow an efficient heat transfer between gaseous and liquid medias.

## References

- [1] K. Bartholomé, J. D. König, H.-F. Pernau, and B. Balke, “Abwärme als Energiequelle,” *Phys. unserer Zeit*, vol. 48, no. 2, pp. 89–95, 2017.
- [2] D. M. Edited by Rowe, *Thermoelectrics Handbook Macro to Nano*. CRC Press, 2005.
- [3] A. Patyk, “Thermoelectrics: Impacts on the environment and sustainability,” *J. Electron. Mater.*, vol. 39, no. 9, pp. 2023–2028, 2010.
- [4] C. B. Vining, “An inconvenient truth about thermoelectrics.,” *Nat. Mater.*, vol. 8, no. 2, pp. 83–85, 2009.
- [5] D. Champier, “Thermoelectric generators: A review of applications,” *Energy Convers. Manag.*, vol. 140, pp. 167–181, 2017.
- [6] K. P. Pipe, G.-H. Kim, L. Shao, and K. Zhang, “Engineered doping of organic semiconductors for enhanced thermoelectric efficiency,” *Nat. Mater.*, vol. 12, no. 8, pp. 719–723, 2013.
- [7] B. Russ, A. Glauddell, J. J. Urban, M. L. Chabinyk, and R. A. Segalman, “Organic thermoelectric materials for energy harvesting and temperature control,” *Nat. Rev. Mater.*, vol. 1, p. 16050, 2016.
- [8] A. Gall, M. Gültig, S. Kettlitz, and U. Lemmer, “Gewickeltes und gefaltetes thermoelektrisches System und Verfahren zu dessen Herstellung,” DE102012105086 A12013.
- [9] A. Rösch, “Design und Simulation druckbarer organischer thermoelektrischer Generatoren,” 2017.
- [10] F. Lessmann, “Identifizierung und Bewertung von Anwendungsfeldern organischer thermoelektrischer Generatoren,” 2014.
- [11] M. T. Børset, Ø. Wilhelmsen, S. Kjelstrup, and O. S. Burheim, “Exploring the potential for waste heat recovery during metal casting with thermoelectric generators: On-site experiments and mathematical modeling,” *Energy*, pp. 1–11, 2016.

- [12] F. Felgner, L. Exel, and G. Frey, "Model-based Design and Validation of Waste Heat Recovery Systems," *Adv. Energy Convers. Symp Model.*, pp. 265–270, 2012.
- [13] AG Heidelberg Zement, "Niedertemperaturverstromung mittels einer ORC-Anlage im Werk Lengfurt der Heidelberger Zement AG," no. August, 2001.
- [14] VDI e.V., *VDI-Wärmeatlas, 11. Auflage*, 11th ed. Springer Vieweg, 2013.
- [15] R. K. Shah, "Laminar flow friction and forced convection heat transfer in ducts of arbitrary geometry," *Int. J. Heat Mass Transf.*, vol. 18, no. 7, pp. 849–862, 1975.
- [16] V. Gnielinski, "Neue Gleichungen für den Wärme- und den Stoffübergang in turbulent durchströmten Rohren und Kanälen," *Forsch. im Ingenieurwes.*, vol. 41, no. 1, pp. 8–16, 1975.
- [17] S. Patankar, *Numerical heat transfer and fluid flow*. 1980.

## List of Tables

Table 1   Turbulence simulation .....	19
Table 2   Parameters of universal quantities .....	21
Table 3   Thickness, heat conductivity and thermal resistance of the stacking order .....	32
Table 4   Thermal resistance due to convection for ambient pressure and stream velocities of $v_1=0.18 \text{ ms}^{-1}$ and $v_2=0.27 \text{ ms}^{-1}$ .....	32
Table 5   Boundary conditions for the characteristic channel flows (wall, outlet and symmetry) .....	46
Table 6   Boundary conditions of the channel flow for the inlet (laminar, turbulent) .....	47
Table 7   Boundary conditions of the bypass flow for the inlet (laminar, turbulent) .....	47
Table 8   Additional OpenFOAM files in the folder system .....	49
Table 9   Results of checkMesh for the bypass flow created with snappyHexMesh .....	59
Table 10   Results for the addLayers step of snappyHexMesh .....	59
Table 11   Comparison numerical and analytical solutions (channel flow) .....	67
Table 12   Specifications of the WHRS I .....	73
Table 13   Specifications of the WHRS II .....	73

## List of Figures

Figure 1   Industrial waste heat potential.....	3
Figure 2   Seebeck effect.....	5
Figure 3   Thermoelectrical parameters $S$ , $\kappa$ and $\sigma$ .....	8
Figure 4   Efficiencies and $ZT$ of current thermoelectric materials .....	9
Figure 5   Classical TEG design .....	10
Figure 6   Thermocouple design.....	10
Figure 7   Printing layout.....	11
Figure 8   Production process of a printed OTEG (Patent holder KIT) .....	12
Figure 9   OTEGs.....	12
Figure 10   Computational fluid dynamics .....	19
Figure 11   Temperature development in heat exchangers .....	22
Figure 12   Case scenario.....	26
Figure 13   Heat exchanger stack.....	27
Figure 14   Stream lines and stacking order of the heat exchanger stack .....	28
Figure 15   Fluid transfer to the next channel .....	29
Figure 16   Reduction of complexity.....	30
Figure 17   Calculation of the total thermal resistance .....	33
Figure 18   Maximum Power Point Tracking.....	34
Figure 19   Temperature development in the WHRS .....	34
Figure 20   Effective temperature difference .....	35
Figure 21   Operation chart of the thermal model .....	37
Figure 22   Characteristic channel flows .....	39
Figure 23   OpenFOAM utility blockMesh.....	40
Figure 24   Simple grading of blockMesh .....	40
Figure 25   File format STL .....	41
Figure 26   Bounding box for snappyHexMesh.....	42
Figure 27   castellatedMesh .....	43
Figure 28   castellatedMesh and snapping.....	43
Figure 29   addLayers.....	44
Figure 30   Exemplary OpenFOAM case folder .....	50
Figure 31   Working principle of the shell script for simulation automation purpose .....	52
Figure 32   Simulation methodology .....	53
Figure 33   Optimum thickness of OTEG .....	54
Figure 34   Temperature development (parallel flow).....	55

Figure 35   Temperature development (counter flow) .....	56
Figure 36   Local temperature development (parallel flow).....	56
Figure 37   Results blockMesh and checkMesh .....	57
Figure 38   Meshing results snappyHexMesh.....	58
Figure 39   Results snappyHexMesh .....	59
Figure 40   mapping.....	60
Figure 41   swak4Foam and groovyBC .....	61
Figure 42   Velocity profiles .....	62
Figure 43   interpolation2DTable k.....	63
Figure 44   interpolation2DTable epsilon.....	63
Figure 45   Pressure drop ( $\Delta p$ ) for laminar flow conditions.....	64
Figure 46   Pressure drop ( $\Delta p$ ) for turbulent flow conditions.....	65
Figure 47   Effective Temperature Difference .....	66
Figure 48   Simulation monitoring of the residuals (channel flow) .....	68
Figure 49   Simulation monitoring (bypass flow) .....	69
Figure 50   Efficiencies of the WHRS ( $ZT \approx 0.1$ ).....	70
Figure 51   Power output of the WHRS ( $ZT \approx 0.1$ ) .....	71
Figure 52   Power output and efficiency of the WHRS ( $ZT \approx 0.5$ ).....	71
Figure 53   Potential of the WHRS for different $ZT$ values .....	72

# Appendix

## Fluid Parameters

Density  $\rho$  of water in  $\text{kg m}^{-3}$  depending on pressure and temperature

Druck $p$ bar	Temperatur $t$ in °C									
	0	25	50	75	100	125	150	200	250	300
1	999,84	997,05	988,05	974,86	958,96	939,16	917,02	885,43	842,10	789,10
5	1000,0	997,23	988,22	975,03	958,54	939,16	917,02	885,43	842,10	789,10
10	1000,3	997,45	988,44	975,25	958,77	939,41	917,30	885,43	842,10	789,10

Specific isobar heat capacity  $c_p$  of water in  $\text{kJ kg}^{-1} \text{K}^{-1}$  depending on pressure and temperature

Druck $p$ bar	Temperatur $t$ in °C									
	0	25	50	75	100	125	150	200	250	300
1	4,219	4,182	4,180	4,192	4,204	4,211	4,216	4,219	4,221	4,222
5	4,217	4,181	4,179	4,191	4,216	4,255	4,310	4,376	4,451	4,534
10	4,215	4,179	4,177	4,190	4,215	4,253	4,309	4,376	4,451	4,534

Kinematic viscosity  $\nu$  of water in  $\text{m}^2 \text{s}^{-1}$  depending on pressure and temperature

Druck $p$ bar	Temperatur $t$ in °C									
	0	25	50	75	100	125	150	200	250	300
1	1,792	0,8927	0,5531	0,3872	0,275	0,200	0,150	0,100	0,060	0,030
5	1,791	0,8924	0,5531	0,3872	0,2939	0,2366	0,1991	0,1626	0,1268	0,0909
10	1,789	0,8922	0,5531	0,3872	0,2939	0,2366	0,1992	0,1627	0,1270	0,0911

Thermal conductivity  $\lambda$  of water in  $\text{W kg}^{-1} \text{m}^{-1}$  depending on pressure and temperature

Druck $p$ bar	Temperatur $t$ in °C									
	0	25	50	75	100	125	150	200	250	300
1	555,7	606,5	640,6	663,6	677,4	682,9	681,0	671,0	651,0	621,0
5	556,0	606,7	640,8	663,8	677,4	682,9	681,0	671,0	651,0	621,0
10	556,3	607,0	641,1	664,1	677,7	683,2	681,4	671,4	651,4	621,4

Prandtl number of water depending on pressure and temperature

Druck $p$ bar	Temperatur $t$ in °C									
	0	25	50	75	100	125	150	200	250	300
1	13,61	6,137	3,566	2,384	1,033	0,9957	0,9771	0,9575	0,9468	0,9389
5	13,59	6,132	3,564	2,383	1,753	1,384	1,156	0,9941	0,9617	0,9466
10	13,56	6,127	3,562	2,383	1,753	1,384	1,156	1,062	0,9870	0,9587

## Fluid parameters of Mobiltherm 594

Stoff (Handelsname)	Temperatur (°C)	Dampfdruck (mbar)	Dichte (kg/m <sup>3</sup> )	spezifische Wärmekapazität (kJ/kg K)	Wärmeleitfähigkeit (W/mK)	dynamische Viskosität (10 <sup>-3</sup> Ns/m <sup>2</sup> )	kinematische Viskosität (10 <sup>-6</sup> m <sup>2</sup> /s)	Temperaturleitfähigkeit (10 <sup>-8</sup> m <sup>2</sup> /s)	Prandtl-Zahl
Mobiltherm 594	-10		892	1,76	0,135	22,7	25,4	8,60	296
	0		886	1,80	0,134	14,1	15,9	8,40	189
	20	0,01	847	1,901	0,137	7	8,63	8,5	102
	40	0,05	834	1,974	0,135	4	4,79	8,2	58
	60	0,19	820	2,048	0,134	2,5	3,03	8,0	38
	80	0,6	807	2,121	0,132	1,69	2,09	7,7	27
	100	1,7	794	2,195	0,131	1,22	1,54	7,5	21
	150	14	761	2,379	0,127	0,66	0,87	7,0	12
	200	72	728	2,563	0,123	0,428	0,59	6,6	9
	250	276	695	2,747	0,120	0,309	0,44	6,3	7

## Exemplary Script snappyHexMesh

```

/*-----* C++ *-----*/
| ===== |
| \ \ / F i e l d | OpenFOAM: The Open Source CFD Toolbox |
| \ \ / O p e r a t i o n | Version: 2.3.0 |
| \ \ / A n d | Web: www.OpenFOAM.org |
| \ \ / M a n i p u l a t i o n |
|-----*/
FoamFile
{
    version      2.0;
    format       ascii;
    class        dictionary;
    object       snappyHexMeshDict;
}
// *****

// Which of the steps to run
castellatedMesh true;
snap true;
addLayers true;

//Optional: single region surfaces get patch names according to
// surface only. Multi-region surfaces get patch name
// surface "_" "region. Default is true
singleRegionName false;

// Geometry. Definition of all surfaces. All surfaces are of class
// searchableSurface.
// Surfaces are used
// - to specify refinement for any mesh cell intersecting it
// - to specify refinement for any mesh cell inside/outside/near
// - to 'snap' the mesh boundary to the surface
geometry
{
    inlet.stl {type triSurfaceMesh; name inletOutlet;}
    outlet.stl {type triSurfaceMesh; name inletOutlet;}
    wallPipe.stl {type triSurfaceMesh; name wallPipe;}
    wallInside.stl {type triSurfaceMesh; name wallSurfaceInner;}
    wallOutside.stl {type triSurfaceMesh; name wallSurfaceOuter;}
    wallSide.stl {type triSurfaceMesh; name wallSurfaceSide;}
    //myRegion.stl {type triSurfaceMesh; name myRegion;}
}

refinementCylinder
{
    type searchableCylinder;
    point1 (0.105 0.28 0.005);
    point2 (0.105 0.28 0.018);
    radius 0.036;
};

// Settings for the castellatedMesh generation.
castellatedMeshControls
{
    // Refinement parameters
    // ~~~~~

```



```

// If local number of cells is >= maxLocalCells on any processor
// switches from refinement followed by balancing
// (current method) to (weighted) balancing before refinement.
maxLocalCells 3000000;

// Overall cell limit (approximately). Refinement will stop immediately
// upon reaching this number so a refinement level might not complete.
// Note that this is the number of cells before removing the part which
// is not 'visible' from the keepPoint. The final number of cells might
// actually be a lot less.
maxGlobalCells 12000000;

// The surface refinement loop might spend lots of iterations refining just a
// few cells. This setting will cause refinement to stop if <= minimumRefine
// are selected for refinement. Note: it will at least do one iteration
// (unless the number of cells to refine is 0)
minRefinementCells 10;

// Number of buffer layers between different levels.
// 1 means normal 2:1 refinement restriction, larger means slower
// refinement.
nCellsBetweenLevels 1;

// Explicit feature edge refinement
// ~~~~~

// Specifies a level for any cell intersected by its edges.
// This is a featureEdgeMesh, read from constant/triSurface for now.

features
(
    {
        file "myRegion.eMesh";
        level 4;
    }
);

// Surface based refinement
// ~~~~~

// Specifies two levels for every surface. The first is the minimum level,
// every cell intersecting a surface gets refined up to the minimum level.
// The second level is the maximum level. Cells that 'see' multiple
// intersections where the intersections make an
// angle > resolveFeatureAngle get refined up to the maximum level.

refinementSurfaces
{
    inletOutlet          { level (1 1); patchInfo {type patch; } }
    wallSurfaceSide      { level (4 4); patchInfo {type wall; } }
    wallPipe             { level (4 4); patchInfo {type wall; } }
    wallSurfaceInner     { level (4 4); patchInfo {type wall; } }
    wallSurfaceOuter     { level (4 4); patchInfo {type wall; } }
}

resolveFeatureAngle 30;

// Region-wise refinement
// ~~~~~

// Specifies refinement level for cells in relation to a surface. One of
// three modes
// - distance. 'levels' specifies per distance to the surface the
//   wanted refinement level. The distances need to be specified in
//   descending order.
// - inside. 'levels' is only one entry and only the level is used. All
//   cells inside the surface get refined up to the level. The surface
//   needs to be closed for this to be possible.
// - outside. Same but cells outside.

refinementRegions
{
    wallPipe              {mode distance; levels ((0.0003 4) (0.0005 3) (0.0007 2) (0.001
1));}
    wallSurfaceSide       {mode distance; levels ((0.00025 4) (0.0005 3) (0.0007 2) (0.001
1));}

```

```
1));} wallSurfaceInner {mode distance; levels ((0.0004 4) (0.0005 3) (0.0007 2) (0.001
1));} wallSurfaceOuter {mode distance; levels ((0.0001 4) (0.0005 3) (0.0007 2) (0.001
1));} refinementCylinder {mode inside; levels ((0.05 4));}

// Mesh selection
// ~~~~~

// After refinement patches get added for all refinementSurfaces and
// all cells intersecting the surfaces get put into these patches. The
// section reachable from the locationInMesh is kept.
// NOTE: This point should never be on a face, always inside a cell, even
// after refinement.

locationInMesh (0.00383228 0.00112273 0.00169262); // Inside point

// Whether any faceZones (as specified in the refinementSurfaces)
// are only on the boundary of corresponding cellZones or also allow
// free-standing zone faces. Not used if there are no faceZones.
allowFreeStandingZoneFaces true;
}

// Settings for the snapping.
snapControls
{
    //- Number of patch smoothing iterations before finding correspondence
    // to surface
    nSmoothPatch 3;

    //- Relative distance for points to be attracted by surface feature point
    // or edge. True distance is this factor times local
    // maximum edge length.
    tolerance 1.0;

    //- Number of mesh displacement relaxation iterations.
    nSolveIter 200;

    //- Maximum number of snapping relaxation iterations. Should stop
    // before upon reaching a correct mesh.
    nRelaxIter 5;

    // Feature snapping

    //- Number of feature edge snapping iterations.
    // Leave out altogether to disable.
    nFeatureSnapIter 6;

    //- Detect (geometric) features by sampling the surface
    implicitFeatureSnap false;

    //- Use castellatedMeshControls::features
    explicitFeatureSnap true;

    //- Detect features between multiple surfaces
    // (only for explicitFeatureSnap, default = false)
    multiRegionFeatureSnap true;
}

// Settings for the layer addition.
addLayersControls
{
    // Are the thickness parameters below relative to the undistorted
    // size of the refined cell outside layer (true) or absolute sizes (false).
    relativeSizes false;

    // Per final patch (so not geometry!) the layer information
    layers
    {
        wallSurfaceInner_wallInside
        {
            nSurfaceLayers 3;
        }
    }
}
```

```
    }

    wallSurfaceOuter_wallOutside
    {
        nSurfaceLayers 3;
    }

    wallPipe_wallPipe
    {
        nSurfaceLayers 3;
    }

    wallSurfaceSide_wallSide
    {
        nSurfaceLayers 3;
    }
}

// Expansion factor for layer mesh
expansionRatio 1.3;

// Wanted thickness of final added cell layer. If multiple layers
// is the thickness of the layer furthest away from the wall.
// See relativeSizes parameter.
finalLayerThickness 0.0001;

// Minimum thickness of cell layer. If for any reason layer
// cannot be above minThickness do not add layer.
// See relativeSizes parameter.
minThickness 0.00001;

// If points get not extruded do nGrow layers of connected faces that are
// also not grown. This helps convergence of the layer addition process
// close to features.
nGrow 0;

// Advanced settings

// When not to extrude surface. 0 is flat surface, 90 is when two faces
// are perpendicular
featureAngle 180;

// Maximum number of snapping relaxation iterations. Should stop
// before upon reaching a correct mesh.
nRelaxIter 5;

// Number of smoothing iterations of surface normals
nSmoothSurfaceNormals 1;

// Number of smoothing iterations of interior mesh movement direction
nSmoothNormals 3;

// Smooth layer thickness over surface patches
nSmoothThickness 10;

// Stop layer growth on highly warped cells
maxFaceThicknessRatio 1.0;

// Reduce layer growth where ratio thickness to medial
// distance is large
maxThicknessToMedialRatio 1.0;

// Angle used to pick up medial axis points
minMedianAxisAngle 90;

// Create buffer region for new layer terminations
```

```
nBufferCellsNoExtrude 0;

// Overall max number of layer addition iterations. The mesher will exit
// if it reaches this number of iterations; possibly with an illegal
// mesh.
nLayerIter 30;

// Max number of iterations after which relaxed meshQuality controls
// get used. Up to nRelaxIter it uses the settings in meshQualityControls,
// after nRelaxIter it uses the values in meshQualityControls::relaxed.
nRelaxedIter 10;
}

// Generic mesh quality settings. At any undoable phase these determine
// where to undo.
meshQualityControls
{
    #include "meshQualityDict" //quality values are defined in the file
    meshQualityDict

    // Optional : some meshing phases allow usage of relaxed rules.
    // See e.g. addLayersControls::nRelaxedIter.
    relaxed
    {
        //- Maximum non-orthogonality allowed. Set to 180 to disable.
        maxNonOrtho 180;
    }

    // Advanced

    //- Number of error distribution iterations
    nSmoothScale 4;
    //- amount to scale back displacement at error points
    errorReduction 0.75;
}

// Advanced

// Write flags
writeFlags
(
    scalarLevels    // write volScalarField with cellLevel for postprocessing
    layerSets        // write cellSets, faceSets of faces in layer
    layerFields      // write volScalarField for layer coverage
);

// Merge tolerance. Is fraction of overall bounding box of initial mesh.
// Note: the write tolerance needs to be higher than this.
mergeTolerance 1E-6;

// ***** //
```

## Shell Script

Command	Description	Example
<b>#!/bin/sh</b>	Tells with which shell to interpret and run the script e.g. <i>bash</i> , <i>ksh</i> , <i>sh</i> , ...	<b>#!/bin/sh</b> <b>#!/bin/bash</b>
<b>echo</b>	A command that outputs a string of an argument that is passed to it. It is generally used to bring a variable into	<b>echo</b> "Hello World!" <b>a=\$(echo "1.2+(\$b*0.7)"   bc)</b>

	scope or output the string to an interface, file or other command.	
<b>&gt;&gt;</b>	Appends passed argument to a file	<code>echo -e "Hello World!" &gt;&gt; file_name</code>
<b>&gt;</b>	Overwrites file with the argument passed to it	<code>echo -e "Hello World!" &gt; file_name</code>
<b>\$</b>	With the \$ in front of a declared variable name the value of the variable is called	<code>echo \$a</code>
<b> </b>	Piping is used to pass the output of one command as input for a following command	<code>a=\$(sed -n "\$j"p file   cut -d "," -f6)</code>
<b>mkdir</b>	Creates a directory (make directory)	<code>mkdir folder_name</code>
<b>cp</b>	Copies files and directories (copy)	<code>cp -r file_name ./path</code>
<b>cd</b>	Defines the current directory (current directory)	<code>cd ./path</code>
<b>sed</b>	Calls a command line based text editor (stream editor) that allows manipulations of files and scripts. In the elaborated scripts, it was often used to find a certain string in a text file and replace it with an argument.	<code>sed -i 's/] .*/] '\$a';/' file_name</code>
<b>awk</b>	Like <i>sed</i> a command line based text editor for analyzing and editing of a file	<code>b=\$(tail -n1 file_name   awk -F " " '{print +\$2}')</code>
<b>tail</b>	Outputs the last line of a file	<code>b=\$(tail -n1 file_name)</code>
<b>touch</b>	The command is often used to create an empty file. But its real purpose is the manipulation of access and modification time	<code>touch file</code>
<b>rm</b>	Deletes files and folder (remove)	<code>rm -rf processor0</code>
<b>sleep</b>	The script will pause for a specified amount of time	<code>sleep 10</code>
<b>bc</b>	A (basic calculator) for arithmetic calculations. Most shells if at all only support simple integer arithmetic to do more complex calculations <i>bc</i> can be used. However, <i>bc</i> does not support scientific number such 2e-06 to still be able use <i>bc</i> for these calculations it is common practice to replace the e with a 10^ using <i>sed</i> or <i>awk</i> .	<code>a=\$(echo "1.2+(\$b*0.7)"   bc)</code>
<b>cut</b>	Cuts portion of a text line	
<b>wc</b>	Counts words, charactes, lines and bytes in text files	<code>i=\$(wc -l file_name   cut -d " " -f1)</code>
<b>man</b>	To call the manual of a command	<code>man bc</code>
<b>./</b>	From this current folder	<code>cd ./path/file_name</code>

### Exemplary script

```
#!/bin/sh
#-----
# ===== |
# \ \      / F i e l d           | OpenFOAM: The Open Source CFD Toolbox
# \ \      / O p e r a t i o n    |
# \ \      / A n d                | Copyright (C) 2011-2014 OpenFOAM Foundation
#  \ \     M a n i p u l a t i o n |
#-----
#
# Script
#
# Description
# This script was created to:
# - read from fluidParameter.csv
# - setup the OpenFOAM folder structure
# - set custom initial and boundary conditions
# - run the solver
# - do several post-processing steps
#-----
echo "Shit is about to get real!!!"
#creates the working directory
mkdir pressureDropChannelBlockMeshSimpleFoamWater
# creates a file "pressDropOverTempAndVelRANS"
touch ./pressureDropChannelBlockMeshSimpleFoamWater/pressDropOverTempAndVelRANS
#adds line to file "pressDropOverTempAndVelRANS"
echo -e "#vel [m/s] \t temp [°C] \t Xi [Pa/m]" >>
./pressureDropChannelBlockMeshSimpleFoamWater/pressDropOverTempAndVelRANS
#copies gnuplot script plotPressureDropOverTempAndVel to the working directory
cp -r ./vorlageSimpleFoamBlockMeshTurbulentFlow/plotPressureDropOverTempAndVel
./pressureDropChannelBlockMeshSimpleFoamWater/
sleep 1s
k=1
l=0
#declares the variable i with the number of lines in fluidParameter.csv
i=$(wc -l fluidParameter.csv|cut -d " " -f1)
#outer while loop changes velocity (make sure there are spaces between the brackets
and your condition)
while [ $i -le $k ]
do
#declares the variable inletVel, to do arithmetic calculations the equation must be
piped to bc
inletVel=$(echo "0.11+($i*0.075)" | bc)
echo "$inletVel"
#creates velocity directory
mkdir ./pressureDropChannelBlockMeshSimpleFoamWater/vel"$inletVel"

j=2
#inner while loop changes temperature
while [ $j -le $l ]
do
#declares global variabel temperature with the value of the 6th line j in
fluidParameter.csv
temp=$(sed -n "$j"p ./fluidParameter.csv|cut -d "," -f6)
#declares global variabel kinematic viscosity with the value of the 8th number of
line j from fluidParameter.csv
kinVis=$(sed -n "$j"p ./fluidParameter.csv|cut -d "," -f8)
#declares global variabel density with the value of the 7th number of line j from
fluidParameter.csv
phi=$(sed -n "$j"p ./fluidParameter.csv|cut -d "," -f7)
echo $temp
echo $kinVis
#creates case directory
mkdir ./pressureDropChannelBlockMeshSimpleFoamWater/vel"$inletVel"/"$temp"
#create OpenFOAM case structure from template folder
cp -r ./vorlageSimpleFoamBlockMeshTurbulentFlow/0
./pressureDropChannelBlockMeshSimpleFoamWater/vel"$inletVel"/"$temp"/
cp -r ./vorlageSimpleFoamBlockMeshTurbulentFlow/constant
./pressureDropChannelBlockMeshSimpleFoamWater/vel"$inletVel"/"$temp"/
cp -r ./vorlageSimpleFoamBlockMeshTurbulentFlow/system
```

```

./pressureDropChannelBlockMeshSimpleFoamWater/vel"$inletVel"/"$temp"/
cp -r ./vorlageSimpleFoamBlockMeshTurbulentFlow/Allclean
./pressureDropChannelBlockMeshSimpleFoamWater/vel"$inletVel"/"$temp"/
cp -r ./vorlageSimpleFoamBlockMeshTurbulentFlow/runParallel
./pressureDropChannelBlockMeshSimpleFoamWater/vel"$inletVel"/"$temp"/
cp -r ./vorlageSimpleFoamBlockMeshTurbulentFlow/runSingleCore
./pressureDropChannelBlockMeshSimpleFoamWater/vel"$inletVel"/"$temp"/
cp -r ./vorlageSimpleFoamBlockMeshTurbulentFlow/residual_plots
./pressureDropChannelBlockMeshSimpleFoamWater/vel"$inletVel"/"$temp"/
cp -r ./vorlageSimpleFoamBlockMeshTurbulentFlow/fitVelocityProfile
./pressureDropChannelBlockMeshSimpleFoamWater/vel"$inletVel"/"$temp"/
cp -r ./vorlageSimpleFoamBlockMeshTurbulentFlow/fitVelocityProfileWallSide
./pressureDropChannelBlockMeshSimpleFoamWater/vel"$inletVel"/"$temp"/
cp -r ./vorlageSimpleFoamBlockMeshTurbulentFlow/maxVelFromPostProcessing
./pressureDropChannelBlockMeshSimpleFoamWater/vel"$inletVel"/"$temp"/
cp -r ./vorlageSimpleFoamBlockMeshTurbulentFlow/wallSideSetUpVelocityProfile
./pressureDropChannelBlockMeshSimpleFoamWater/vel"$inletVel"/"$temp"/
cp -r ./vorlageSimpleFoamBlockMeshTurbulentFlow/splitCropAndSortEpsilon
./pressureDropChannelBlockMeshSimpleFoamWater/vel"$inletVel"/"$temp"/
cp -r ./vorlageSimpleFoamBlockMeshTurbulentFlow/splitCropAndSortK
./pressureDropChannelBlockMeshSimpleFoamWater/vel"$inletVel"/"$temp"/
#move into the constant archive of the current case
cd ./pressureDropChannelBlockMeshSimpleFoamWater/vel"$inletVel"/"$temp"/constant/
#the value of the kinetic viscosity is changed according to the kinetic viscosity
from the fluidParameter.csv the -i tells sed to overwrite the data
sed -i 's/[] .*/[] '$SkinVis';/' transportProperties
#move up
cd ..
#move into folder 0
cd ./0/
#file U in folder 0 is manipulated with new velocity
sed -i 's/inlet{type mapped; value uniform ( 0.2662 0 0 ); setAverage true; average
( 0.2662 0 0 );}/inlet{type mapped; value uniform ( '$inletVel' 0 0 ); setAverage
true; average ( '$inletVel' 0 0 );}/' U
cd ..
#distributes the problem on the different cores
decomposePar
#runs the solver simpleFoam in parallel on the four cores and writes the output in
a log-file
mpirun -np 4 simpleFoam -parallel > log.simpleFoam
#puts the four single solution of the problem back together
reconstructPar
#unnecessary folders are removed
rm -rf processor0
rm -rf processor1
rm -rf processor2
rm -rf processor3
# run postProcess utility to determine flow velocity profiles
postProcess -func sampleDict
#declares variable of the relative pressure difference between inlet and outlet
deltaPphi=$(tail -n1 ./postProcessing/pressureDifferencePatch/0/fieldValueDelta.dat
| awk -F " " '{print +$2}')
echo "$deltaPphi $phi"
#declares variable of the pressure drop per m for the channel flow
deltaP=$(echo "$deltaPphi*$phi*10" | bc)
echo "$deltaP"
sleep 10
#additional script that fits the flow velocity profiles with polynomial functions
bash wallSideSetUpVelocityProfile
#additional script that creates lookup table for the turbulent variable epsilon
bash splitCropAndSortEpsilon
#additional script that creates lookup table for the turbulent variable k
bash splitCropAndSortK
cd ..
cd ..
#writes velocity, temperature and pressure drop to the file
echo -e "$inletVel \t $temp \t $deltaP" >> pressDropOverTempAndVelRANS
j=`expr $j + 1` #inner loop +1
done

```

```
l=expr $1 + 1 #outer loop +1
done
#clear variables and start script for bypass flow simulation
env -i bash runSimpleFoamWithChangingKinVisAndVelWaterTurbulentFlowKanalbiegung

echo "Shit got real, really bad!!!"

# ----- end-of-file
```

## Hardware

```
*-cpu
  description: CPU
  product: Intel(R) Core(TM) i5-7500 CPU @ 3.40GHz
  vendor: Intel Corp.
  physical id: 1a
  bus info: cpu@0
  version: Intel(R) Core(TM) i5-7500 CPU @ 3.40GHz
  serial: To Be Filled By O.E.M.
  slot: CPU1
  size: 900MHz
  capacity: 4005MHz
  width: 64 bits
  clock: 100MHz
  configuration: cores=4 enabledcores=4 threads=4

*-memory
  description: System Memory
  physical id: 11
  slot: System board or motherboard
  size: 16GiB
  *-bank:0
    description: [empty]
    physical id: 0
    slot: DIMM CHA3
  *-bank:1
    description: DIMM Synchronous 2400 MHz (0.4 ns)
    product: M378A2K43BB1-CRC
    vendor: Samsung
    physical id: 1
    serial: 34D02BC8
    slot: DIMM CHA1
    size: 16GiB
    width: 64 bits
    clock: 2400MHz (0.4ns)
  *-bank:2
    description: [empty]
    physical id: 2
    slot: DIMM CHB4
  *-bank:3
    description: [empty]
    physical id: 3
    slot: DIMM CHB2
```