



Deutsches Zentrum
für Luft- und Raumfahrt



Untersuchung des Einflusses der Filletradien auf die aerodynamische Performanz und die Fertigungskosten bei radialen Turbomaschinen für Hochtemperaturwärmepumpen

Bachelorarbeit

für die Prüfung zum
Bachelor of Science

des Studienganges
Informationstechnik

an der Dualen Hochschule Baden-Württemberg Mannheim

von

Andreas Rafael Hoffmann

27. August 2024

Bearbeitungszeitraum	4.06.2024-27.08.2024
Matrikelnummer, Kurs	6812048, TINF21IT1
Ausbildungsfirma	DLR (Deutsches Zentrum für Luft und Raumfahrt e.V.), Cottbus
Betreuer der Ausbildungsfirma	Michael Lockan
Gutachter der Dualen Hochschule	Jürgen Schultheis

Eidesstattliche Erklärung

Ich versichere hiermit, dass ich meine Projektarbeit mit dem Thema: *Untersuchung des Einflusses der Filletradien auf die aerodynamische Performance und die Fertigungskosten bei radialen Turbomaschinen für Hochtemperaturwärmepumpen* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Ort Cottbus

Datum 27.08.2024 Unterschrift Hoffmann

Abstract

Um den fortschreitenden Klimawandel aufhalten zu können, müssen neue Technologien erforscht und bestehende verbessert werden. Ein großer Teil der CO₂-Abgase entsteht in der Industrie. Einige Anlagen, welche Wärme benötigen und dafür aktuell auf Verbrennung setzen, könnten mithilfe von Hochtemperaturwärmepumpen klimafreundlicher oder sogar neutral gemacht werden. Um der Industrie die für die jeweilige Anwendung optimale Anlage aufzeigen zu können, müssen die Bestandteile optimiert werden. Ein Bestandteil ist der in der Wärmepumpe verbaute Verdichter. Den gesamten Verdichter zu optimieren würde den Rahmen dieser Arbeit übersteigen. Die Frage ist deshalb, welcher Wirkungsgrad für eine bestimmte Zahl an Betriebsstunden ideal ist, wobei die Anschaffungskosten mit dem Wirkungsgrad steigen. Dafür wird in dieser Arbeit die Verdichtereffizienz in Abhängigkeit vom Filletradius, einem Parameter bei der Verdichterauslegung, betrachtet. Dafür werden die Grundlagen von Clausius-Rankine-HTWPs, Verdichtern und des Fräsens als Fertigungsmethode dargestellt. Für die Umsetzung wurde ein Pythonprogramm geschrieben, das über Fidelity aufgerufen wird. Dadurch wird ein Netz erzeugt, welches in eine bestehende Prozesskette eingesetzt wird, um FEM-Analysen zu ermöglichen. Das neue Programm muss in die bestehende Kette eingefügt werden. Auf der anderen Seite wurde die Fertigung betrachtet. Dafür wurde ein Kostenmodell aufgestellt, das die reine Fertigungszeit unter Idealbedingungen ohne Werkzeugabnutzung berechnet. Als beides ins Verhältnis gesetzt wurde, kam heraus, dass die Hauptkosten der Fertigung nicht in der reinen idealen Bearbeitungszeit lagen, wie zuerst angenommen. Außerdem konnte das geschriebene Programm zum aktuellen Zeitpunkt nicht in die Prozesskette integriert werden.

To stop the progress of the climate change new technologies need to be researched and existing ones improved. A big part of the CO₂-emissions originates in the industrial sector. Factories that require heat to function, that currently rely on combustion, could, with the help of high temperature heatpumps, be made more climate friendly or even neutral. To present optimal plant setups to the industry, the parts need to be optimized. One component build into heatpumps is the compressor. To optimize the entire compressor would be to extensive for this work. The question is which efficiency is ideal for a set amount of operating hours. The acquisition cost increases with a higher efficiency. For that the topic of this thesis is the efficiency of compressors viewing only the influence of the fillet radii, which is a parameter for compressors. For this the basics of clausius-rankine-

high-temperature-heatpumps, compressors and of milling techniques will be covered. For the realisation a python program will be written, which will be executed in fidelity. The programm will create a mesh, that will be used in an existing process chain, to enable Finite-Elemente-Methode (FEM)-analyses. For that the new programm must be made compatible with the chain. The second topic of the thesis was the fabrication process. For it a cost-model was created, that computes the pure fabrication time under ideal conditions without work wear. The result of the combination of the two parts was, that the main cost factor for the fabrication is not in the ideal worktime as assumed previously. In the end the written programm could not yet be integrated into the process chain.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Problemraumbeschreibung	2
1.2	Strukturierte Aufgabenstellung	4
1.3	Methoden und Verfahren	5
2	Auslegung des Verdichters und Verdichterarten	6
2.1	Technische Grundlagen	6
2.1.1	Der inverse Clausius-Rankine Zyklus	6
2.1.2	Der Aufbau eines Verdichters	8
2.1.3	Stand der Technik	10
2.2	Umsetzung der Verdichteroptimierung	10
2.2.1	Kriterien und Zielbedingungen	10
2.2.2	Automatisierte Prozesskette zur Optimierung	11
2.2.3	Geometrieerstellung mittels Fidelity	12
2.2.4	Vernetzung mit Fidelity	19
2.2.5	Export des Netzes mittels des OpenFoam Dateiformats	24
2.2.6	Schreiben der Calculix-Inputdateien	27
3	Kosten bei der Fertigung und im Betrieb	31
3.1	Der Einfluss des Wirkungsgrades auf die Betriebskosten	31
3.2	Kosten beim Fräsvorgang	33
3.2.1	Grundlagen des Fräsens und der Modellerstellung mit Einschränkungen	33
3.2.2	Stand der Technik	36
3.2.3	Umsetzung des Kostenmodells im Programm	36
3.3	Betrachtung der Kosten an einem Beispielprozess	40
4	Optimierung auf einen Kompromiss aus Betriebs- und Fertigungskosten	45
4.1	Grundlagen der Optimierung	45
4.1.1	Stand der Technik	46
4.2	Optimierung des Rotors anhand der Filletradien	46
4.3	Betrachtung der Wirtschaftlichkeit	51

Inhaltsverzeichnis

5 Ergebnis	53
6 Zusammenfassung und Ausblick	54
Literaturverzeichnis	IX

Abkürzungsverzeichnis

API	Programmierschnittstelle
CAM	Computer-aided-manufacturing
CSM	Computational structural mechanics
CFD	computational fluid dynamics
DLR	Deutsches Zentrum für Luft- und Raumfahrt
FEM	Finite-Elemente-Methode
GUI	Grafische Nutzeroberfläche
GTA	Geometric Task Arena
HTWP	Hochtemperatur-Wärmepumpe

Abbildungsverzeichnis

2.1	Schaltbild einer einfachen Rankine Wärmepumpe aus [8]	7
2.2	T-S Diagramm des Rankine Zyklus [10]	8
2.3	Fehlgeschlagene Generierung des Blattmantels bei ganzen Kurven	16
2.4	Scheibe mit einer Splitt(rechts)- und einer Hauptschaufel(links) als ein Körper in Fidelity	20
2.5	Fehlerhafte Volume to surface Netze, rechts mit doppelt so hoher Netzauflösung	21
2.6	Vernetzte Geometrien, links mit 0.5mm Zellgröße, links mit 1mm Zellgröße und Kurvenverfeinerung	22
2.7	Vernetzte Geometrien, links mit 0.5mm Zellgröße, links mit 1mm Zellgröße und Kurvenverfeinerung	22
2.8	Finales Netz mit Zoomausschnitt auf das Fillet. Rechts geöffnet, um die Netzelemente zu zeigen	24
2.9	Darstellung der Fehlerhaften Netzelemente in PrePoMax	30
3.1	Darstellung des Kostenverlaufs nach Wirkungsgrad für eine Beispielwärmepumpe aus [19]	32
3.2	Darstellung der hier wichtigen Parameter eines Fräskopfs. Der Fräskopf selbst stammt aus [23]. Es handelt sich um die Darstellung des Kopfes H7490140	35
3.3	Darstellungen der Schruppreihen. Links mit einem Filletradius von zwei Millimetern (blau), rechts mit 0.5 Millimeter Radius	39
3.4	Zoom auf den Filletradius, bei dem die Zwischenreihen als rote Kästen eingezeichnet wurden.	40
3.5	Der Kostenverlauf der vier betrachteten Bahnarten einzeln mit höherer Abtastung (0.01 mm). I. Die Schruppbahnen II. Das Bufferlayer III. Schlichtbahnen im Radius IV. Restschichtbahnen	41
3.6	Kostenverlauf in Abhängigkeit vom Filletradius bei einem angenommenen quadratischen Materialstart von 3mm Seitenlänge. Programmwert in blau, Wert nach Formel berechnet in orange	43
4.1	Darstellung der isentropen Wirkungsgrade der Geometrie über die verschiedenen Abläufe der Optimierung	47

Abbildungsverzeichnis

4.2	Darstellung der isentropen Wirkungsgrade der Geometrie über die verschiedenen Abläufe der Optimierung. Gezeigt werden nur Werte mit eingehaltenen Nebenbedingungen	48
4.3	Darstellung der Spannung der Geometrie über die verschiedenen Abläufe der Optimierung	48
4.4	Darstellung des Massenstroms der Geometrie über die verschiedenen Abläufe der Optimierung	49
4.5	Darstellung des empfohlenen Radius für eine bestimmte Zahl an Betriebsstunden	51

1 Einleitung

Die Klimaerwärmung schreitet Jahr für Jahr, getrieben unter anderem durch die vom Menschen produzierten Abgase, weiter voran. Je nachdem wie die Menschheit in den nächsten Jahren mit den Abgasen und ihrem Energieverbrauch umgehen wird, wird 2045 ein Punkt überschritten sein, an dem sich die Erde nicht mehr erholen kann. Einen nicht zu vernachlässigenden Teil der CO₂ Emissionen von in etwa 17% trägt die Industrie zu den Emissionen bei. Das sind mehr als sechs Milliarden Tonnen pro Jahr. Das entspricht Stand 2022 ca. dem Ausstoß Indiens, Japans und der EU zusammen.[1] Um den Klimawandel aufzuhalten, müssen laut dem Weltklimarat die Treibhausgasemissionen im Vergleich zu denen des Jahres 2019 bis 2030 um 43% gesenkt werden.[2] Obwohl sich diese Zahl nicht ausschließlich aus Kohlenstoffdioxid zusammensetzt, macht es dennoch zumindest in der EU rund 80% der Treibhausgase aus.[3] Eine Minderung der Abgase kann zum Einen durch Sparmaßnahmen erreicht werden, zum Anderen durch technische Neuerungen. Diese Arbeit befasst sich mit letzteren, spezifischer der Technik der Hochtemperatur-Wärmepumpe (HTWP). Das Konzept und die Technik an sich ist bereits seit langer Zeit bekannt. In kleinerem Maßstab und niedrigeren Temperaturen wird es auch im Alltag zum Beispiel bei Heizungen oder Kühlschränken verwendet. Unabhängig vom Anwendungsbereich ist das Funktionsprinzip, dass einem Raum Wärmeenergie entnommen wird, wodurch der Raum gekühlt wird, um die Temperatur in einem Medium durch thermodynamische Vorgänge zu erhöhen, wodurch sie an ein anderes Medium abgegeben werden kann.

Eigentlich interessant sind die Wärmepumpen in hohen Temperaturbereichen. Ab welchem Temperaturbereich eine Wärmepumpe als Hochtemperaturwärmepumpe bezeichnet wird, ist Definitionssache. Temperaturen von über 100°C sind jedoch geläufig[4]. Um Zieltemperaturen in dieser Größenordnung zu erzeugen, gibt es verschieden angelegte Kreisprozesse, welche das Prinzip umsetzen. Diese Arbeit wird sich dabei auf den Clausius-Rankine Zyklus fokussieren. Daneben gibt es zum Beispiel noch den Brayton Zyklus.

Neben den Zyklen werden HTWP danach unterschieden, welches Medium sie verwenden und wie viele Stufen sie haben.

Die Rankine-Forschungspumpe, die am Standort Zittau entwickelt wird, verwendet als Medium Wasser beziehungsweise Dampf. Das Medium wird beim Rankine Prozess in einem Kreislauf gehalten und dabei durch die Umgebungstemperatur erwärmt. Im Anschluss wird der entstandene Dampf mit einem Verdichter komprimiert, um höhere Temperaturen zu erreichen. Für diese Komprimierung wird Energie verbraucht. Dieser Schritt be-

einflusst maßgeblich die Effizienz des Systems. Eine starke Druckerhöhung bedeutet eine höhere Leistungsfähigkeit der Wärmepumpe und braucht mehr Energie. Um einen kontinuierlichen Massefluss zu ermöglichen, werden Turboverdichter verwendet, welche den Fluss mittels Rotoren verdichten. Bei den Turbomaschinen wird in diesem Fall ein Radialverdichter genutzt. Sie zeichnen sich, verglichen mit axialen Verdichtern, typischerweise durch höhere Druckverhältnisse pro Stufe aus[11]. Alternativ könnte ein mehrstufiger Axialverdichter in der Wärmepumpe verwendet werden, um denselben Verdichtungsgrad zu erreichen. Dieser würde jedoch mehr Platz brauchen. Bei dem betrachteten Anwendungsbereich bezüglich Drehzahl und Durchmesser ist der Wirkungsgrad eines Radialverdichters außerdem höher[6]. Dabei handelt es sich um den optimal erreichbaren Wirkungsgrad bei den gegebenen Betriebsbedingungen. Dieses Optimum ist baulich mit aktuellen Maßnahmen nicht zu erreichen. Eine Näherung an den Wirkungsgrad bei gleichbleibenden Druckverhältnissen ist für die Wirtschaftlichkeit späterer Anlagen erstrebenswert. Deshalb sollen die Rotorblätter, welche Kernbestandteil der Verdichter sind, optimiert werden. Neuanfertigungen zu jeder Änderung, um die Auswirkung zu betrachten, würde zu viel Geld und Zeit in Anspruch nehmen. Deshalb sollen die Änderungen simuliert werden. Neben dem Wirkungsgrad, welcher die Fluidmechanik betrachtet, muss auch die Haltbarkeit des Verdichters beachtet werden. Zu dünne Stellen könnten zu Brüchen und damit zur Zerstörung der Komponente führen.

Über Zeit sind Abnutzungserscheinungen nicht zu vermeiden. Entsprechend ergeben sich zwei Optimierungsparameter: Die erwartete Haltbarkeit und der Wirkungsgrad. Aus ihnen erfolgt, wirtschaftlich gesehen übertragen auf einen Anwendungsfall, die Optimierung auf einen möglichst geringen Preis bei gegebenen Betriebsstunden. Es wird dabei erwartet, dass die Kosten, je höher der Wirkungsgrad ist, steigen. In dieser Arbeit soll der optimale Filletradius für einen bestimmten Satz an Betriebsstunden durch eine Optimierung des Verdichters gefunden werden. Das Ergebnis wird dann mit einem Kostenmodell verglichen. Das Ziel dabei ist, dass die Summe aus Anschaffungskosten und Betriebskosten möglichst klein wird. Optimiert wird dabei auf einen optimalen Wirkungsgrad. Das Kostenmodell wird damit dann für eine bestimmte Menge an Betriebsstunden verglichen. Für die Fertigungskosten werden gefräste Verdichter betrachtet.

1.1 Problemraumbeschreibung

Für die Abstimmung von Wirkungsgrad und Anschaffungskosten des Verdichters muss für beides eine annähernd genaue Simulation oder Studie erstellt werden. Als wichtiger Parameter für beide wurde der Filletradius identifiziert. Das Fillet ist ein Bestandteil des Verdichterrisors. Es findet sich am Übergang zwischen der Scheibe und den Schaufeln. Den Ansatz letzterer auf der Scheibe umläuft das Fillet. Für den Wirkungsgrad liegt bereits eine Prozesskette für die Optimierung vor. Bislang wurde das Netz dafür mit Gmsh

1 Einleitung

erzeugt. Stattdessen soll jetzt das Programm Fidelity verwendet werden.

Das heißt, dass die Geometrie, welche vorher durch die Programme Geometric Task Arena und Bladegen definiert wurde, in das Programm geladen werden muss. Dabei muss darauf geachtet werden, dass die Geometrie mit ausreichender Genauigkeit übernommen wird. Da Fidelity die vorliegende Geometrie nicht direkt einlesen kann, muss dafür ein Programm geschrieben werden. Das vorliegende Gmsh Programm kann dabei als Leitfaden dienen, obwohl viele Operationen sich unterscheiden oder im neuen Programm nicht empfohlen sind. Wie schon im Gmsh Programm genügt es, je ein Exemplar jeder Flügelart mit dem dazugehörigen verbindenden Teil der Scheibe zu modellieren. Am Ende soll die Geometrie im Programm als Körper erkannt werden. Das heißt, dass die verschiedenen Schaufelarten ohne Lücken und Hohlräume, aber auch ohne Überschneidung zwischen den Schaufeln, mit der Scheibe verbunden sind. Wenn das vorhandene Segment entsprechend der Anzahl der Klingen dupliziert und um die Zentralachse rotiert wird, kann der Rotor falls nötig vervollständigt werden.

Im Anschluss muss das Werkstück vernetzt werden, damit Computational structural mechanics (CSM)-Simulationen darauf laufen können. Die Vernetzung kann ebenfalls in Fidelity durchgeführt werden. Dabei müssen die richtigen Optionen für den Vernetzer gewählt werden. Die Zahl der Elemente, aus denen das erzeugte Netz besteht, soll dabei idealerweise zwischen 100.000 und 150.000 liegen. Außerdem sollen die Schaufeln überall mit mindestens zwei Elementen in der Breite und die Scheibe mit dreien vernetzt sein.

Im Anschluss muss das Netz in einer für den Rest der Prozesskette verwertbaren Form, das heißt in erster Linie für den nächsten Schritt mit dem Programm Calculix, exportiert werden. In der Prozesskette wird die Netzgeometrie mit Fokus auf die Elemente an den äußeren Flächen eingelesen. Diese Module müssen vollautomatisch in die Prozesskette eingebunden werden. Dafür bieten sich Systemaufrufe an. Die eigentliche Optimierung läuft dann auf einem Cluster ab, da dort viele Simulationen gleichzeitig laufen können. Simuliert wird mit dem Autoopti Tool. Dieses muss entsprechend mit dem Fidelity-Code gestartet werden können. Das Programm muss die Inputdateien erfassen und lesen können. Für den zweiten Teil müssen Informationen über den Herstellungsprozess der Verdichterro-toren gesammelt werden. Der Fokus dabei liegt auf der Fertigung der Fillets, da vorerst nur der Wirkungsgrad und die Kosten in Abhängigkeit von diesem Parameter betrachtet werden. Um den Rahmen nicht zu sprengen und für eine übersichtlichere Darstellung zu sorgen wird nur das Fräsen als Fertigungsmittel betrachtet. Damit soll zuerst die erwartete Fertigungszeit und dann die erwarteten Kosten für eine Anzahl verschiedener Filletradien berechnet werden. Daraus ergeben sich Diagramme, welche die Fertigungskosten mit den Betriebskosten in Abhängigkeit vom Wirkungsgrad vergleichbar machen. Im letzten Schritt werden die Ergebnisse zusammengefügt. Als Ergebnis soll der optimale Filletradius in Abhängigkeit von der erwarteten Betriebszeit berechnet werden.

1.2 Strukturierte Aufgabenstellung

1. Geometrieerstellung in Fidelity
 - 1.1 Fidelity wird automatisch über Konsolenaufruf o.ä. gestartet
 - 1.2 Das ab Kapitel 2.2.3 beschriebene Script wird in Fidelity ausgeführt
 - 1.3 Die Geometrie der Schaufelscheibe wird über das Pythonscript nach Fidelity importiert
 - 1.4 Die Geometrie der Hauptschaufel wird über das Pythonscript nach Fidelity importiert
 - 1.5 Die Geometrie der Leitschaufeln wird über das Pythonscript nach Fidelity importiert und mit der Scheibe verbunden
2. Vernetzung über Fidelity
 - 2.1 Die Geometrie wird über das geschriebene Script automatisch vernetzt
 - 2.2 Die Netzkonvergenz für die CSM wird erreicht
 - 2.3 Das Netz wird für Calculix lesbar exportiert
3. Optimierung
 - 3.1 Der Fidelity Scriptaufruf ist in Autoopti integriert
 - 3.2 Die Optimierung kann gestartet werden und wirft keine programmbedingten Fehler
4. Kostenmodell
 - 4.1 Die Anzahl der Fräsbahnen kann näherungsweise berechnet werden
 - 4.2 Das Kostenmodell in Abhängigkeit von der Filletradiengröße ist aufgestellt und grafisch dargestellt
 - 4.3 Für das Kostenmodell wurde eine Näherungsgleichung gefunden
5. Der Einfluss von Wirkungsgrad und Kostenmodell ist dargestellt und erklärt

1.3 Methoden und Verfahren

Programm	Zweck	Version
AutoOpti ¹	Automatisierter Optimierer	Date 04-02-2020 Revision 6021
Bladegen ¹	Erstellung der Punkte der Geometrie	1.4.1
Calculix	CFD und CSM	2.18
Fidelity	Geometrie und Simulationsprogramm	2023.2
Gmc ²	Teil der Prozesskette	9.4.5
GMSH	Teil der Prozesskette	4.12.2
GTA ¹	Teil der Prozesskette	1.
POST ²	Teil der Prozesskette	9.5.1
PrePoMax	Darstellung von Meshgeometrien	v.2.1.6 dev
PREP ²	Teil der Prozesskette	9.5.1
PyMesh ¹	Teil der Prozesskette	v1.01.027
Python	Programmiersprache	3.8
TRACE ²	Teil der Prozesskette	9.5.1
fidelity	Library	Pyton 3.8, Fidelity 2023.2
matplotlib	Library	3.5.3
numpy	Library	1.21.6
scipy	Library	1.5.2
Tespy	Simulationsprogramm für Berechnung der Kosten pro Wirkungsgrad	0.6.3

Tabelle 1.1: Übersicht über die verwendeten Programme und Python Bibliotheken.
(¹ Software des DLR, ² gemeinsame Software DLR und MTU)

Die in Kapitel 2.2.2 beschriebene Prozesskette, das grundlegende Gmsh-Programm, auf dem das Fidelity Programm aufbaut, und das Tespy-Pythonprogramm aus Quelle [19] sind beim DLR-Institut für CO₂-arme Industrieprozesse gespeichert und wurden dort entwickelt. Mit Prozesskette ist im Kontext dieser Tabelle immer die in Kapitel 2.2.2 beschriebene Prozesskette gemeint.

2 Auslegung des Verdichters und Verdichterarten

In diesem Kapitel werden die grundlegenden Prozesse, welche als Vorarbeit für die Auslegung des Verdichters nach dem Filletradius wichtig sind, vorgestellt. Dazu gehört ein Grundwissen über den Anwendungsprozess HTWP, um die Aufgabe des Verdichters, der optimiert werden soll, zu verstehen. Für das Verständnis der Begriffe und Ziele ist auch der Aufbau von Verdichtern an sich wichtig. Die Grundlagen der Optimierung folgen in Kapitel 4.1. Die Schritte der Prozesskette werden nach den technischen Grundlagen zusammen mit den Zielkriterien vorgestellt. Letztlich geht es zu den tatsächlichen Änderungen in besagter Prozesskette, welche einer der Hauptpunkte der Arbeit sind. Hier geht es um das Einlesen der Geometrie in das Programm Fidelity und die anschließende Vernetzung. Um den Anschluss an die Prozesskette wieder herzustellen ist auch der Export des gebildeten Netzes Thema.

2.1 Technische Grundlagen

2.1.1 Der inverse Clausius-Rankine Zyklus

Der Clausius-Rankine Zyklus, auch oft nur Rankine-Zyklus genannt, ist wie zum Beispiel auch der Joule- und der Ericsson Prozess, einer der thermodynamischen Kreisprozesse, bei denen in erster Linie das erste Gesetz der Thermodynamik ausgenutzt wird, um Wärmeenergie in elektrische Energie umzuwandeln.

Das erste thermodynamische Gesetz besagt dabei, dass die Summe der mechanischen, chemischen, elektrischen und thermischen Energien sowie der Kernenergie in einem geschlossenen System gleich bleibt[7]. Entsprechend ist es möglich Energie im System in eine andere Energieform umzusetzen. Beim normalen rechtsläufigen Prozess von Kraftmaschinen ist das eine Umwandlung von thermischer Energie in letztendlich elektrischen Strom das Ziel. Beim hier betrachteten linksläufigen Prozess einer Wärmepumpe ist dagegen die thermische Energie das Ziel. Der grundlegende Rankine-Prozess besteht aus einem Kompressor, zwei Wärmetauschern und einem Ventil. Die Komponenten sind entsprechend Abbildung 2.1 angeordnet. Rechts und links sind die beiden Wärmetauscher, entsprechend ihrer spezifischeren Funktion Verdampfer, bzw. Evaporator, und Kondensa-

2 Auslegung des Verdichters und Verdichterarten

tor, Condenser, genannt. Die Tauscher sind durch den Verdichter und das Ausbreitungsventil getrennt. Um die Abläufe besser darzustellen, sind sie in 2.2 als Diagramm über

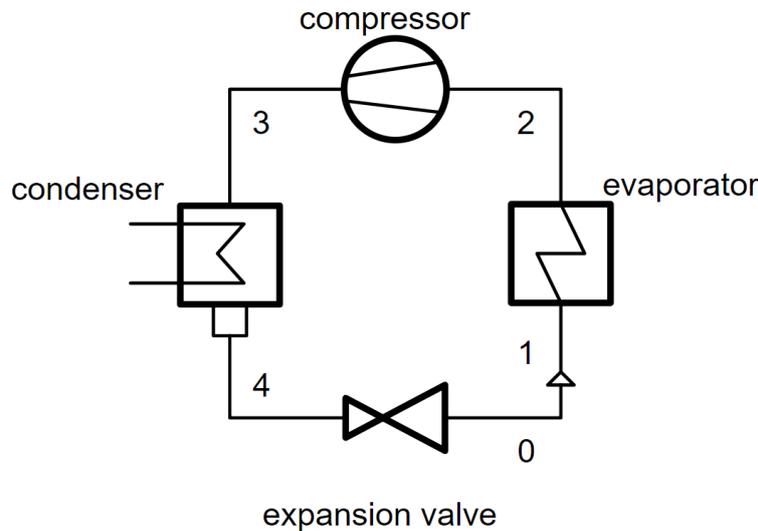


Abbildung 2.1: Schaltbild einer einfachen Rankine Wärmepumpe aus [8]

die spezifische Entropie und Temperatur dargestellt. Die Entropie ist dabei das Maß der Unordnung. Sie ist ein Maß dafür, wie viele Zustände ein betrachtetes System auf Molekülebene annehmen kann[9]. Die Entropie gasförmigen Wassers ist zum Beispiel höher als die von Eis, da in Eis die Moleküle klarer geordnet sind.

Im Diagramm ist der Verlauf des Zyklus rot dargestellt. Vier Eckpunkte sind durch Zahlen markiert. Zur Veranschaulichung wird beim ersten Punkt begonnen. Das Medium ist hier vollständig gasförmig und hat wie aus der Grafik hervorgeht eine verhältnismäßig niedrige Temperatur. Das Gas erreicht dann den Kompressor und wird verdichtet, wobei die Energie pro Volumen steigt und damit die Temperatur. Die Entropie bleibt dabei idealerweise unverändert. Im nächsten Schritt kommt der übererhitzte Dampf in den Kondensator, wo die Temperatur abfällt. Teile des Mediums werden wieder flüssig. Es handelt sich um Nassdampf. Der Druck ist dabei konstant. Thermische Energie wird über den Kondensator nach außen abgegeben. Die Entropie sinkt hier. Das Medium wird weniger chaotisch. Dieser Übergang findet zwischen Punkt zwei und drei statt. Zwischen drei und vier wird das Medium durch das Drosselventil geleitet. Dabei gibt es einen Druckabfall hinter dem Ventil und die Temperatur fällt damit ab. Die Entropie steigt. Im nächsten und letzten Schritt welcher den Kreis schließt wird dem System über den Verdampfer wieder Energie hinzugefügt. Die Temperatur bleibt dabei dennoch gleich. Sie fließt in den Übergang vom Nassdampf- zum übererhitzen Zustand. Nach diesem Schritt ist der Prozess wieder an Ausgangspunkt Eins angelangt. Die Verringerung der Temperatur zwischendurch ist nötig, damit dem System leichter thermische Energie hinzugefügt werden kann.

Schwarz dargestellt ist die Siede- und Taulinie. Allgemein lässt sich sagen, dass das Me-

dium links von der steigenden Flanke flüssig ist und innerhalb der Kuppel als Nassdampf gemischt. Rechts von der Taulinie ist es komplett gasförmig[7]. Da der Clausius-Rankine Zyklus nicht auf ein Medium beschränkt ist, ist die Kurve allgemein und damit nicht quantifiziert. Selbiges zählt für die verschiedenen Temperaturpunkte und die rote Linie. Limitiert wird die Effizienz des Verfahrens durch die Gesetzmäßigkeit, die durch den zwei-

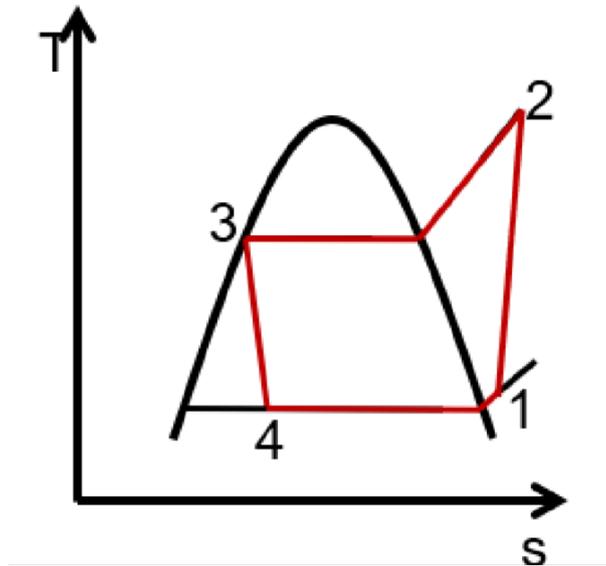


Abbildung 2.2: T-S Diagramm des Rankine Zyklus [10]

ten Hauptsatz der Thermodynamik ausgedrückt wird. Dieser besagt, dass alle natürlichen und technischen Prozesse nicht umkehrbar sind. Angewendet bedeutet das, dass bei jedem Prozess Energie, oft in Form von Wärme, verloren geht. Ein Prozess, der in einem Zustand gestartet ist und sich dieser Zustand geändert hat, verliert also Energie und kann nicht ohne Zufuhr von Energie in den ursprünglichen Zustand zurückkehren[7]. Entsprechend dem Gesetz werden auch beim Verdichter Teile der vorhandenen Energie in unerwünschte Energieformen umgewandelt, oder an der falschen Stelle freigesetzt. Letzteres bezieht sich auf den Wärmeabfluss an die Umgebung sofern diese nicht Ziel des Heizens ist. Das heißt, dass ein Verdichter mit einem Wirkungsgrad von 100% nicht möglich ist. Eine Optimierung, um möglichst nahe an diesen Optimalwirkungsgrad heranzukommen und entsprechend möglichst viel der eingehenden Energie in nutzbare Ausgangsenergie umzuwandeln, ist anzustreben.

2.1.2 Der Aufbau eines Verdichters

Verdichter sind wie der Name bereits sagt Maschinen, die die Dichte eines gasförmigen Mediums erhöhen sollen. Dafür muss gemäß des Energieerhaltungssatzes Energie aufgewandt werden[11]. Sie bilden das Gegenstück zur Turbine, bei der Energie durch Ent-

spannung eines Mediums gewonnen wird[11]. Beide können zur Kategorie der Turbomaschinen gehören. Kompressoren, welche auf dem Verdrängungsprinzip z.B. mit Kolben arbeiten werden hier nicht betrachtet, da für die Anwendung eine stetige Verdichtung gefordert wird. Der Energieinput erfolgt beim Kompressor, bzw. Verdichter, zum Beispiel über einen Elektromotor, welcher Kraft auf das Medium ausübt und es so verdichtet[11]. Bei den Verdichtern gibt es zwei Hauptkategorien. Es wird zwischen Radialkompressoren und Axialkompressoren unterschieden. Bei beiden erfolgt der Energieaustausch durch die Bewegungsenergie der Schaufeln[11]. Es kann außerdem gesagt werden, dass Radialkompressoren im allgemeinen eher genutzt werden, wenn eine hohe Druckerhöhung bei geringem Massenfluss wichtig ist[11]. Der Fokus liegt deshalb hier auf Radialkompressoren.

Die Hauptbestandteile eines derartigen Kompressors sind Nabe, Impeller und Diffusor. Die Nabe ist in der Mitte des Impellers. An ihr tritt das Medium in den Verdichter ein. Im Kompressor gelangt das Medium dann in den Impeller[5]. Dieser ist ein mittig auf der Welle gelagertes Schaufelrad. Die Scheibe auf der die Schaufeln befestigt sind wird oft Hub genannt. Die Schaufeln sind auf der jeweils dem Hub gegenüberliegenden Seite durch das Gehäuse begrenzt. Die Kontaktfläche heißt Tip. Des weiteren werden die Spitzen der Schaufeln an der Nabe als führendes Ende angesehen, während das andere Ende als folgende Kante oder trailing Edge bezeichnet wird[12]. Die Schaufeln sind gebeugt. Die konvexe Seite wird als Druckseite bezeichnet, die konkave Seite als Saugseite[12]. Der Impeller wird durch Energie von außen angetrieben und beschleunigt das eintretende Medium. Dabei wird der Druck und die kinetische Energie durch die Fliehkraft erhöht. Die Richtung des Mediums wird um 90° geändert. Im Anschluss wird das Medium in den stationären Diffusor gedrückt. Dort staut es sich und die kinetische Energie wird in Druck umgewandelt[5]. Dabei erhöht sich auch die Temperatur des Mediums. Die Verdichter sind durch verschiedene Parameter spezifiziert. Dazu gehören zum Beispiel Wellendurchmesser, die Durchmesser des Impelleraus- und Einlasses und die Spezifikationen des Rotors an sich[12]. Außerdem ist die Zahl der Schaufeln, sowie ihre Form und Neigung wichtig. Des weiteren können auch Splitterschaufeln in die Geometrie eingefügt werden, die die Strömungseigenschaften verbessern können. Die Hauptschaufeln haben eine andere Geometrie als die Splitterschaufeln. Ein Parameter den es bei beiden gibt ist der Filletradius, welcher Thema der Arbeit ist. Dabei handelt es sich allgemein um eine abgerundete Stelle. Diese kann konvex oder konkav sein[13]. In diesem Fall werden konkave Fillets betrachtet. Sie sind an der Grenze zwischen Hub und Klinge zu finden. Sie umlaufen die gesamte Schaufel, mit Ausnahme des Bereichs, an dem das Medium den Impeller verlässt. Er kann die Strömungseigenschaften und Materialstabilität beeinflussen.

2.1.3 Stand der Technik

Die Technologie der Wärmepumpe an sich ist mittlerweile weit verbreitet. Haushaltsübliche Kühlschränke verwenden das Prinzip. Besonders im Hochtemperaturbereich, wobei der Bereich der Hochtemperatur nicht klar definiert ist, gibt es noch Forschungspotential. Besonders im aktuellen Kontext des Klimawandels ist die Technologie wichtig, um fossile Brennstoffe so weit es geht abzulösen. Daran arbeitet auch das Deutsche Zentrum für Luft- und Raumfahrt (DLR) am Institut für CO₂ arme Industrieprozesse[14]. HTWPs mit einem Temperaturhub von 95 Grad Kelvin existieren allerdings bereits und sind im Einsatz. Dabei wird ein COP von 5.8 erreicht[4]. Kommerzielle Wärmepumpen sind bis zu 150 °C verfügbar[15]. Das Temperaturziel des Forschungsbereich beim DLR liegt letztendlich bei 200°C bis 400°C, und in Sonderfällen bei 600 Grad Celsius[15]. Auch wenn das Prinzip der Hochtemperatur-Wärmepumpe den Alltag bereits prägt, bedarf die Technik dennoch weiterer Forschung und Verbesserung.

Das Prinzip des Verdichters ist, wie bereits aus der Verwendung in HTWPs hervorgeht, ebenfalls weit verbreitet und erforscht. Kompressoren finden Anwendung sowohl in Wärmepumpen als auch in Motoren und sogar bei Flugzeugen[5] um nur ein paar zu nennen. Das Design eines radialen Kompressors, wenn auch auf Flüssigkeiten als Pumpe ausgelegt, wurde schon am Ende des 17ten Jahrhunderts erfunden[6]. Das Prinzip ist lange bekannt und mehrstufige Verdichter finden oft Anwendung. Die optimale Auslegung der Verdichtergeometrien ist, wie das Thema dieser Arbeit zeigt, weiterhin problematisch. Geometrien können aber durch computational fluid dynamics (CFD)- und CSM-Simulationen berechnet werden. Derartig gefundene Verdichter müssen aber auch noch gebaut werden. Dabei stellen sich weitere Herausforderungen, wie später in Kapitel 3.2 näher erklärt wird.

2.2 Umsetzung der Verdichteroptimierung

2.2.1 Kriterien und Zielbedingungen

Sowohl für die Optimierung des Verdichters als auch für die dazu gehörenden vorbereitenden Schritte gibt es bestimmte Ziele, die angestrebt werden, und Kriterien, welche eingehalten werden müssen, oder sollen.

Die Zahlenwerte, auf die die Kriterien gesetzt werden, sind vorerst arbiträr gewählt, oder durch die Material- und Geometrieigenschaften gegeben. Bezüglich des Materials wurde eine Legierung benötigt, die hohe Festigkeit mit geringem Gewicht vereinbart. Bei zu hohem Gewicht, wie zum Beispiel Stahl, werden die Fliehkräfte unter anderem durch die schnell drehenden Flügel zu stark. Bei Aluminium wären diese Kräfte geringer. Allerdings verfügt es nicht über die benötigte Festigkeit bei den Betriebstemperaturen. Letztere

sind ebenfalls ein einschränkender Parameter. Der Schmelzpunkt des Materials darf nicht zu niedrig sein. Deshalb wird eine Titanlegierung verwendet. Sie trägt die Bezeichnung Ti6Al4V ELI. Sie hat eine bei den Temperaturen ausreichende Härte bei geringem Gewicht. Eine Zusatzbedingung welche das Material auch erfüllt ist die Korrosionsfestigkeit des Materials. Das ist nötig, da wie aus den Grundlagen in Kapitel 2.1.1 hervorgeht Wasser verdichtet wird. Entsprechend könnte z.B. Eisen Rost ansetzen, was das Material schwächen würde und Fremdteilchen freisetzen könnte.

Auch bei passenden Materialeigenschaften kann der Verdichter bei zu hoher Last kaputtgehen oder sich schneller abnutzen. Daraus ergibt sich, dass Spannungen im Material einen bestimmten Wert nicht überschreiten dürfen. Dieser Wert, der unter der tatsächlichen Belastungsgrenze liegen muss, um Unfälle zu vermeiden, ist eine der Einschränkungen in der Prozesskette. Des weiteren muss noch ein bestimmter anwendungsabhängiger Massenfluss erreicht werden.

Damit die Optimierung ausreichend nahe an der Realität liegt, wurden auch Anforderungen an das zu generierende Netz gestellt. Die Untergrenze der Elemente liegt bei 50.000 Elementen. Damit die Berechnungen aber nicht zu komplex werden, soll das Netz nicht mehr als 150.000 Elemente haben. Zusätzlich soll jede Schaufel der Dicke nach mit mindestens zwei Elementen vernetzt werden, die Hauptscheibe mit dreien. Später, nach den ersten Meshversuchen, wurde durch das Calkulix Programm noch die Anforderung ergänzt, dass das Netz keine Pyramidensegmente enthalten darf. Dieser Zusatz ergab sich dadurch, dass es in Calculix keine Elemente gibt, welche Pyramiden handhaben[16]. Anzumerken ist, dass die Qualität des Netzes nur an der dargestellten Beispielgeometrie getestet wurde. Da sich die Geometrie des Verdichters während der Optimierung ändert, sind die erreichten Elementzahlen nur für die Testgeometrie gültig. Es wird aber davon ausgegangen, dass ein guter Ansatz bei vergleichbaren Geometrien zu ähnlich guten Ergebnissen führt. Diese Anforderungen an die Geometrie treten bis auf weiteres an die Stelle einer Analyse der Netzkonvergenz.

2.2.2 Automatisierte Prozesskette zur Optimierung

Für die Optimierung, welche in Kapitel 4.1 behandelt wird, braucht es eine Prozesskette. Diese Kette soll die Schritte von den Eingangsparametern über eine komplette Geometrie zu den Ausgangsparametern enthalten. Die verwendete Kette war im Institut bereits vorhanden. Sie läuft auf Linux Betriebssystemen. Die Versionen der verwendeten Programme sind den Methoden aus Tabelle 1.1 zu entnehmen. Der Programmcode, der ab Kapitel 2.2.3 beschrieben wird, wäre bei Erfolg in die Prozesskette eingebunden worden. Nachdem in Kapitel 2.2.6 beschriebenen Fehler konnten die neuen Elemente nicht eingebunden werden.

Als Eingangsparameter erwartet die Kette ein Standard S2M Netz und die Werte für

die Freiheitsgrade. Ein S2M Netz ist dabei meridionaler Querschnitt der Basisgeometrie. Das heißt bildlich gesprochen, dass der halbe Querschnitt der Grundfläche, auf der die Schaufeln befestigt sind, von der Geometriemitte aus betrachtet wird. Die Achse, auf der die Antriebswelle des Verdichters liegt, bildet eine Seite des 2D-Koordinatensystems. Im ersten Schritt wird das Programm GTA verwendet, um die Hub- und Tiplinien der Schaufeln zu verändern. Danach wird bereits das erste Netz für die Geometrie erzeugt. Wichtig dabei ist, wie häufig die Geometrie in K- und I-Richtung geschnitten wird. Dadurch ergibt sich die Zahl der Hilfslinien, welche die Geometrie bei späteren Schritten stützen, und damit die Genauigkeit. Es handelt sich dabei nicht um ein FEM-Netz, sondern um ein Oberflächennetz, das nur dazu dient, die festgelegte Geometrie ausreichend genau zu exportieren. Das Netz ist an diesem Punkt noch zweidimensional.

Im nächsten Schritt werden die Schaufeln mit Bladegen erzeugt. Als Ausgabe liegen verschiedene Dateien vor, welche zum Beispiel die Schaufel mit Verlängerung über die eigentliche Außenbegrenzung beinhaltet, was die Darstellung genauer macht, und bereits zurechtgeschnitten. Erstere erwähnte Datei wird zum Beispiel in Fidelity für das Erzeugen der Geometrie verwendet, welche erst später zurechtgeschnitten wird. Die zweite genannte Datei hingegen wird für die Bestimmung der Länge des Fillets für das Fräsen herangezogen. Im nächsten Schritt wird die Schaufel zum einen durch Autogrid vernetzt und zum anderen durch Gmsh. Die Autogrid Vernetzung wird für die CFD-Simulation verwendet. Das wird mit PREP durch Parallelisierung beschleunigt. Das Gmsh-Netz, welches eigentlich durch das Fidelity-Netz ersetzt werden sollte, wird für die CSM-Analyse verwendet. Diese wird in Calculix ausgeführt. Sie läuft parallel zur CFD-Analyse. Für die Strömungsanalyse werden die Daten durch das Gmc Programm vorbereitet. Die Parameter für den Löser werden bestimmt. Die eigentliche CFD übernimmt das Programm TRACE. In POST werden die Ergebnisse ausgewertet, wobei auch PyMesh gebraucht wird. Kennwerte werden bestimmt. Sobald die CSM-Simulation in Calculix durchgelaufen ist, werden auch deren Ergebnisse ausgewertet. Im letzten Schritt wird geprüft, ob die Nebenbedingungen erfüllt sind und welchen Wert die Zielbedingung erreicht hat. Die Ergebnisse werden in Dateien abgespeichert.

2.2.3 Geometrierstellung mittels Fidelity

Der in erster Linie bearbeitete Teil der Prozesskette ist die Erstellung der Geometrie des Verdichters mit anschließender Vernetzung. Dieser Schritt ist für die später laufende Optimierung nötig. Das Netz hat dabei die Aufgabe das zu berechnende Stück in Elemente aufzuteilen. Anstatt den ganzen Rotor zu betrachten werden zuerst die Eigenschaften pro Element berechnet. Eigenschaften bezieht sich hierbei auf Spannung, Dehnung, Kräfte und Reaktionskräfte des Werkstücks[17]. Dabei sind auch die Materialeigenschaften wichtig.

Die Lösung der Gleichungen, welche Aussagen über diese Eigenschaften treffen, wird über FEM-Programme berechnet. Die Hauptaufgabe, welche der Mensch an sich noch übernehmen muss, ist das Bilden des Netzes. In diesem Anwendungsfall, bei dem die gesamte Prozesskette automatisiert werden soll, ist das aber nicht möglich. Eine manuelle oder überwachte Vernetzung würde den Prozess verlangsamen und ist für die angestrebte Automatisierung nicht praktikabel. Deshalb wurde das Netz bisher mit einem Gmsh Python Programm vernetzt und exportiert. Um das Programm möglicherweise zu verbessern, soll an die Stelle Gmshs Fidelity treten und die Netzgenerierung der Geometrie übernehmen. Da die verfügbare Version von Fidelity die vorliegenden Daten nicht direkt lesen kann, geschieht das über die Python Programmierschnittstelle (API) der Anwendung und ein entsprechendes Python Programm, das aus den Parametern und Punkten, welche vom vorherigen Arbeitsschritt kommen, die Verdichterscheibe und die Blätter in Fidelity nachstellt.

Das Programm wurde auf Windows 10, bzw. 11 entwickelt. Pfade und Systemcommandos wurden unabhängig vom Betriebssystem geschrieben, weshalb es auch auf Linux-Maschinen laufen sollte. Das wurde aber nicht final getestet. Im Programm werden außerdem noch einzelne Fragmente aus dem Gmsh-Programm z.B. für mathematische Umformungen verwendet. Auf diese wird nicht näher eingegangen. Der Aufruf des Programms erfolgt durch den vorherigen Arbeitsschritt über ein Systemkommando, wobei direkt die Ausführung des im folgenden beschriebenen Pythoncodes startet. Die Aufrufkommandos unterscheiden sich je nach Betriebssystem. Zur Vereinfachung und als Beispiel wurden ausführbare Dateien für Linux und Windows mit den Kommandos geschrieben. Bei der Windows-Datei muss der Systempfad manuell auf den Installationspfad von Fidelity gesetzt werden. Außerdem ist anzumerken, dass im Pythonprogramm der Pfad zur aktuellen Datei bei dieser Art der Ausführung an den Systempfad angehängt werden musste, damit Nebendateien importiert werden konnten. Dazu gehört das in Kapitel 2.2.6 beschriebene Programm und eine Inputdatei, welche noch vom Gmsh Programm übernommen wurde. Sie enthält Systemparameter. Nach Ausführung des jeweiligen Startscripts wird die Hexpress Umgebung in Fidelity geladen, um dort die Geometrie aufzubauen. Für die Geometrie werden die Punkte eingelesen. Ab hier wird eine bestimmte Ordnung der Punktdateien vorausgesetzt. Bestimmte direkt ansprechbare Punktgruppen werden zuerst zu Linien, bzw. oft Splines, geformt.

Splines allgemein sind dabei Kurven, welche durch Funktionen über bestimmte Punkte definiert werden. Sie erfüllen bestimmte Eigenschaften. Damit eine Linie als Spline bezeichnet wird, muss sie ausreichend schnell berechenbar und eindeutig beschreibbar sein.[18] Die hier verwendete Sonderform wird als B-Spline oder auch Basic Spline bezeichnet. Durch die Splines und ihre Eigenschaften können komplexe Geometrien modelliert und gespeichert werden, ohne dass alle Flächenpunkte zu sichern sind.

Die erzeugten Splines werden dann entweder mit einer Oberfläche verbunden oder ei-

ne Oberfläche wird aus ihnen durch Rotation erstellt. Beim zweiten Typ wird von einer Revolutionsoberfläche gesprochen. Im Programm wird die Generierung der Flächen und Linien über Aufrufe gesteuert. Die Punkte werden als Liste von Tupeln aus Koordinaten gesehen. Sie kommen aus Daten im ASCII *Dat* Format, welches z.B. das Programm Tecplot darstellen kann. In den verwendeten Dateien findet sich zuerst ein Header welcher die enthaltenen Variablen benennt. Danach kommen Zahlen, welche die Punkte, die darauf folgen, logisch ordnen. Die einzelnen Tupel zu den Punkten werden durch sie zu einer Matrix angeordnet. Die Punkte in den Zeilen und Spalten ergeben dann Linien. Sie definieren die Geometrie.

Alternativ werden teilweise bei ausreichenden Wissen bestimmte Punktsätze direkt angesprochen, welche wieder Schlüssellinien der Geometrie ergeben.

Neben den Linien der eigentlichen Geometrie werden auch Hilfsflächen und Linien erzeugt. Sie werden verwendet, um die Geometrien später zurechtzuschneiden.

Dabei wird zuerst die Grundfläche auf der die Rotorblätter angebracht sind erzeugt. Sechs verschiedene Linien, davon zwei Splines, formen die Form der Grundfläche. Ursprünglich sollten aus diesen dann so wie schon im Gmsh Programm direkt Flächen erstellt werden, welche durch eine Drehung um die X-Achse dann einen Drehkörper erstellt hätten. In Fidelity wurden die Kurven, obwohl zusammenhängend, jedoch als getrennte Kurven betrachtet. Eine Fläche konnte nicht direkt zwischen ihnen erstellt werden. Der erste Ansatz war es, die Flächen zwischen je zwei der Linien zu erstellen, wobei die parallel zur X-Achse ausgelassen wurden. Das funktionierte zuerst und die Flächen konnten erstellt werden. Bei dem Versuch aus diesen zwei Flächen, obwohl sie zusammengenäht worden waren und als ein Sheet zählten, einen Körper zu erstellen, wurden zwei erzeugt. An diesem Punkt hätte eine boolesche Vereinigung das Problem behoben. Für diese Vereinigung hätten vorher zusätzlich die Flächen, welche gedreht wurden, gelöscht werden müssen, da beim Drehen der neue Körper in der selben Einheit im Baum gespeichert wird wie die erzeugende Fläche. Dasselbe gilt auch, wenn eine Fläche extrudiert wird. Boolean Operationen wie die Vereinigung funktionieren in Fidelity allerdings nur bei gleichen Objekttypen. Für das Löschen hätten die Unterobjekte also auf ihren Typ untersucht werden müssen, um alle als Fläche gelisteten Körper zu löschen. Da allerdings schon mehrere Schritte für diese Vorgehensweise benötigt wurden, wurden zur Optimierung stattdessen die ursprünglichen Linien direkt um die X-Achse gedreht. Um das Erzeugen von Flächen an den Enden zu vermeiden, wurde die Fläche vorerst auf 360 Grad ausgelegt. Die entstandenen Flächen wurden dann zu einem soliden Körper, kurz Solid, zusammengenäht. Eine weitere Fläche, diesmal leicht gewellt, der groben Form der späteren Blätter angepasst, soll dann genutzt werden, um das eigentlich wichtige Segment aus der Grundplatte zu schneiden. Der Anteil, der wichtig ist, ergibt sich aus der Gesamtzahl der Blätter. Zur Beschleunigung der Berechnung wird nur ein Rotorblatt und ein Splitterblatt auf die Scheibe gesetzt. Entsprechend wird nur der Teil der Platte gebraucht, die vom Anteil her zu diesen Blattpaar

gehört.

Für den Schnitt könnte die gewellte Fläche an zwei Querschnitten die Grundfläche schneiden, um dann den größeren Teil wegzuworfen. Einfacher ist es aber, aus der Wellfläche ebenfalls einen Drehkörper zu machen. Da die Fläche einen Radius mehrmals schneidet, wird die direkte Bildung eines Drehkörpers von Fidelity nicht unterstützt. Als Workaround wurde dieser Teil, der nicht innerhalb der Grundfläche lag, mit einem Zylinder, der anstelle der späteren Achse des Rotors erzeugt wurde, ausgeschnitten. Damit konnte der Drehkörper erzeugt werden und entsprechend das benötigte Segment der Grundplatte mit einer boolschen Operation, welche die gemeinsame Fläche des Hilfsobjekts und der Fläche betrachtet, ausgeschnitten werden. Die Hilfskörper wurden im Anschluss gelöscht.

Zwei weitere Hilfsflächen werden im Anschluss erzeugt. Sie begrenzen die späteren Klappen nach oben und unten. Ihre Form ähnelt der der Oberseite der Grundfläche. Sie werden durch die selbe Funktion mit anderen Eingabeparametern erzeugt. Was noch bleibt sind die eigentlichen Blätter und die anschließende Vereinigung der Komponenten. Für die Generierung beider Blätter wird dieselbe Methode verwendet. Linien, welche den Mantel des Blatts, einschließlich des Fillets an der Verbindungsstelle zur Grundplatte, definieren, werden eingelesen und als B-Spline dargestellt. Ursprünglich im Gmsh Programm wurden diese Linien, die in sich pro Ebene geschlossen sind, mit einer Loftoberfläche verbunden. Die Linien sind zusammen mit dem Problem, welches bei diesem Vorgehen in Fidelity auftritt, in Abbildung 2.3 dargestellt. Dort sind in Schwarz die komplett umlaufenden Linien dargestellt. Eine ist zum leichteren Folgen Pink hinterlegt. Die Fläche, welche bei einem Loft über die umlaufenden Linien erstellt wird, ist hier grün. Wie erkennbar ist, wurde nur die eine Hälfte der Fläche erstellt.

Der erste Lösungsansatz war es, eine Lücke in die Linie und damit die Fläche einzubauen, und diese im Anschluss über eine weitere wieder zu schließen. Durch diese Aufteilung konnte ein kompletter Blattmantel erzeugt werden. Da das Ziel war, aus diesem Mantel einen Körper zu erhalten, war der nächste Schritt, den Mantel mit den vorher erzeugten Hilfskörpern zurecht zu schneiden und aus Kopien der Hilfskörper die Begrenzungsflächen auszuschneiden. Anstatt von boolschen Operationen wurden hier Imprint-Operationen, bei denen eine Fläche eine andere zerteilt, verwendet, da boolsche Operationen in Fidelity nur bei einem kompletten Schnitt funktionieren. Imprints hingegen hinterlassen auch bei teilweiser Überdeckung, welche Flächen nicht klar teilt, eine Spur, welche zusammen mit anderen Imprints zu neuen Flächenabstufungen werden können. Zwischenstufe zwischen den beiden Versionen war es, die oberste und unterste Linie jeweils zu extrudieren, um damit einen Schnitt mit Boolean-Operationen zu ermöglichen. Das wurde aber durch die Form des Blattmantels erschwert und hätte durch die überragenden Teile zu Ungenauigkeiten geführt. Deshalb wurden mit Imprints die überstehenden Teile des Blattmantels abgetrennt und Abdrücke in das Hilfsobjekt gestanzt. Um Problemen durch Abweichungen bei dem Füllsegment, welches für das Ausschneiden des oberen und unteren Abschlusses

noch gebraucht wurde, entgegenzuwirken, wurde außerdem ein Schnitt, der eigentlich erst am Ende nach der Körpererstellung gemacht werden sollte, vorgezogen. Der Schnitt löscht alles außerhalb eines bestimmten Radius. Dadurch wurde auch das Füllsegment, welches außerhalb lag, überflüssig. Gleichzeitig ergab sich daraus, dass auch diese äußere Wand mittels Imprints aus dem hohlen Zylinder, welcher für den Schnitt am Radius verwendet wurde, ausgeschnitten werden musste. Alternativ hätte auch ein Loft über die nun offenen Enden funktioniert. Das hätte aber das Ergebnis leicht verfälscht. Außerdem sind die Linien nicht direkt verfügbar und hätten erst aus den Flächen extrahiert werden müssen, was durch die unklare Reihenfolge erschwert worden wäre. Durch das Ausschneiden über Im-

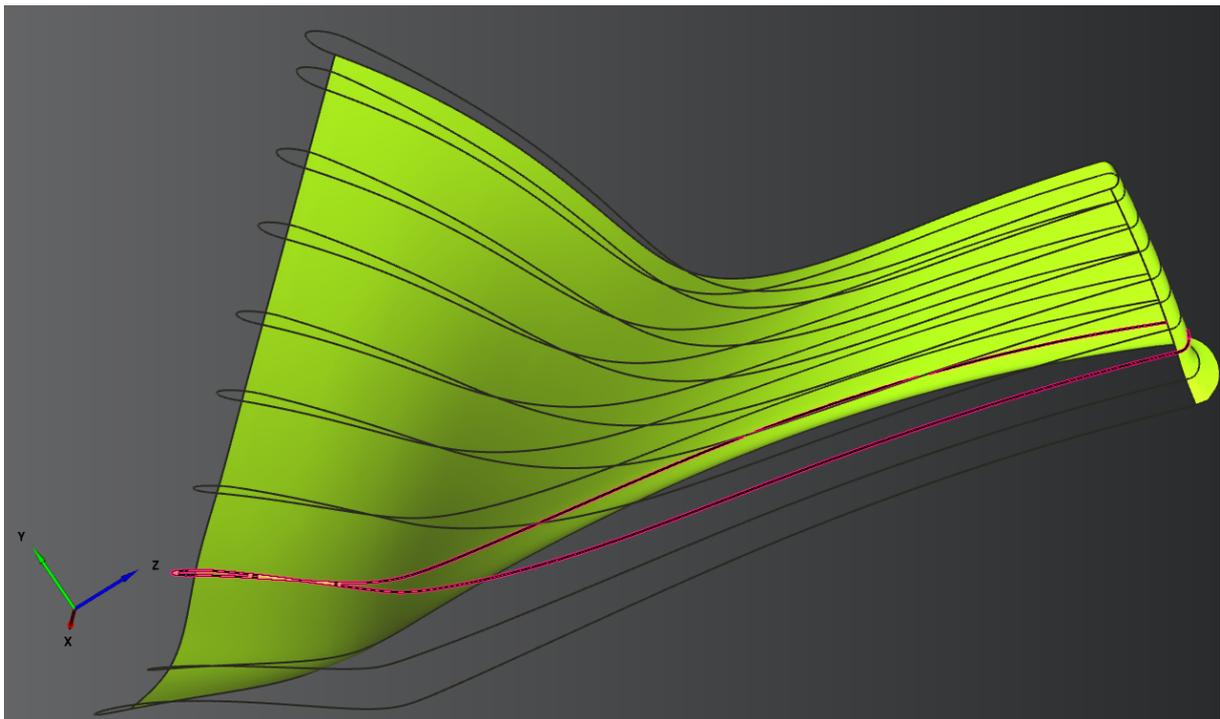


Abbildung 2.3: Fehlgeschlagene Generierung des Blattmantels bei ganzen Kurven

prints wurden drei neue Flächen erzeugt, wobei die obere Fläche und die untere ebenfalls durch den Zylinder gestutzt wurden. Die Auswahl, welche der Teilflächen die benötigte Fläche war, wurde an diesem Punkt vorerst über Indizes festgelegt. Da die Reihenfolge aber beim ersten Aufruf im Vergleich zum zweiten Aufruf der Methode anders war, sollte das nach Konzepttests auf Größenvergleiche der Fläche umgestellt werden. Nachdem die Flächen derartig vorbereitet wurden, sollten sie zu einem Körper vernäht werden. Beim Versuch konnten allerdings nicht alle Flächen miteinander verbunden werden. Der Grund dafür wird in mathematischen Ungenauigkeiten in der Geometrie vermutet.

Der nächste Ansatz war es, anstatt des Ausschneidens von Flächen früher zu einem Körper zu kommen und die überragenden Bestandteile des Blattes nachträglich zu kürzen. Beim Schneiden von Körpern werden die Flächen direkt generiert. Das gewählte Vorgehen war es, über die oberen und unteren Linien eine Fläche zu legen, um so den Deckel direkt

zu erhalten. Über einzelne Linien kann aber nicht geloftet werden. Deshalb wurden die Konturlinien des Blattmantels in zwei Teile und damit der Mantel selbst in zwei Flächen, Saug- und Druckseite, aufgeteilt. Anders als bei der Füllfläche sind beide Flächen groß genug, um die Form genauer wider zu spiegeln. Sie weisen geringere Ungenauigkeiten an den Berührungsflächen auf. Für die Erstellung wurden die Punkte der Gesamtlinien geteilt. Über die Kanten der zwei Flächen konnten Loftoberflächen gelegt werden, um die Flächen zu einem Körper zu verbinden. Die Flächen waren allerdings weit von den Zielflächen entfernt und schnitten teilweise den Mantel. Um bessere Oberflächen zu bekommen, sollten Kanten gefunden werden, die möglichst ähnlich verlaufen und sich dabei nicht treffen. Deshalb wurde die Hauptfläche erneut aufgeteilt. Dabei kamen extra Flächen für die führende und folgende Kanten des Rotorblatts dazu. Gleichzeitig wurden immer die beiden Enden der Saug- und Druckseite der obersten bzw. untersten Konturlinie durch eine Linie verbunden, sodass sich grob ein rechter Winkel ergab. Mit dieser und den Kanten der neuen Flächen konnten Oberflächen gebildet werden, welche zusammen mit den ebenfalls neu gebildeten Flächen zwischen den Saug- und Druckflächenenden einen sauberen Abschluss oben und unten an den Blättern bildeten. Ursprünglich schlug das Bilden einer Fläche über diese neu eingeteilten Linien fehl. Das konnte behoben werden, indem die Punktemenge verringert wurde. Die Ungenauigkeit, die dadurch dazukommt, wird als ausreichend gering angesehen.

Diese mittlerweile zehn verschiedenen Flächen konnten zu einem Solid zusammengenäht werden. An diesem Punkt ist aufgefallen, dass die Flächen, welche die Fillets beinhalten, also die Saug-, Druck- und Spitzenflächen, schlecht generiert wurden. Das liegt daran, dass die Oberfläche bereits vor dem eigentlichen Biegung durch das Fillet sich auf die Biegung vorbereitet. Dadurch entsteht eine Art Wellenform, welche im Abschnitt vor dem Fillet für eine Eindellung sorgt. Eine Lösung für dieses Problem wäre die Generierung der Flächen in kleineren Segmenten, wodurch das Fillet die anderen Teile nicht beeinflussen könnte. Dadurch würde aber auch die Biegung der Fillets verloren gehen. Auch ein Mittelweg zwischen der bestehenden Geometrie und den Segmenten ist schwer möglich. Dieser hätte vorgesehen, das Fillet von der alten Geometrie zu nehmen und den Rest durch kleinere Segmente neu zu generieren. Das hieße, dass eine komplexe Schnittfläche direkt über dem Fillet erzeugt werden müsste, was Raum für Ungenauigkeiten eröffnet. Bei der Verwendung von weniger Linien, um das Loft zu vereinfachen, wird das Überschwingen nur deutlicher. Da diese Varianten als nicht gut genug anwendbar gesehen werden, wurden stattdessen mehr Linien verwendet. Diese wurden aus den Punkten der existierenden Linie berechnet. Dazu wurde immer der Punkt in der Mitte zweier Punkte von Linien darüber und darunter zu einem Punkt für eine neue Linie. Auf diese Weise können mehr Randparameter für die Oberflächenbildung geliefert werden. Das Problem des Schwingens wird dadurch nicht behoben, dessen Stärke kann jedoch mit steigender Anzahl weiterer Linien geschwächt werden, bis es ausreichend wenig in die Geometrie eingeht. Zur leich-

teren Handhabung werden die Flächen in eine Gruppe verschoben, welche im Anschluss umbenannt wurde, da die direkte Benennung fehlschlug. Statt des angegebenen Namens wurde der Name einer der Oberflächen mit dem Keyword "merged" versehen und als Name der Gruppe verwendet. Entsprechend wurde mit regulären Ausdrücken nach dem Wortteil in den Namen aller Obergruppen gesucht, um die Referenz zu erhalten und das Objekt umzubenennen. Nachträglich erhielt jede Fläche noch eine eigene Gruppe, damit sie im Vernetzungsbereich besser angesprochen werden kann. Die Untergruppen sind weiterhin in eine einzige Obergruppe eingliedert. Das Solid muss an diesem Punkt noch von den Hilfsflächen zurechtgeschnitten werden. Um das Problem mit verschiedenen Teilen des Körpers zu vermeiden wurde die obere Hilfsfläche nach oben extrudiert und so zu einem Körper gemacht. Durch das Löschen der Bestandteile des Körpers, welche sowohl zum Hilfskörper als auch zum Blatt gehören, konnte das Problem mit verschiedenen Flächen mit womöglich wechselnden Indizes hier umgangen werden. Für die untere Fläche ist der Schnitt nicht nötig, da überstehende Bestandteile ohnehin in die Schaufelscheibe übergehen. Da das Zurechtschneiden vom ursprünglichen Programm her in einer Funktion zusammen mit der gesamten Blatterstellung gekapselt war, wird die Extrusion der Hilfsobjekte aktuell pro Aufruf der Methode ausgeführt. Dafür wird das Original kopiert. Gearbeitet wird dann dennoch mit dem Original, da die Referenzierung des Duplikats über Rückgabewerte fehlschlug. Auch wurde z.B. beim Verschieben zwischen Einheiten im logischen Baum der Geometrien eine namentliche Referenz auf das Duplikat weiterhin auf das ursprüngliche Objekt bezogen. Das Verhalten der PythonAPI wird über die Verwendung des Originals umgangen. Nach der Schnittoperation für die das Hilfsobjekt gebraucht wird, wird das Original gelöscht und das Duplikat erhält den Namen des eben gelöschten Objekts. Dadurch konnte diese Schwierigkeit umgangen werden. Für den früher bereits genannten Radiuschnitt wurde ein Zylinder erstellt. Nur Bereiche, die sowohl in diesem als auch in den Blattkörpern lagen, wurden beibehalten. Die Zylinder wurden nicht gelöscht und außerhalb der Methode noch zum Zurechtschneiden der Grundfläche verwendet.

Das letzte Problem vor dem Komplettkörper war ein Hohlraum zwischen den Blättern und der Grundfläche. Der erste Ansatz für dieses Problem wäre es gewesen, den Hohlraum nachträglich zu löschen. Da das aber mehr Wissen über die Geometrie voraussetzt und nicht ausgeschlossen werden kann, dass sich bei anderen Geometrien mehrere Zwischenräume bilden, wurde stattdessen die Bildung des Hohlraums verhindert. Da der Hohlraum dadurch entstanden ist, dass sich die Geometrien nicht vollständig überlappen, wurde die Bodenfläche, oder zumindest der mittlere Teil davon, welcher den Hohlraum betrifft, extrudiert. Eine direkte Erweiterung nach unten entlang der X-Achse schlug fehl, weshalb die Fläche nach schräg unten extrudiert wurde. Die Länge der Extrusion musste dabei kleiner sein als die Breite der Grundfläche an jeder betroffenen Stelle, damit die Geometrie, von dem Löschen des Hohlraums abgesehen, nicht verändert wird.

Die Methode wurde zweimal mit verschiedenen Koordinaten ausgeführt, um so das Hauptblatt und das Leitblatt zu erzeugen. Da in den Methoden Teile der Geometrie oder Hilfsflächen oft mit einer bestimmten Bezeichnung generiert bzw. mit ihr angesprochen werden, wurden den Funktionen, welche mehrmals aufgerufen wurden, Zusatzbezeichnungen oder Indizes mitgegeben, welche den Namen neuer Bestandteile angehängt wurden, um Doppelungen zu vermeiden und die Zuordnung nicht zu verfälschen. Da dieses System bei Namen, welche nicht aufgerufen werden, nicht immer verwendet wurde, gibt das Programm Warnungen über Umbenennungen aus. Des weiteren wird ein Fehler bei einer boolschen Operation ausgegeben. In diesem Fall ist das jedoch gewollt, da über den Fehlercode bestätigt wird, dass die Hauptschaufel und Leitschaufel sich nicht schneiden. Könnten sich die beiden vereinen, wäre die Geometrie für die Anwendung fehlerhaft und das Programm würde enden.

Berühren sich die beiden Schaufeln nicht, so werden sie mit der Scheibe über eine Boolean-Operation zu einem Körper vereint. Das Resultat ist in Abbildung 2.4 zu sehen. Die Flächen sind dabei zur Unterscheidung verschieden gefärbt. Die längere Schaufel links ist die Hauptschaufel. Die kleinere rechts die Splitterschaufel. Die blaue Fläche, welche im Vordergrund zu sehen ist, ist die Außenseite, welche vorher durch das Zurechtschneiden mit dem Zylinder gebildet wurde.

2.2.4 Vernetzung mit Fidelity

Nachdem der Körper gebildet wurde, wird er in den Domänenraum übertragen. Dadurch werden alle nicht benötigten Teile zurückgelassen und können von diesem Punkt an ignoriert werden. In dieser Umgebung werden Voreinstellungen für die Vernetzung getroffen. Die wohl wichtigste davon ist das Setzen des Basispunkts. Auch wenn die verschiedenen Flächen in Fidelity als Körper erkannt werden, muss für das Vernetzen dennoch festgelegt werden, wo sich die Innenseite, also die eigentliche Masse des Körpers befindet. Das geschieht über den Basispunkt. Aus dieser Domäneninstanz wird im nächsten Schritt eine Vernetzungsinstanz mit Hexpress als Vernetzer gebildet. Damit wird in die dritte Ansicht von Fidelity gewechselt. Sämtliche Aktionen werden weiterhin über das Script ausgeführt, obwohl die Sinnhaftigkeit der Einstellungen vorher in der GUI getestet wurde.

Einstellungen werden vor allem an zwei Stellen vorgenommen. Es gibt zum einen Optionen, welche den Vernetzer selbst einstellen und zum anderen solche, die die Geometrie oder Teile davon betreffen. Zudem gibt es noch die grundlegende Unterscheidung von wo das Netz starten soll. Die ersten Netze wurden mit der Standardeinstellung 'Volume to Surface Mesh' generiert. Dabei startet das Netz im Kern des Objekts, quasi an den Knochen, und breitet sich von dort nach außen bis hin zu den Oberflächen aus. Das Ergebnis war ein Netz, das die Geometrie stark verfälschte und nur unzureichend erfasste. Die Grundplatte wurde größtenteils erfolgreich vernetzt, wie in Abbildung 2.5 links zu sehen

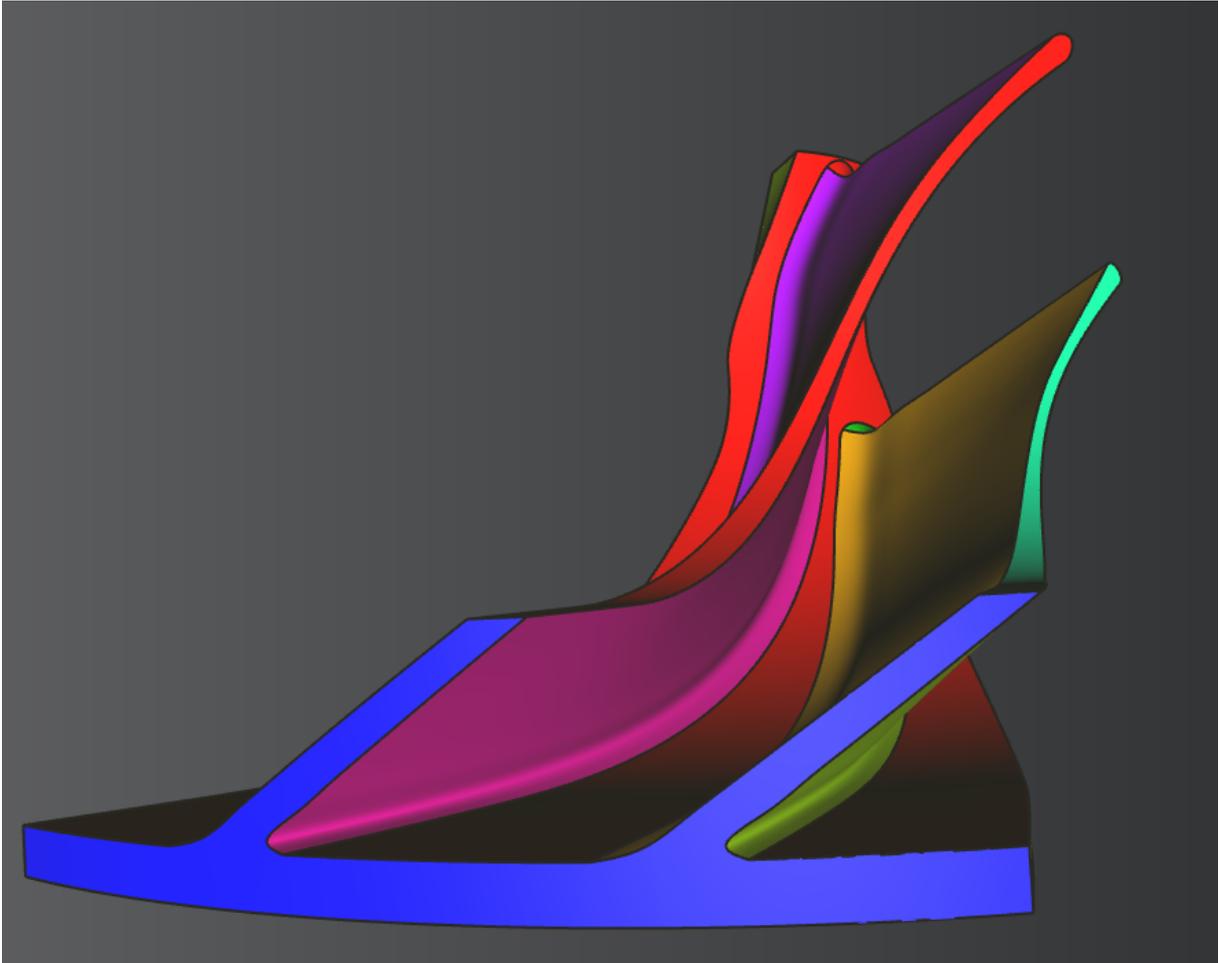


Abbildung 2.4: Scheibe mit einer Splitt(rechts)- und einer Hauptschaufel(links) als ein Körper in Fidelity

ist. Anders sah es bei den Klingen aus. Da diese schmäler und komplexer sind als die Scheibe, war das erste Netz weniger zutreffend. Das Ergebnis beinhaltete spitz zulaufende Rotorblätter, welche zum Teil Lücken aufwiesen. Außerdem wurden die Fillets, welche für die späteren Simulationen wichtig sind, besonders an der Seite, zu der sich das Blatt neigt, je nach Zellgröße sehr kantig dargestellt und teils nur mit einem Element. In der Abbildung liegt ein solcher Fall in dem roten Kreis vor. Obwohl dieses Netz stark von der eigentlichen Geometrie abweicht, hatte es bereits über 140.000 Zellen. Auch wenn eine weitere Verfeinerung die Zahl der Zellen zu sehr erhöhen würde, wurde dennoch ein Versuch mit doppelter Auflösung gestartet. Das Netz mit einer Standardauflösung von einem Viertel Millimeter ist rechts in der Abbildung zu sehen. Auch wenn das Netz dadurch die Geometrie um einiges genauer darstellt, treten, wie im roten Kreis sichtbar ist, dennoch Fehler in Form von Zuspitzungen bei der Klinge auf. In diesem Versuch belief sich die Zahl der Elemente auf über eine Million. Durch weitere Einstellungen wie zum Beispiel andere Elemente, hätte diese Zahl etwas verringert werden können. Eine Feinheit, die ein ausreichend akkurates Netz geliefert hätte, hätte auf diese Art dennoch zu viele Zellen

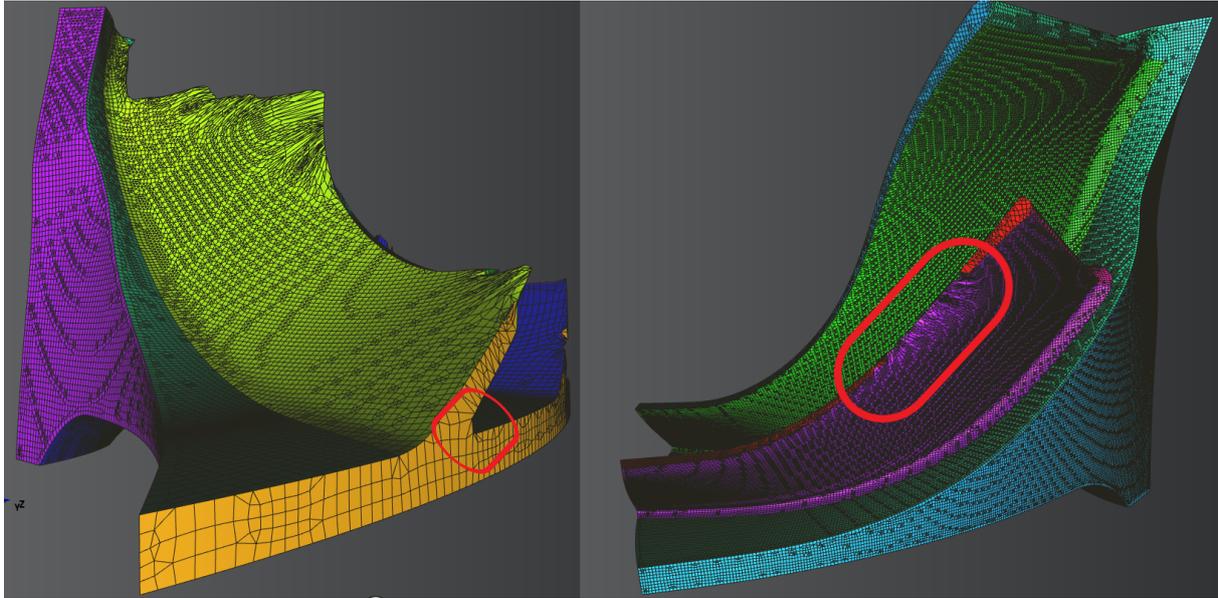


Abbildung 2.5: Fehlerhafte Volume to surface Netze, rechts mit doppelt so hoher Netzauflösung

gehabt. Direkte Umstellungen, um die Kantenerfassung zu erzwingen, oder die Kurven weiter zu unterteilen, erbrachten keine Verbesserung. Erst ein Vernetzungsansatz, welcher den Geometriepunkt aus der Domäne ignorierte, mit einer Standardzellengröße von einem halben Millimeter und der Einstellung zum Erfassen der Oberflächenkanten, brachte einen Fortschritt. Bei diesem Ansatz waren sowohl die Blätter als auch die Scheibe ohne Geometrieverlust vernetzt. Das Fillet an der bereits genannten Seite mit Neigung war wie im vergrößerten Ausschnitt in 2.6 links weiterhin unzureichend vernetzt. Die Zahl der Elemente ist dabei mit 250.000 schon näher an der Zielregion. Um die Zahl der Elemente zu verringern, wurden im nächsten Versuch Startzellgrößen von einem Millimeter gewählt. Damit die Fillets sauber vernetzt werden, wurde die Kurvenverfeinerung aktiviert. Das Resultat ist rechts in 2.6 gezeigt. Das Netz ist, wie in der Vergrößerung in der oberen Ecke zu sehen ist, um die Fillets herum stark verfeinert. Das Fillet wird damit gut abgebildet. Die Feinheit an der oberen Seite der Blätter ist allerdings mit zum Teil nur einem Element zu gering. Obwohl Teile des Netzes noch feiner generiert werden müssten, ist die Zahl der Elemente aber weiterhin zu hoch und hat sich im Vergleich zum linken Versuch fast verdreifacht. An diesem Punkt wurde ersichtlich, dass ein anderer Ansatz verfolgt werden muss. Deshalb wurde der Generierungsansatz umgestellt. Anstatt vom Kern der Geometrien nach außen zu gehen, wurde zuerst das Netz der Oberflächen erstellt, bevor es sich nach innen fortsetzte.

Bereits beim ersten Schritt, bei dem nur die Netzart umgestellt wurde und alle anderen Optionen vom vorherigen Netz übernommen wurden, wurde ein passables Netz mit weniger Elementen generiert. Im Rahmen der Umstellung wurde allerdings die bevor-

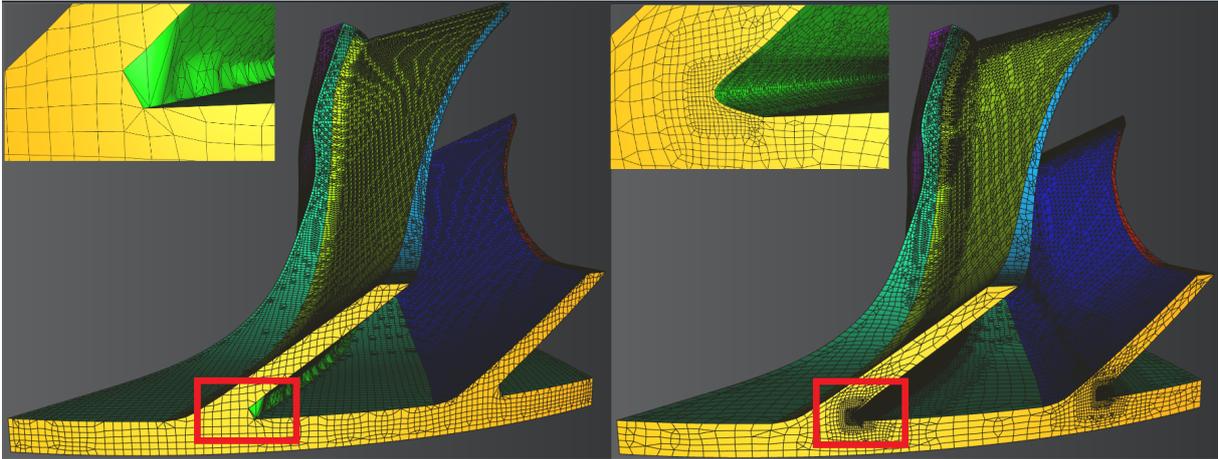


Abbildung 2.6: Vernetzte Geometrien, links mit 0.5mm Zellgröße, links mit 1mm Zellgröße und Kurvenverfeinerung

zugte Elementart von einem Gemisch aus Körperformen auf eine Hexaeder dominierte Selektion gelegt. Die bisher verwendete reine Mischoption ist für Oberflächen-zu-Körper Netze nicht verfügbar. Das Resultat ist in Abbildung 2.7 links zu sehen. Der vergrößerte

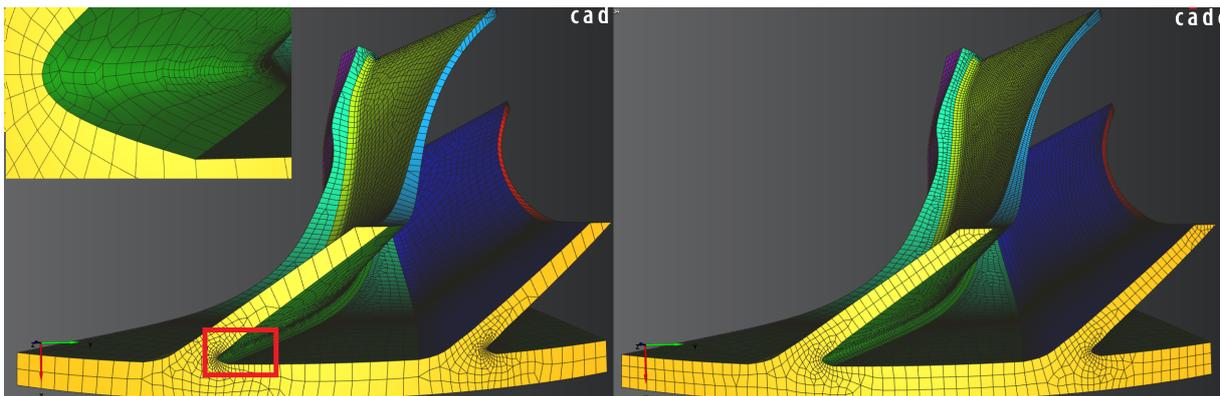


Abbildung 2.7: Vernetzte Geometrien, links mit 0.5mm Zellgröße, links mit 1mm Zellgröße und Kurvenverfeinerung

Bildausschnitt zeigt einen Teil des Fillets. Es ist in dieser Vernetzungsform, bei der eine Verfeinerung an den Kurven weiterhin aktiv ist, ausreichend nahe an der tatsächlichen Geometrie. Eine weitere Annäherung wäre nur mit noch feineren Netzen, was die Bearbeitungszeit hochtreiben würde, oder mit gerundeten Objekten möglich. Letztere sind in Fidelity vorgesehen. Die Funktion hat aber noch keinen erkennbaren Einfluss auf die Vernetzung. Dieses anfängliche Netz beinhaltet 87.000 Zellen und ist damit gut innerhalb des Zielbereichs. Auch die Feinheit der Fillets wird als vorerst ausreichend angesehen. Wie groß der Unterschied zu einem infinit kleinen Netz ist kann erst nach den ersten Simulationen bewertet werden. Ziele, die aber mit diesem Netz noch nicht erreicht wurden, sind die bezüglich der Unterteilung der Scheibe und der Oberseite der beiden Schaufeln.

Dafür wurden einzelnen Flächen von den globalen Einstellungen abweichende Standardelementgrößen zugewiesen. Damit konnte das in Abbildung 2.7 rechts zu sehende Gitternetz erzeugt werden. Mit 130.000 Elementen ist es noch innerhalb der Parameter und, wie zu sehen ist, ist auch die nach Kapitel 2.2.1 angestrebte Unterteilung der Schaufel in drei Fragmente der Höhe nach erreicht. Selbiges gilt für die Unterteilung in zwei Elemente in der Breite für sowohl die Hauptschaufel als auch die Splitterschaufel. Da dieses Netz für die nächsten Schritte ausreicht und weitere Verfeinerungen sich erst mit mehr Wissen anbieten, wurde die Erstellung dieses Netzes automatisiert.

Da im Fidelity Workflow das Projekt erst gespeichert werden muss, bevor die Vernetzung gestartet werden kann, wurde das Projekt über einen Befehl der Python-API an einem relativen Pfad gesichert. Die Erstellung wurde in eine Methode ausgelagert, die nach dem Domänensetup ansetzt. Die getroffenen Einstellungen im Hexpress Mesher sehen eine Startzellengröße von einem Millimeter vor. Die weitere Unterteilung für Krümmungen wurde für die gesamte Geometrie aktiviert. Die maximale Unterteilung wurde auf vier Elemente gesetzt. Die Mindestgröße für die Zellen der zusätzlichen Unterteilung steht auf einem Zehntel. Die Zielzellgröße der äußeren Grenze des Objekts, in der Abbildung 2.7 gelb dargestellt, wurde auf sieben Zehntel eines Millimeters gesetzt, die der Vorderseite der Blätter, in hellblau und rot, auf drei Zehntel. Damit sich die Zellverteilung durch die gesamte Schaufel hindurch fortsetzt, wurden auch die Zellgrößen der Saug- und Druckflächen der beiden Schaufeln geändert. Sie wurde auf einen halben Millimeter gesetzt. Das Ansprechen der Flächen erfolgt über den Namen der obersten Gruppe der Geometrie und der Gruppe darunter, welche die Fläche enthält. Einzelne Flächen können nicht direkt angesprochen werden. Durch eine Umstellung der Vernetzung auf Hexaeder-Dominant konnte das Netz mit 130.000 Elementen generiert werden. Da dieses Netz Pyramiden enthielt, wurde es mit dem Hinzukommen der Calkulix-Beschränkung, die am Ende von Kapitel 2.2.1 genannt wurde, verworfen. Für das neue und diesmal finale Netz wurde deshalb wieder das Tetraeder-Netz verwendet. Die Grundgröße der Zellen wurde auf zwei Millimeter verdoppelt. Die Kurvenverfeinerung blieb gleich. Die Außenflächen wurden alle mit Ausnahme der roten Außenfläche aus Abbildung 2.8 mit einem Millimeter vernetzt. Damit das Ziel von drei Elementen in der Höhe der Grundscheibe und zweien in der Breite der Schaufeln erreicht werden konnte, wurde sie auf auf sieben Zehnteln eines Millimeters belassen. Mit diesen Einstellungen würde ein Netz generiert, das die Zielbedingungen nicht erfüllt. Deshalb wurde im Gegensatz zu den vorherigen Netzen die Bufferschicht-Option aktiviert. Sie verändert das Verhalten des Vernetzers wodurch eine Zwischenschicht außen an den Flächen erzeugt wird, während innerhalb der Geometrie anders vernetzt wird. Dadurch kommen im Anwendungsfall zu den Tetraedern Prismen hinzu. Die Option ist gut in der rechten Hälfte von Abbildung 2.8 zu erkennen. Die Prismen bilden eine Schicht zwischen dem Rand der Geometrie und dem Tetraeder-Netz im Inneren. Außerdem kann eine weitere Option aktiviert werden, welche davon abhängt. Sie

erlaubt es, die Zahl der Elemente anisotropisch, also von der Richtung abhängig, direkt zu beeinflussen. In diesem Fall liegt die Richtung rechtwinklig zum Verlauf der langen Seite des Objektes. Durch die Option kann bestimmt werden, wie viele Elemente mindestens an Engstellen der Geometrie erzeugt werden sollen. Für die einmal sattgrüne und einmal violette Oberkante der Schaufelblätter wurde dieser Wert auf drei gesetzt. Außerdem ist noch zu erwähnen, dass standardmäßig in Fidelity eine viereckdominierte Zwischenschicht generiert wird. Diese Option wurde nach der Einstellung auf das Tetraeder-Netz und der Aktivierung der Zwischenschicht vor dem Start der Vernetzung deaktiviert. Wird vor den beiden Schritten versucht die Einstellung zu deaktivieren, schlägt das fehl, da die Option noch nicht verfügbar ist. Vorgreifend auf die Exportoptionen wurde eine Option aktiviert,

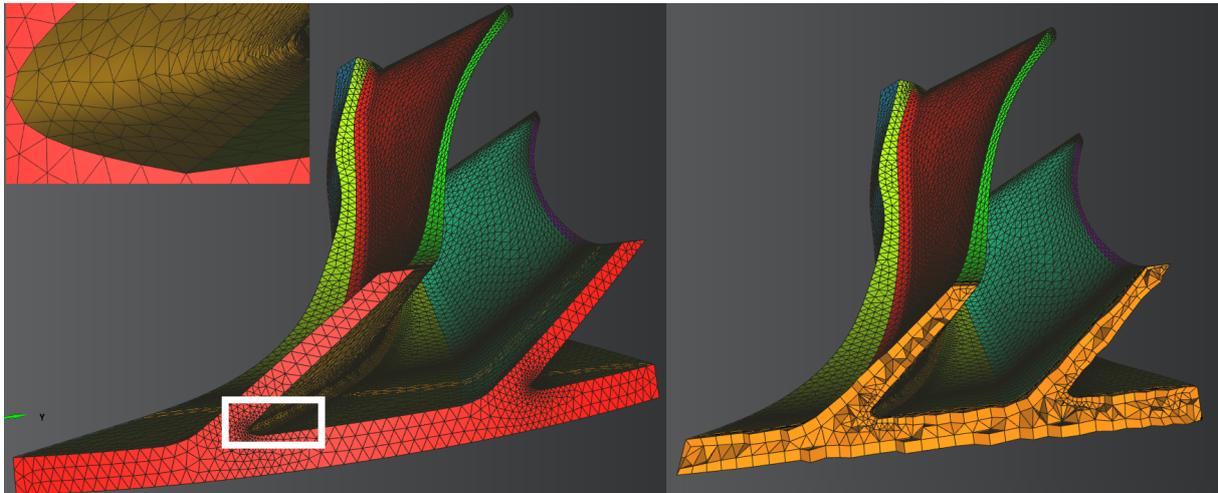


Abbildung 2.8: Finales Netz mit Zoomausschnitt auf das Fillet. Rechts geöffnet, um die Netzelemente zu zeigen

die das Netz für besagtes Exportformat optimieren soll. Danach wurde die Vernetzung gestartet. Das Programm wartet darauf, dass die Vernetzung abgeschlossen ist. Dann wird das Netz exportiert.

2.2.5 Export des Netzes mittels des OpenFoam Dateiformats

Fidelity bietet verschiedene Exportformate an. Darunter sind Dateitypen speziell für Hexpress Netze, Fluent Dateien, Tau Dateien, Wavefront Obj Netzdateien und auch das OpenFoam Format. Es gibt noch weitere Exportformate. Einige der Formate sind nur für zweidimensionale Netze geeignet und können entsprechend direkt ausgeschlossen werden. Ein weiterer betrachteter Punkt ist die Lesbarkeit. Einige der Formate benötigen externe Programme oder Bibliotheken, um sie lesen zu können. Um die Nutzung solcher Programme gering zu halten, da dies außerdem mit dem wegen Fidelity genutzten Interpreter zu Problemen hätte führen können, wurden auch diese ausgeschlossen. Übrig blieb OpenFOAM und Fluent. Fluent ist dabei dasselbe Format, welches auch das bisher

verwendete Gmsh nutzt. Das bisherige Programm zum Verarbeiten des Netzes sieht allerdings ein festes Schema vor, welches hier nicht gegeben war. Entsprechend bringt das Fluent Format keine Vorteile. Der OpenFOAM Export ist auf sechs Dateien aufgeteilt und ermöglicht eine genaue Zuordnung der Punkte zu den Zellen und Flächen. Es kann sowohl als Binärdatei als auch als ASCII Datei gesichert werden. Dieses Format wurde wegen diesen Eigenschaften ASCII-formatiert gewählt.

Die erste Datei des Foam Exports ist eine leere Datei mit dem Dateityp Foam. Im selben Ordner wie diese Datei findet sich der *constant* und darin der *polyMesh* Ordner. Darin sind die eigentlichen Dateien mit den Koordinaten der Punkte und Informationen zu deren Anordnung. Jede Datei startet dabei mit zwei Headern. Der erste enthält Informationen zu OpenFOAM. Darunter die Version. Der andere Header trifft Aussagen über die Datei direkt. Dazu gehört die Unterscheidung zwischen ASCII und Binary und die Information über den Datentyp in welchem die Daten vorliegen. Da hier von einem festen Schema ausgegangen wird, in dem die Dateien vorliegen, werden die Header beim eigentlichen Auslesen ignoriert. Die erste Datei enthält die Liste der Punkte. Da der Export dreidimensional ist, hat jeder Punkt und damit jede Zeile drei Gleitkommazahlwerte. Vor den Werten steht noch die Anzahl der Punkte. Die Werte werden über Python ausgelesen und in ein Gleitkomma-Array umgewandelt. Diese Punkte werden direkt in der mit der nächsten Datei verbunden. Dieses File, *faces* genannt, enthält die Punktindizes der einzelnen Flächen, aus denen die Zellen des Netzes bestehen. Die Datei ist ähnlich aufgebaut. Erneut folgt eine Liste von Werten nach den Headern und der Gesamtzahl der Flächen. Anders als bei der Punktdatei steht hier am Anfang jeder Zeile noch die Zahl der Punkte pro Gesicht, was es auch ohne die Zeilentrennung möglich machen würde, die Datei auszulesen. In einem reinen Hexaeder Netz wären alle Flächen Vierecke und jeder Eintrag hätte entsprechend vier Werte. Bei einem Tetraeder Netz wären es drei Punkte pro Dreieck. Im Anwendungsfall wurde zuerst ein Hexa-dominantes Netz betrachtet, bevor der Wechsel zu einem Netz aus Tetraedern und Prismen erfolgte. Entsprechend enthält die *faces*-Datei Punktlisten für Drei- und Vierecke.

Bei den Werten handelt es sich um positive Ganzzahlwerte. Diese werden als Indizes verwendet, welche auf die ausgelesenen Punkte verweisen. Um das Programm zu beschleunigen, wurden diese Indizes zuerst nicht komplett ausgelesen und in eine Liste geschrieben, sondern direkt nach dem Auslesen einer Zeile in die entsprechenden Punkte umgewandelt. Nachdem bei einem späteren Schritt die indexbasierte Zuordnung gebraucht wurde, wurde davon abgesehen.

Die Seiten werden über die *Owner* und *Neighbour*-Datei den Zellen zugewiesen. In der ersten Datei befindet sich, wieder nach der Zahl mit der Länge, eine bei Null startende Liste von Zahlen. In diesem Fall kommen viele Zahlen mehrmals vor. Auch wenn es Anfangs danach aussieht, sind die Zahlen, die die Zelle bezeichnen, nicht zwingend gruppiert. Die Position der Zahl steht für eine Seite des Netzelements, wobei der Index der Zahl die

Fläche anspricht. Das heißt, dass die erste Null die erste Fläche anspricht und diese dann zur nullten Zelle gehört. Entsprechend könnte jede Fläche aus der *Face*-Datei nur zu einer Zelle gehören. Deshalb wird die Datei durch die Nachbardatei ergänzt. Die Zahlen in ihr sind scheinbar ungeordnet. Diese Zahlen beziehen sich auf Zellen. Die Stelle an der die Zahl steht, ihr Index, steht für den Index der Fläche. Das heißt, dass die selbe Zahl teils mehrmals vorkommen kann. Im Umkehrschluss heißt es, dass jede Fläche zu genau zwei Zellen gehört. Entsprechend können die verbleibenden Seiten der Zellen, die nicht in der *Owner*-Datei aufgeführt waren, den Elementen zugeordnet werden. Das geht soweit, dass nicht alle Zellen in der *Owner*-Datei aufgeführt sind. Das ist der Fall, wenn die Seiten, die zur ausgelassenen Zelle gehören, bereits alle bei anderen Zellen aufgeführt wurden. In diesem Fall wird die Zelle komplett durch die Nachbar-Datei definiert. Die letzte Datei ist die *Boundary*-Datei. Bislang wurden die Elemente nur an sich gebildet. Das gesamte Netz wurde eingelesen und könnte entsprechend dargestellt werden. Was für die Umsetzung aber auch noch benötigt wird ist die Information darüber, welche der Zellen Grenzzellen nach außen sind. Die Information darüber enthält besagte letzte Datei. In ihr gibt es einen Eintrag für jede Fläche. Von den vier Einträgen für jede Geometriebegrenzung sind zwei wichtig. Sie beziehen sich auf die *Faces*-Datei. Der erste Eintrag nennt einen Startindex, der zweite eine Länge. Dadurch werden Teile der Flächen zu den Außenwänden zugeordnet. Entsprechend können Subsets der Liste gebildet werden. Damit sind alle Informationen, die für die weitere Nutzung des Netzes nötig sind, erkannt. Zum Auslesen der Dateien wurde je eine Methode erstellt. Am Anfang wurden die nicht benötigten Teile der Datei, also der Header, aus der eingelesenen Kopie entfernt. Wenn die Zeilenaufteilung für die Daten wichtig ist, wird der Inhalt zuerst in die Zeilen aufgeteilt. Danach werden die Zeilen, bzw. jetzt Arrayelemente, mit einem Ausdruck verglichen, der in den Dateien das Ende des Headers kennzeichnet. Relativ dazu konnte die Länge der Listen in den Dateien gefunden werden, um auch das Ende des Files abzuschneiden. Was übrig blieb, war eine gleichförmige Liste der Punkt- bzw. Gesichtselemente, welche entsprechend leicht bearbeitbar war. Später wurde die Suche durch einen festen Index ersetzt, vor dem die Zeilen abgeschnitten wurden. Dabei wird davon ausgegangen, dass der Header immer die selbe Länge hat. Die verbleibenden Zahlen wurden dann zeilenweise aufgeteilt und schließlich von den Stringwerten, in denen sie ausgelesen wurden, in Ganzzahlwerte transformiert. Anders wurde bei der *Owner*- und *Faces*-Datei vorgegangen. Hier kam es nicht auf die Zeilen an. Deshalb wurden hier nach dem ersten Vorkommen eines regulären Ausdrucks gesucht, aus dem die Länge direkt ausgelesen wurde. Auf die selbe Art wurde das Ende identifiziert, indem die Suche von hinten gestartet wurde. Da die Dateien die Punkte und Zusatzinformationen zu über 100.000 Zellen enthalten, sind sie entsprechend groß. Deshalb wurde versucht die Menge der Durchläufe durch diese Listen zu reduzieren. Die einzelnen Methoden, um die Punkte, Faces und übergeordneten Seiten der Geometrie auszulesen, werden dann von einem anderen Script genutzt, um die

Inputdateien für Calculix zu schreiben. Ursprünglich wurden mehr der Indizes direkt zu Punkten umgewandelt. Gemäß den Anforderungen für die folgenden Dateitypen wurden stattdessen die Punktindizes bei den Elementseiten, Elementen und Seiten zurückgegeben.

2.2.6 Schreiben der Calculix-Inputdateien

Für die Anbindung an den Rest der Prozesskette, müssen die Inputdateien für Calculix geschrieben werden. Dafür werden mittels Python zwei ASCII-Dateien erzeugt. Die erste Datei, welche die Endung *inp* hat, enthält Grundeinstellungen für Calculix. Dazu gehören die Materialeigenschaften zu Elastizität und Dichte, aber letztlich auch Informationen zur Geometrie. Auf die Beschreibung des Modells erfolgen verschiedene Segmente, welche durch *Step*-Header und -Trailer voneinander abgegrenzt sind. In den Segmenten sind Lasten und Eigenschaften eingetragen, welche bei der folgenden FEM wichtig sind[16]. Darauf folgen noch weitere Karten mit anderen Headern, welche die Form der Ausgangsdateien definieren. Diese Werte werden alle direkt mit dem Programm in die Datei geschrieben. Der Programmteil existierte dabei bereits vorher und wurde weiterverwendet. Der Teil, der an den Rest des Programms angepasst wurde, enthält die Punkte und Elemente. Er kommt noch vor der besagten Definition und beginnt wieder mit einem Header. An sich kann dieser Abschnitt direkt in die Datei geschrieben werden. Alternativ können sie auch in einem anderen File aufgeführt werden, da Include-Statements zur Übernahme von Inhalten aus anderen ASCII-Dateien unterstützt werden. Das wurde in diesem Fall gemacht. Die Punkte- und Elementdefinitionen stehen in einer *mdl*-Datei. Bei den Punkten zeigt er nur den Start des Punkteabschnitts an. Bei den Elementen dagegen wird auch der Elementtyp angegeben. Dadurch wird bestimmt, wie viele Punkte pro Element erwartet werden, und wie diese angeordnet werden. Die Bezeichnung für ein Tetraeder aus vier Punkten ist zum Beispiel *C3D4*. Gibt es mehrere Elementarten, dann gibt es auch mehrere dieser Abschnitte. Zusätzlich wird die Elementkategorie noch einem logischen Set zugeordnet. Das wird über die *ELSET*-Option gemacht. Bei den Elementindizes ist es für das Dateiformat wichtig, dass ihre Reihenfolge monoton steigt. Der dritte Teil, der nicht direkt eingetragen werden kann, ist ein Referenzpunkt. Dieser Punkt wird bei der Simulation als Fixpunkt verwendet. Er soll möglichst nahe an der späteren Drehachse des Rotors sein und zusätzlich mittig von der erstellten Geometrie. Dazu kommen noch Gruppen von Punkten, welche in diesem Fall die Rotationsflächen darstellen. Das sind die Flächen, welche bei der richtigen Geometrie eigentlich im Inneren liegen.

Für die Überführung der ausgelesenen Punkte in die *mdl*-Datei existierten bereits Methoden, welche aber an die Struktur der aus den OpenFoam-Dateien ausgelesenen Werte angepasst werden mussten. Mathematische Grundumformungen konnten beibehalten werden. Die Punkte, oder Nodes wie sie in der Datei genannt werden, erhalten können beinahe

direkt eingetragen werden. Für sie werden die zugehörigen Indizes gebildet. Sie kommen direkt an den Anfang der Zeile. Danach werden mit Kommas abgetrennt die Koordinaten angehängt. Dann erfolgt ein Zeilenumbruch. Nachdem alle Punkte eingetragen wurden, wird der Elementheader mit Elementtyp und Gruppenzuordnung eingefügt. Bei einem umfassenderen Programm würden alle von Calculix unterstützten Formate behandelt. Da die Methode auf einen bestimmten Anwendungsfall zugeschnitten ist, werden nur besagte $C3D4$ -Elemente behandelt. Nach der Netzänderung müssten die Prisma-Elemente gesondert abgehandelt werden und in eine neue Kategorie geschrieben werden. Nach einem Problem, welches später behandelt wird, wurde das nicht mehr gemacht. Der Input für das Schreiben der Elemente der aus dem OpenFoam-Leser kommt, besteht aus mehreren Listen von Punktindizes. Im Fall der Tetraeder handelt es sich um vier davon. Jede Liste enthält die Punkte für eine der vier Seiten des Objektes. Da das Element nur von vier Knoten gebildet wird, sind die Punktindizes mehrfach vorhanden. Die Listen werden deshalb aneinandergehängt und in ein Set einer Integer-map umgewandelt, um keine Punktindizes doppelt zu schreiben. Davon werden sie wieder in Liste, bzw. letztendlich in ein Numpy-Array umgewandelt. In dieser Form werden sie aufsteigend sortiert, um die richtige Reihenfolge zu gewährleisten. Dann werden sie, auch wieder mit ihrem bei Eins startenden Index versehen, in der der selben Form wie vorher die Punkte in die Datei eingetragen. Pro Zeile und Element ergeben sich bei Tetraeder damit mit dem Index insgesamt fünf Einträge.

Der nächste Schritt war es, die Punkte der Rotationsflächen, das sind die in Abbildung 2.5 am besten zu erkennenden Flächen links am linken Rand in violett und auf der rechten Bildhälfte unten in hellblau, in die Datei als spezielle Punktwolke einzutragen. Die Rotationsflächen sind die Sektorgrenzen, also die Flächen welche sich jeweils von den verschiedenen Teilen, von denen eines erstellt wurde, wenn das Modell vervollständigt würde, berühren würden. Da die *Boundary*-Datei von OpenFoam bereits auf die Elementseiten verweist, welche die Seiten bilden, lagen die Punktindizes nach Seiten aus dem Auslese-Programm direkt vor. Wie schon davor bei den Elementen lagen die Punktindizes entsprechend mehrmals vor und wurden auf die selbe Art bearbeitet, sodass jeder Punkt nur einmal aufgeführt wurde. Die Punktewolken wurden mit dem *Nset*-Header als Gruppe definiert. Das wurde für beide Rotationsflächen gemacht, wodurch sich zwei dieser Gruppen ergaben. In die Datei wurde die Punktliste dann mit einem Punkt pro Zeile eingetragen. Beim letzten einzutragenden Element handelt es sich um den Fixpunkt. Auch er ist als Gruppe eingetragen, auch wenn diese nur einen Punkt beinhaltet. Da aufgrund der sich während der Optimierung ändernden Geometrie und damit Punktezahl nicht von einer festen Reihenfolge der Punkte oder festen Koordinaten ausgegangen werden kann, musste für das Finden des Punktes ein flexibler Ansatz gefunden werden. Eine Möglichkeit wäre es gewesen, die Punktewolken verschiedener Flächen miteinander zu vergleichen. Diese Wolken lagen, wie anhand der Rotationsflächen sichtbar ist, vor. Über einen Vergleich

der Listen hätten die Punkte identifiziert werden können, welche auf der Kante zwischen zwei Flächen liegen. Durch das hinzunehmen zweier weiterer Flächen hätten auch die Kanten identifiziert werden können, da nur die Punkte auf der Kante, bzw. Ecke, in allen Punktlisten vorhanden sind. An sich hätte ein Punkt, der nicht gleichzeitig Ecke der Kante ist, für Calculix gereicht. Dieses Vorgehen hätte aber mehrere Vergleiche mit langen Listen gebraucht. Außerdem hätte er nicht den optimalen Punkt in der Mitte geliefert. Das Verfahren hätte durch die Berechnung des Schwerpunktes der Punkte auf der Kante verbessert werden können. Der Punkt am nächsten zu diesem Schwerpunkt wäre dann der gesuchte mittlere Punkt. Obwohl dieser Ansatz zum Ziel geführt hätte, wurden stattdessen die Punkte der gemäß der Ansicht aus Abbildung 2.8 untersten Ebene betrachtet. Sie ist verhältnismäßig klein. Entsprechend wenige Punkte müssen beachtet werden. Von ihnen wurden dann die tatsächlichen Koordinaten genommen und in x-y-Theta Koordinaten um die X-Achse der Geometrie umgerechnet. X ist dabei konstant, da sich die gesamte Ebene auf der selben Höhe befindet. Y entspricht für die Punkte also der Entfernung zur Mitte des Systems. Da die Innenseite der Geometrie in der Realität um eine Welle gelegt ist, ist sie rund. Entsprechend können aus den Punkten der Fläche die Punkte der Innenkante über ihre y-Koordinate gefunden werden. Ihr Wert muss minimal sein. Dabei werden zur Sicherheit, um Abweichungen durch das Netz zu kompensieren, Punkte genommen, die eine y-Koordinate haben, die minimal größer sein kann, als das kleinste gefundene Y. Dieser Fehlerbereich muss klein genug sein, dass nur Punkte der Innenkante gewählt werden können. Als Orientierung wird die Standardgröße der Netzelemente hergenommen. Die Punkteauswahl kann jetzt nach dem Winkel Theta, der angibt wo auf dem Umkreis um die X-Achse sich der Punkt befindet, sortiert werden. Der mittlere Punkt aus der Punktliste ist dann der gesuchte Punkt. Damit dieser wieder auf den Punktindex zurückgeführt werden kann ohne diesen suchen zu müssen, wird eine Kopie der Indexliste parallel zu den Punkten umsortiert. Der Punkt wird abschließend noch als eigene Gruppe der Datei hinzugefügt.

Die Funktionsweise der angefertigten Datei wurde dann mit dem Programm PrePoMax getestet. Dabei ergab sich, dass die Geometrie mit Löchern dargestellt wird. Um die Ursache des Problems zu finden wurde ein Minimalset aus den Elementen ausgewählt. Auch bei einem Herunterbrechen auf drei Elemente mit Punkten der Geometrie wurden weiterhin die Elemente nicht korrekt angezeigt. Dieses Minimalbeispiel ist in 2.9 dargestellt. Zu sehen sind die Kanten der Netzelemente. Mehrere Seiten der Geometrie sind durchsichtig. Das Einfügen der Elemente in eine andere funktionierende Datei ergab lediglich, dass nicht aufsteigend sortierte Punktindizes bei den Elementen für ein ähnliches Problem sorgen können. Die Flächen fehlten weiterhin. Ein Angleichen der Genauigkeit und anderer Parameter hatte keinen Erfolg. Zuletzt wurde, um Fehler im Anzeigeprogramm auszuschließen, die Datei direkt an Calculix übergeben. Das Programm stürzte ohne verwertbare Fehlermeldung ab. Sowohl Grund als auch Lösung des Problems stehen aus. Deshalb wurde

der Code auch nach Umstellung des Netzes auf die Finale Version mit Prismen nicht um besagte neue Elemente ergänzt. Im Anschluss an den Export des Netzes für Calculix

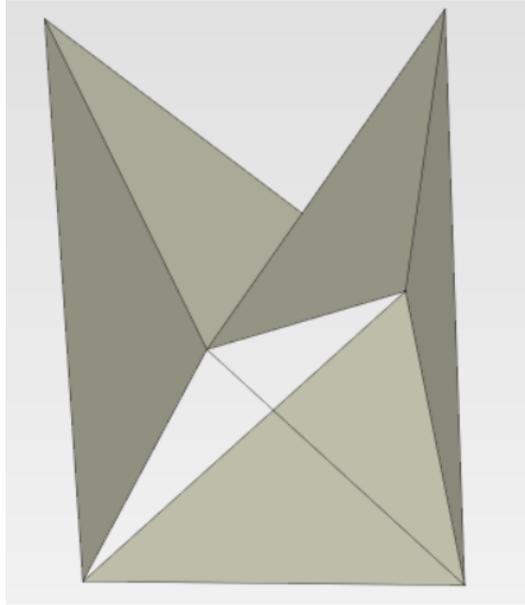


Abbildung 2.9: Darstellung der Fehlerhaften Netzelemente in PrePoMax

würde die Projektdatei über den bekannten relativen Pfad wieder gelöscht, um Speicher freizugeben. Direkt über das Python Skript geht das nicht, da das Projekt als geöffnet angesehen wird. Wenn das Python Programm, das in Fidelity aufgerufen wurde, um die Geometrie und das Netz zu erzeugen, aber endet, kann es die Datei nicht mehr löschen. Um das zu umgehen wird ein Subprozess gestartet, welcher den Ordner über Systemkommandos löscht. Diese Befehle sind betriebssystemspezifisch, weshalb das Betriebssystem vorher abgefragt wird. Entsprechend werden unterschiedliche Befehle gegeben.

3 Kosten bei der Fertigung und im Betrieb

Die zweite Sektion der Themenstellung betrachtet den Preis der Fertigung von Verdichtern. Dabei wird gemäß des Themas nur der Filletradius an sich betrachtet. Das Ziel ist es entsprechend, ein Modell dafür zu erstellen. Ein komplettes Modell, welches alle Umstände des tatsächlichen Vorgangs betrachtet, kann hier aber aufgrund der Komplexität nicht gebildet werden. Die Betrachtung, welche Bestandteile des Preises wie betrachtet werden sollen und können, ist die Basis des Kostenmodells für die Herstellung.

Der Fertigung gegenüber steht der Betrieb, bei dem auch Kosten anfallen. Aus der Optimierung gingen Wirkungsgrade für verschiedene Filletradien hervor, was einen Vergleich ermöglicht. Auch dieser Bereich muss betrachtet werden.

3.1 Der Einfluss des Wirkungsgrades auf die Betriebskosten

Eine HTWP in Betrieb verbraucht Ressourcen. Dazu gehören zum Beispiel Wartungskosten und auch Kosten für Betriebsmittel wie Öl. Daneben benötigt die Anlage während des Betriebs permanent Strom. Ein Teil davon wird für die Steuerung der Sensoren und Bauteile verwendet. Der wichtigere und hier betrachtete Anteil aber fällt im Rahmen der Verdichtung an. Die Menge ist von der Bauweise und den Anforderungen des Systems abhängig. Die Anforderungen wurden bereits in Kapitel 2.2.1 vorgestellt. Daraus ergibt sich, dass der Verdichter einen bestimmten Druckanstieg bewirken muss. Dafür wird Arbeit verrichtet und Energie verbraucht. Im Optimalfall wäre die Beziehung zwischen Druckanstieg und Energiebedarf linear. Tatsächlich treten zu viele Verluste dafür auf. Der Verlauf von Preis zu Wirkungsgrad wird in [19] anhand eines Fallbeispiels betrachtet. Für die Simulation wird dabei eine Wärmepumpe wie sie in Abbildung 2.2 dargestellt ist verwendet. Der Wirkungsgrad wurde dabei mit Tespy berechnet. Die Ergebnisse sind abhängig von den Werten des Systems. Für die Darstellung wurde eine Temperaturänderung von 120 auf 150 Grad Celsius eingestellt. Der Druckerhalt bei Verdampfer und Kondensator lag jeweils bei 98%. Bei einer Wärmeleistung von einem Megawatt ergaben sich die Ergebnisse, welche in Abbildung 3.1 dargestellt sind. Dabei sind die Wirkungsgrade

von 70% bis 90% mit Schritten von einem Prozent auf der X-Achse angetragen. Die Kosten entsprechend zu der jeweiligen Prozentzahl finden sich auf der Y-Achse. Um zu zeigen, dass der Preisverlauf nicht linear ist wurden eine Gerade zwischen den ersten und letzten berechneten Preispunkt gezeichnet. Die Einzelpunkte sind die eigentlichen Werte. Im Pro-

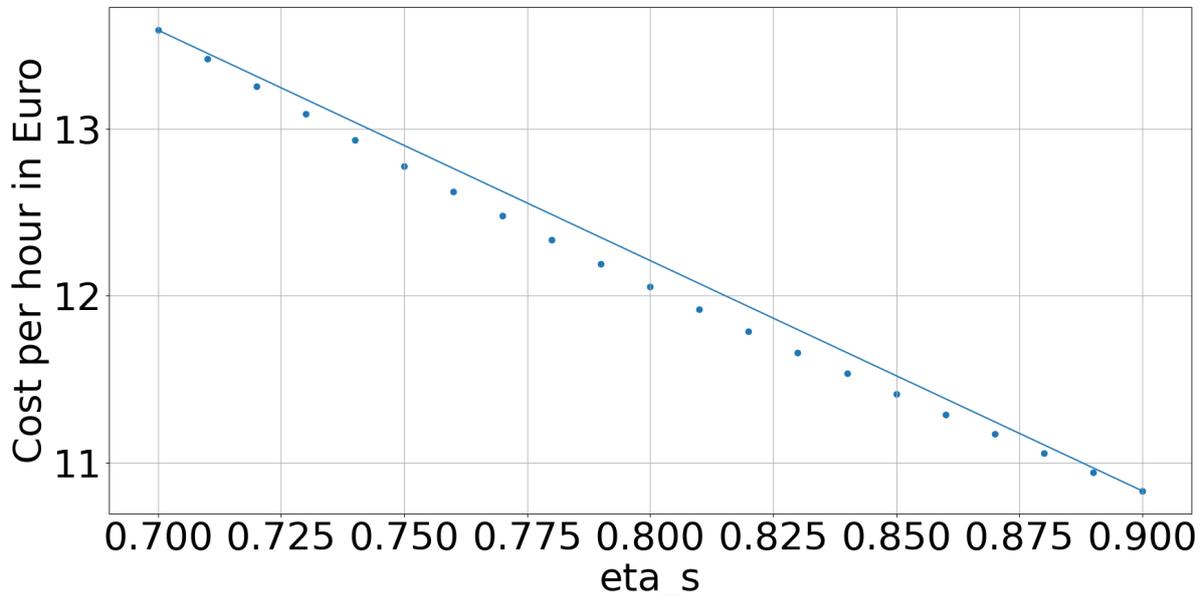


Abbildung 3.1: Darstellung des Kostenverlaufs nach Wirkungsgrad für eine Beispielwärmepumpe aus [19]

ogramm wird die Anlage definiert und vom Tespy-internen Löser berechnet. Als Eingabe erhält das Programm eine Liste von Wirkungsgraden für den Kompressor. Für jeden Wirkungsgrad wird der Löser einmal gestartet. Zurückgegeben werden verschiedene Werte welche Aussagen über die Eigenschaften des Systems treffen. Darunter befindet sich die vom Kompressor benötigte Energie. Diese wird mit einem angenommen Strompreis, der im Programm auf 45.16 Cent pro Kilowattstunde gesetzt wurde, verrechnet um den Teil des Preises pro Betriebsstunde zu erhalten, der durch den Kompressor anfällt. Dadurch gibt es pro Wirkungsgrad einen Stundenpreis. Der Wirkungsgrad wirkt sich also auf den Stromverbrauch aus und durch die Stromkosten auf die Betriebskosten[19]. Der Preis pro kWh hängt vom jeweils aktuellen Strompreis und etwaigen Verträgen ab. Was für die Betriebskosten bei der Auslegung des Verdichters betrachtet werden muss, ist der Mehrpreis zwischen den verschiedenen erreichten Wirkungsgraden und die entsprechend für die Verbesserung anfallenden Kosten.

3.2 Kosten beim Fräsvorgang

3.2.1 Grundlagen des Fräsens und der Modellerstellung mit Einschränkungen

Verdichterrotoren und Statoren können auf verschiedenen Wegen gefertigt werden. Nicht für alle Anwendungen bietet sich jede Fertigungsweise an. Von Weitem betrachtet wird zwischen subtraktiven und additiven Verfahren unterschieden, wobei es auch Prozesse gibt, welche beide nutzen[20]. Ein Beispiel dafür ist das Zusammenschweißen der Scheibe und der Klingen eines Rotors mit anschließender Abtragung überflüssigen Materials mittels Fräsen[20]. Neben dem Schweißen und Fräsen gibt es noch andere Möglichkeiten wie zum Beispiel das Gießen des Impellers, was für größere Werkstücke ausreicht, oder die Elektroerosive Abtragung, bei der Hitze durch Elektronenentladung erzeugt wird, um kleine Mengen des Materials abzutragen. Das Verfahren benötigt keinen Kontakt zum Werkstück und ist für schwer bearbeitbare Materialien geeignet. Das Verfahren hat aber eine geringe Abtragsrate[20]. Weitere Vorgehensweisen bestehen aus dem direkten 3D-Drucken der Geometrie oder dem Sintern. Die Vorgehensweisen sind vielfältig und oft, wenn auch bei gleichbleibenden Grundprinzip, im Wandel. Für verschiedene Feinheiten bieten sich andere Anwendungen an. Die Elektroerosive Abtragung zum Beispiel kann höhere Feinheiten als aktuelle Fräsvorgänge erreichen. Fräsen ist dafür aber meist schneller. Die Wärme durch das Anschweißen von Bauteilen währenddessen sorgt für Materialeigenspannungen durch die stellenweise Erhitzung, was im Extremfall zu Maschinenschäden und sogar zur Gefährdung von Menschen führen kann, wenn die Spannung nicht der Anwendung entsprechend gering gehalten wird[21]. Es kommt also stark auf den Anwendungsfall an. Auf Grund der in diesem Fall hohen Anforderungen an das Material und die Rauheit, welche auf mindestens N6 festgesetzt wurde, wurde das Fräsen als Fertigungsmethode gewählt.

Eine weitere Unterscheidung wird bei der Achsigkeit und damit den Bewegungsmöglichkeiten des Fertigungsarms getroffen. Beim Fräsen sind häufig drei bis fünf Achsen gebräuchlich. Die ersten drei Achsen beziehen sich dabei normal auf die Freiheitsgrade des Werkzeugs. Die verbleibenden zwei können durch Drehungen des Werkstücks dazukommen. Fünf Achsen sorgen für mehr Beweglichkeit und eignen sich besser für feine Abstufungen und komplexere Geometrien. Im Umkehrschluss gibt es allerdings auch mehr zu programmieren und die Nutzung ist teurer. Da es sich bei der betrachteten Anwendung um ein relativ kleines Werkstück mit komplexer Geometrie handelt, liegt der Fokus auf fünfachsigem Maschinen.

Die Preise für die Nutzung sind komplett vom Hersteller abhängig. Zur Vereinfachung wird hier ein Stundenpreis von 80 Euro angenommen[22]. Die tatsächliche Fertigungszeit, in der das Werkstück bearbeitet wird, setzt sich zusammen aus der Zeit, um Werkzeuge zu

wechseln oder zu ersetzen, etwaigen Wartungszeiten, Wegzeiten und der für die tatsächliche Materialabtragung nötige Zeit.

Da keine professionelle Software für die Berechnung der Zeit und damit der Kosten zur Verfügung steht, muss ein vereinfachtes Modell erstellt werden. Für ein tatsächliches Modell müssten die eben genannten Parameter alle einberechnet werden. Das würde aber den Rahmen weit sprengen und würde letztendlich dem Programmieren eines kompletten Computer-aided-manufacturing (CAM)-Programms gleichkommen. Selbst diese können aber keine Aussage über die Werkzeugabnutzung treffen. Darüber, wie viele Fräsköpfe pro Frässtück kaputtgehen, wird deshalb allgemein keine Aussage getroffen[22]. Entsprechend wird der Faktor auch nicht direkt in das Modell übernommen. Dementsprechend können auch die Wechselzeiten nur Unzureichend in er Näherung berücksichtigt werden, da der Großteil davon voraussichtlich durch Werkzeugverschleiß und nicht durch den Wechsel zwischen Fräsköpfen mit verschiedenen Parametern verursacht wird. Unabhängig davon wird der Wechsel zwischen den Köpfen als Zeitaufwand von fünf Minuten angenommen [22] und spielt damit keine große Rolle. Was ebenfalls im vereinfachten Modell nicht betrachtet wird sind die Wegzeiten, da dafür komplexere Gleichungssysteme für jeden Freiheitsgrad des Werkzeugkopfes aufgestellt werden müssen.

Da auch für den Wirkungsgrad nur der Filletradius betrachtet wird, wird auch hier nur der Radius betrachtet, Der Rest der Geometrie wird außen vorgelassen. Weitere Einschränkungen ergeben sich durch die angenommene Maschine. Die Randangaben dafür ergaben sich aus [22]. Demnach haben die Fräsköpfe eine permanent gleichbleibende Geschwindigkeit. Das gilt auch, wenn bei einer Bahn aktuell kein Kontakt zwischen Werkstück und Werkzeug besteht. Es wird also davon ausgegangen, dass ein Programm strikt abgefahren wird. Diese "Leerphasen" können durch die sich wandelnde Geometrie im Lauf einer Bahn auftreten. Darunter zählen zum Beispiel Unterschiede im Winkel des gesamten Blattes, wie sie zwischen Saug- und Druckseite bestehen. Damit die Berechnung ausreichend funktioniert erfolgt wieder eine Vereinfachung. Es wird von einer Standardausgangsgeometrie ausgegangen, die näherungsweise den Anforderungen entsprechen sollte. Daraus ergibt sich aber auch die Vereinfachung, dass eine konstante Geschwindigkeit für die Bahnen angenommen werden kann und Bahnen immer komplett abgefahren werden. Die gesamte Länge des Fillets kann also für die Strecke hergenommen werden.

Was für die Fertigungszeit und damit die Kosten direkt betrachtet werden kann, ist aber die Anzahl der Bahnen. Dabei ist die Startgeometrie wichtig, um zu wissen wie viel Material abgetragen werden muss. Zusätzlich zu dieser Materialabtragung muss die Endfläche mit einem feineren Fräskopf auf die erforderliche Rauheitsstufe gebracht werden. Direkt diesen Fräskopf zu nehmen würde mehr Zeit in Anspruch nehmen, da dessen Material Abtragsrate geringer ist. Entsprechend wird ein größerer Kopf für einen Großteil des Materials eingesetzt.

Für die Abtragsrate sind hauptsächlich drei Parameter des Bohrkopfes wichtig. Sie

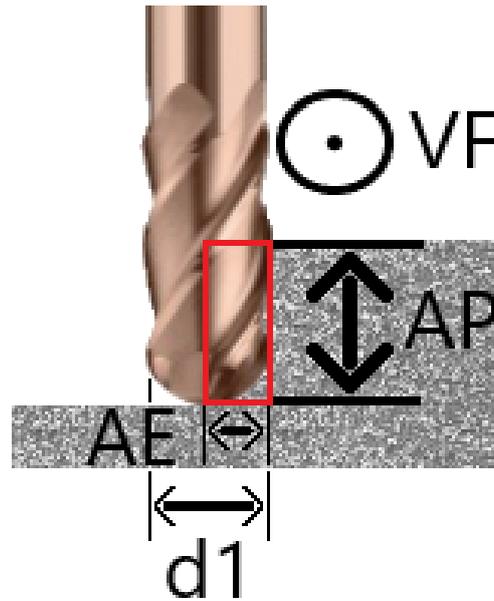


Abbildung 3.2: Darstellung der hier wichtigen Parameter eines Fräskopfes. Der Fräskopf selbst stammt aus [23]. Es handelt sich um die Darstellung des Kopfes H7490140

sind in Abbildung 3.2 anhand des Fräskopfes H7490140 dargestellt. Das Material ist zur Verdeutlichung der Funktionsweise des Fräasers grau. Der erste Parameter gibt, wie in der Abbildung 3.2 erkennbar ist, an, wie tief der Fräskopf vertikal in das Material eindringen kann. Er wird im Folgenden als AP bezeichnet. Ein ähnlicher Parameter ist AE. Er gibt an, wie weit sich der Fräskopf horizontal im Material befinden kann, um effizient zu arbeiten. Beide sind wichtig für den Materialkontakt. Ein tieferes oder weiteres Eindringen könnte die Effizienz negativ beeinflussen, oder den Werkzeugkopf zerstören. Zusammen ergeben sie vereinfacht ein Viereck, welches rot dargestellt ist. Der andere Parameter, der einen direkten Einfluss auf die Abtragsrate hat ist die Vorschubgeschwindigkeit VF. Ihre Richtung geht aus der Abbildung heraus. Dabei handelt es sich um eine berechnete Größe, die sich als Produkt aus der Drehzahl des Fräskopfes, der Zahl der Zähne und dem Vorschub pro Zahn ergibt[24]. Die Einheit kann in Millimeter pro Minute angegeben werden. VF beschreibt die zurückgelegte Strecke pro Zeit. Die gesamte mögliche Abtragung pro Zeit ergibt sich wiederum durch das Produkt aus VF, AE und AP. Pro Vorschub des Bohrkopfes kann er eine Fläche die dem Produkt aus AE und AP, also dem roten Rechteck in Abbildung 3.2, entspricht abtragen.

Neben diesen Parametern ist die Eignung für das Material des Werkstücks wichtig, sowie die Feinheit welche der Werkzeugkopf erreichen kann. Diesbezüglich wird zwischen dem Schruppen, dem Vorschlichten, dem Feinschlichten und dem Feinstschlichten unterschieden. Ein Maß dafür, wie rau eine Oberfläche ist, ist die gemittelte Rauhtiefe. Unter das Schruppen fallen grob die Oberflächen mit Tiefen zwischen 220 und 46 Mikrometern. Das Feinschlichten befasst sich mit Werten zwischen 4,8 und einem Mikrometer und ist damit

weit feiner. Auch wenn das Schruppen gröber ist hat es den Vorteil, dass es allgemein betrachtet schneller Material abträgt. Je feiner die Anforderungen sind, desto länger braucht der Fräsvorgang pro Volumen.

3.2.2 Stand der Technik

Das Fräsen an sich ist keine neue Technik und wurde bereits vielfach betrachtet, beschrieben und angewandt. Ein Patent für eine Fräsmaschine lag in den USA z.B. bereits 1925 vor [25]. Das System des Fräsens gibt es entsprechend bereits länger. Die Systeme, Anwendungsarten und die Feinheit der Werkzeuge wurde mit sich verbessernder Technik ebenfalls besser. Mittlerweile gibt es Metallfräsköpfe, welche einen Radius von 0.1 Millimeter erreichen[23]. Den Möglichkeiten sind aber aufgrund physikalischer Eigenschaften noch Limitationen auferlegt. Je kleiner der Fräskopf zum Beispiel ist, desto stärker muss er eingefasst werden, unter anderem um Schwingungen und Werkzeugverschleiß zu begrenzen. Dadurch ist es besonders bei Geometrien, bei denen wenig Platz ist, schwer möglich bis unmöglich feine Details, für die z.B. besagter 0.1 Millimeter-Fräskopf gebraucht würde, aus dem Material heraus zu fräsen. Für manche Anwendungen gibt es spezialisierte Köpfe oder alternative Herangehensweisen. Deren Fertigung und Anwendung ist aber mit aktuellen Techniken noch nicht ausreichend schnell und billig möglich, was die Preise und die für eine Fertigung benötigte Zeit nach oben treibt[22]. Entsprechend ist das Fräsen zwar bereits eine ältere Technik, aber dennoch eine, welche noch im Wandel begriffen ist.

3.2.3 Umsetzung des Kostenmodells im Programm

Zur Umsetzung wurde ein weiteres Python Programm geschrieben, in das die Parameter und Einschränkungen einfließen sollten. Als Konstante wurden dem Programm verschiedene Fräsköpfe als Ausgangssituation übergeben. Die einzelnen Köpfe, welche in einer Key-Value Tabelle abgelegt wurden, hatten dabei eine Bezeichnung zur späteren Rückführung auf den tatsächlichen Werkzeugkopf, sowie die oben genannten Parameter VF, AE und AP. Als Schlüssel wurde der Durchmesser der Köpfe genutzt. Es wurde eine Liste für das gröbere Schruppen und eine für die Feinarbeit am Ende erstellt. Da es eine Vielzahl an Fräsköpfen gibt, welche in den benötigten Größenordnungen operieren, wurde ein maximaler Durchmesser für die Köpfe gewählt. Dieser Durchmesser ergab sich aus der Betrachtung der Geometrie. Wie in Abbildung 2.7 bereits gut zu erkennen ist, bildet sich hier durch die Geometrie eine Engstelle. Durch sie wird beschränkt, wie tief ein Werkzeug mit einem bestimmten Durchmesser eindringen kann. An sich ist diese Engstelle von der optimierten Geometrie abhängig. Für diese Größe wurde eine Konstante erstellt. Dadurch können bestimmte Köpfe direkt ausgeschlossen werden. Ansonsten wird der Kopf mit der größten Materialabtragungsrate bevorzugt gewählt. Die Wahl des Fräasers ist aber nicht immer nur davon abhängig. Als zweites wird bei einem Teil der Fräsgänge noch betrach-

d1 (mm)	Bezeichnung	ae (mm)	ap (mm)	vf (mm/min)	Abtragung (mm ³ /min)
3	H7490180	0.36	0.36	2431	315.058
2.5	X6836160	0.3	0.3	1891	170.19
2	H7490140	0.28	0.28	2849	23.362
1.5	X6832120	0.18	0.18	1891	61.268
1.2	X6836108	0.144	0.144	1891	39.212
1	H7490100	0.18	0.18	3230	104.652
0.8	X6832080	0.112	0.112	1891	23.721
0.6	X6832060	0.084	0.084	1597	11.268
0.5	X6832050	0.07	0.07	1324	6.488
0.4	X6832040	0.072	0.072	1092	5.660

Tabelle 3.1: Tabelle der im Programm verwendbaren Schruppköpfe nach [23]

tet, ob der Fräskopf nah genug an die Zielgeometrie herankommt.

Auch mit dieser Einschränkung gibt es pro Durchmesser noch viele verschiedene Köpfe. Dabei wurde immer der Kopf genommen, welcher die beste Materialabtragsrate aufwies. Die Endauswahl für die Schruppköpfe ist in Tabelle 3.1 aufgelistet. Sämtliche Angaben sind dabei unter einem Winkel von 30°. Dabei ist in der ersten Zeile der Durchmesser d1 angetragen, welcher im Model für die Zugänglichkeit verwendet wird. Danach kommt die Bezeichnung, AE, AP und VF sowie das Produkt aus den letzten drei Parametern, die Abtragsrate. Die Angaben wurden dabei dem Katalog im Onlinewerkzeug [23] entnommen. Bei AE, AP und VF handelt es sich dabei um maximal Empfohlene Einstellungen fürs Fräsen. Der Bezug zum Werkmaterial Ti6Al4V wurde dabei durch eine Vorauswahl im Werkzeug bereits hergestellt. Die Angaben gelten entsprechend nur für dieses Material.

Zurück im eigentlichen Programm werden diese Parameter eingesetzt, um die Anzahl der Bahnen, welche in Abhängigkeit vom Filletradius für die Fertigung benötigt werden, zu bestimmen. Da für das Fräsen immer komplette Bahnen gefahren werden müssen, wird der Filletradius standardisiert von der Seite aus betrachtet. Des weiteren gilt, dass eine bestimmte Menge an Restmaterial übrig ist, das abgetragen werden muss. Vereinfacht wird die Form des Restmaterials auf eine Box, einen größeren Radius, oder ein Dreieck gesetzt, um verschiedene Fälle angehen zu können. Um zu berechnen wie viele Bahnen gefahren werden müssen, um dieses Restmaterial abzutragen, wird ein Raster über den Querschnitt gelegt. Der Querschnitt ist in Abbildung 3.3 zu sehen. Die Elemente des Rasters entsprechen der roten Box aus Abbildung 3.2, da das die Menge an Material ist, welche pro Schruppgang abgetragen werden kann. Sie sind im Beispiel quadratisch und ebenfalls rot, da auch AE und AP bei den Schruppköpfen aus Tabelle 3.1 immer gleich groß waren. Die Geschwindigkeit VF ist in dieser Querschnittsbetrachtung vorerst irrelevant, da sie in die Zeichnung hineingeht und die Länge einer Bahn betrachtet. Für jedes Feld des Rasters wird dann geprüft, ob ein Teil davon im abzutragenden Material liegt und ob eine Fräsrei-

he an diesem Punkt einen Teil des tatsächlichen Fillets abträgt. Diese Überprüfung wird über die Eckpunkte des Gitterelements gemacht. Die Geometrie des Fillets ist durch eine vereinfachte Kreisformel $x = \sqrt{r^2 + (y - r)^2}$ gegeben. Da nur ein Teil des Kreises von Interesse ist, wird der absolute Wert betrachtet. Außerdem kann der Radius hier auch für die Koordinaten des Mittelpunktes verwendet werden, da der Kreismittelpunkt in diesem Abstand im Koordinatensystem liegt. In der Abbildung ist der Viertelkreis welcher das eigentliche Fillet markiert blau. Dasselbe Vorgehen wird beim Restmaterial angewandt, wobei hier die Formel einer Geraden oder direkt die y-Koordinate für die Überprüfung verwendet werden kann. Für die angenommene Außengrenze des zu fräsenden Bereichs ist in der Abbildung 3.3 ein grüner Viertelkreis mit Radius drei Millimeter eingezeichnet. Dieser kann in verschiedenen Formen und Größen angenommen werden. Für den relativen Preisunterschied zwischen Filletradien macht das aber keinen Unterschied, da bei jedem anderen Radius die selbe Menge an Material wegfällt. Ausnahme wäre, wenn das überstehende Material so klein wäre, dass direkt geschlichtet werden könnte. Dadurch kann die Anzahl der Schruppbahnen berechnet werden, indem die Zahl der verwendbaren Kästen gezählt wird. Anzumerken ist, dass das Netz um entweder AE oder AP des Schlichtkopfes verschoben ist. Es wird das genommen, was kleiner ist. Das Ziel davon ist, dass dieses Material nicht doppelt abgetragen wird. In den meisten Fällen wird es durch die Größen des Schruppkopfes aber keinen Einfluss haben.

Der zweite Teil der zum Programm hinzugefügt wurde war die Betrachtung der Schlichtzüge. An sich muss jeder Teil der final übrig bleibenden Fläche geschlichtet werden. Um die Zahl der Schlichtzüge, welche weniger Material abtragen als die Schruppzüge, klein zu halten, wird dafür die breitere Seite, also das größere von AE und AP, hergenommen. Deshalb wurde das Netz vorher um das Kleinere von beiden verschoben, wie es nahe an den Achsen des Koordinatensystems in Abbildung 3.3 zu erkennen ist. Da davon ausgegangen wird, dass beim Schruppen bereits alles Material beseitigt wurde, außer der schmalen Schicht die geschlichtet werden muss, wird hier keine Tiefe betrachtet. Das zwei dimensionale Problem wird hier eindimensional. Die Strecke außerhalb des Fillets, welche vereinfacht gerade angenommen wird, wird durch das jeweils größere von AE und AP geteilt. Das Ergebnis aufgerundet, da eine Bahn unabhängig von der Materialüberdeckung vollständig durchlaufen werden muss, ist ein Teil der Schlichtbahnen. Für den zweiten Teil, der den eigentlichen Radius betrifft, wurde die Länge des Viertelkreises berechnet. Der Wert wurde wieder durch die Länge geteilt. Ursprünglich wurde die Zahl der Bahnen im Radius auf die selbe Weise berechnet, wie außerhalb des Radius. Lediglich wurde hier eine Bahn abgezogen, da sich die Bahnen an einem Punkt überlappen. Wie aus der Betrachtung von Abbildung 3.3 hervorgeht, sind aber nach dem Abtragen der Rasterbahnen und den feinen Bahnen, die bei der richtigen Anwendung am Schluss kommen, noch große Bereiche vom Material übrig. Da das Schlichten dieses Materials Zeit kosten würde, wird auch dieser Bereich geschruppt. Um sicherzustellen, dass der Bereich abgetragen wird, sodass

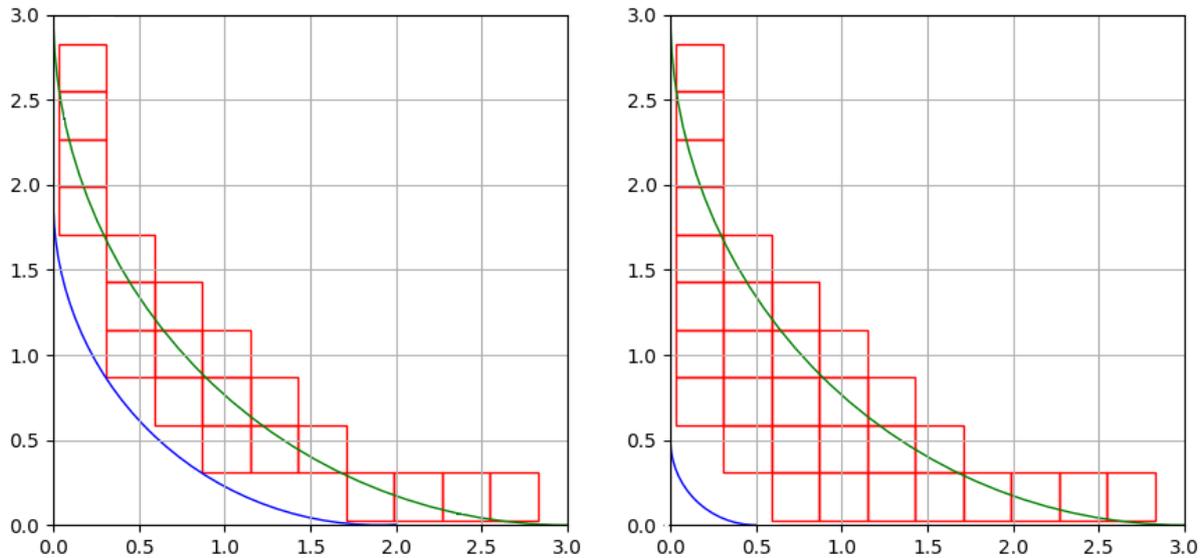


Abbildung 3.3: Darstellungen der Schruppreihen. Links mit einem Filletradius von zwei Millimetern (blau), rechts mit 0.5 Millimeter Radius

wirklich nur ein Minimum geschichtet wird, wird hier überprüft, welcher Schruppkopf für diese Bufferzone geeignet ist. Dafür wird ein Kreis mit dem Filletradius als Radius auf den Ursprung gelegt. Die x-Koordinate des Kreises im oberen rechten Quadrant wird auf der Höhe des Radius abzüglich der Abtragungshöhe des Schlichtkopfes berechnet. Das Doppelte des Wertes ist die maximale Breite die der raue Kopf vereinfacht haben darf. Entsprechend wird er ausgewählt. An sich ist der Bohrkopf natürlich rund, weshalb praktisch größere Köpfe gewählt werden könnten. Die gesamte Länge des Filletradius wird dann durch die passende Abtragungslänge, also AE oder AP, geteilt. Da hier die Schruppköpfe aus Tabelle 3.1 genommen werden, ist es egal, ob AE oder AP für die Berechnung genutzt wird. Bildlich werden diese Flächen auf den Radius gelegt. Das Ziel ist ihn komplett zu belegen. Die Zahl der Bahnen wird wieder aufgerundet. Bildlich ist das in Abbildung 3.4 dargestellt. Die roten schrägen Kästen sind die Zwischenstufen, der blaue Viertelkreis der Filletradius. Wie erkennbar ist, wird der Materialüberstand auf ein Maß reduziert, bei dem es mit einem einzelnen Schlichtgang abtragbar ist. Sollte dennoch Platz übrig sein wie es beim obersten Schrägen Kasten fast der Fall ist, so wird davon ausgegangen, dass eine Verschiebung der einzelnen Rasterzeilen das kompensieren kann. Dadurch ergeben sich drei verschiedene Zahlen für die Anzahl der Bahnen. Wie aus Tabelle 3.1 hervorgeht, haben sie verschiedene Geschwindigkeiten und Abtragungsraten. Jede Bahnzahl wird zusammen mit der Geschwindigkeit des zugehörigen Fräskopfes betrachtet. Die Gesamtlänge, also das Produkt aus Bahnlänge und Bahnzahl wird dann durch die Geschwindigkeit des zugehörigen Fräskopfes geteilt. Dadurch wird die Anzahl der Minuten berechnet. Diese wird dann mit der Zahl der Schaufeln verrechnet. Die verschiedenen Zeiten der unterschiedlichen Bahnarten werden aufsummiert und in Stunden

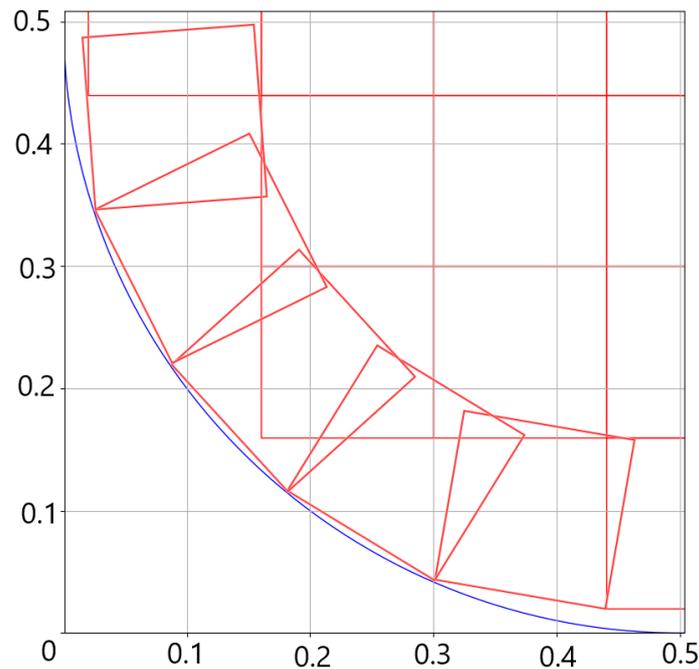


Abbildung 3.4: Zoom auf den Filletradius, bei dem die Zwischenreihen als rote Kästen eingezeichnet wurden.

umgerechnet. Auf sie kann dann der angenommene Preis von 80€ pro Stunde angewandt werden.

Da es an sich verschiedene Arten von Schaufelblättern geben kann, werden hier Splittschaufeln und Hauptschaufeln getrennt betrachtet. Für die Splittschaufeln kann eine andere Länge angenommen werden. Die Kosten werden aufsummiert.

3.3 Betrachtung der Kosten an einem Beispielprozess

Für die Darstellung wurde ein Bereich von zwei zu 0,5 Millimetern für zehn Hauptschaufeln betrachtet. Beide Grenzen sind in 3.3 dargestellt. Für den eigentlichen Vergleich wird dabei eine Änderung von 0,1 Millimetern für den Radius verwendet. Um das Ergebnis des Kostenmodells besser darzustellen, wurde in Abbildung 3.5 ein Schritt von 0,01 Millimetern verwendet. Dadurch ergaben sich mehr Werte und ein leichter auszuwertendes Bild. Das Programm wählte für das Modell in diesem Fall den Schruppkopf H7490140 und den Schlichtkopf H7490100, der bei anderer Anwendung auch zum Schruppen verwendet werden kann. AP liegt in diesem Anwendungsfall zum Schlichten bei 0,03 mm, AE 0,09mm und vf bei 7560 Millimetern pro Minute. In der Abbildung sind die vier verschiedenen betrachteten Bahnarten einzeln dargestellt. Die Summe aus ihnen ergibt das finale Kostenmodell. Zuerst wurden oben links die Kosten angetragen, welche durch die Raster-

3 Kosten bei der Fertigung und im Betrieb

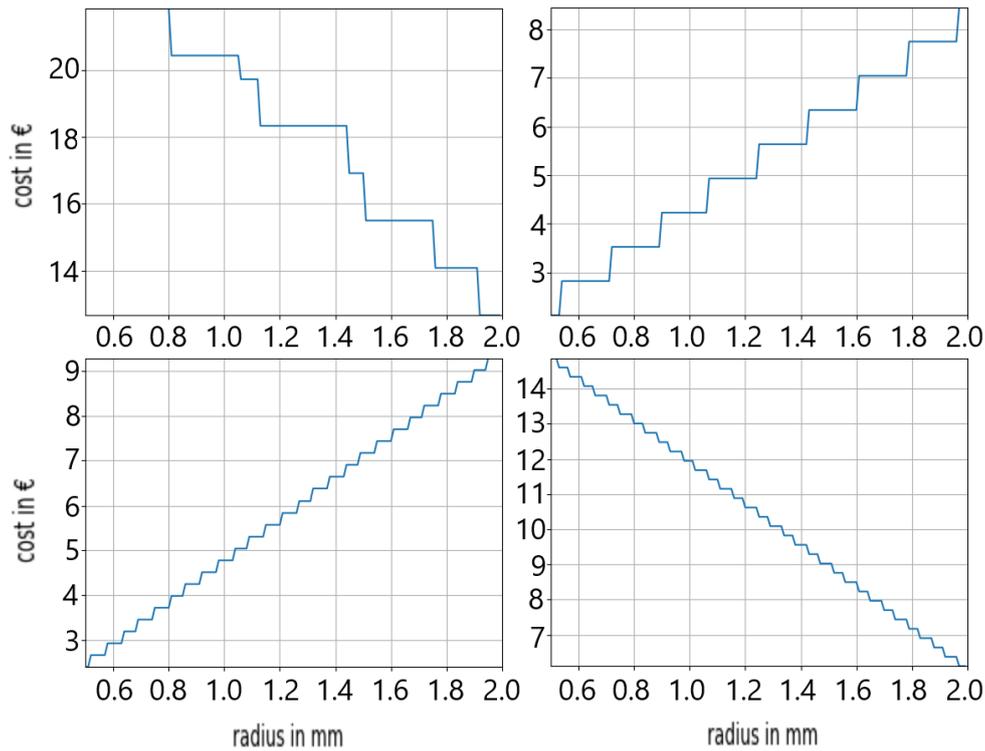


Abbildung 3.5: Der Kostenverlauf der vier betrachteten Bahnarten einzeln mit höherer Abtastung (0.01 mm). I. Die Schruppbahnen II. Das Bufferlayer III. Schlichtbahnen im Radius IV. Restschlichtbahnen

bahnen entstehen. Ihr Verlauf ist, wie sichtbar ist, nicht linear, sondern fällt stufenartig, je größer der Radius ist, bzw. fällt mit der Menge des abzutragenden Materials. Der Verlauf ergibt sich durch die Annäherung des Rasters an die tatsächliche Geometrie. Dabei kann es durchaus vorkommen, wie zum Beispiel bei den Schritten zwischen 1.2 und 1.4 Millimetern, dass kein neues Rechteck zum Raster dazu kommt. An anderen Stellen der Geometrie, besonders da diese symmetrisch zur Schrägachse ist, kommen dafür mehrere gleichzeitig dazu, wie es zwischen 1.1 und 1.2 zu sehen ist. Das wirkt sich auch auf das finale Kostenmodell aus und führt, da hier ein unregelmäßiges Fallen in das System gebracht wird, dass die Preise zwischen zwei Radien sowohl steigen, als auch fallen können, was die Realität nicht widerspiegelt. Was sich ändert, aber nicht vom System erfasst wird, ist die Menge an Material, die tatsächlich abgetragen wird. Durch eine Angleichung an die realen Umstände könnte eine gleichmäßigere Änderung erreicht werden. Es wird aber als realistisch angesehen, dass die Zahl der benötigten Schruppgänge nicht gleichmäßig fällt. Als Vereinfachung wird die Formel

$$\left[\frac{0.5 * r_f^2 + (r_s - r_f) * 2r_s - (r_s - r_f)^2}{AE * AP} \right] * \frac{l}{vf}$$

verwendet. Dabei wird aufgrund der einfacheren Geometrie vom Quadrat als Ausgangsfläche ausgegangen. r_s steht für den Startabstand, bzw. die Länge des Quadrats. r_f steht für den jeweiligen Filletradius. l ist die Länge des Fillets. Für sie werden die Strecken zwischen den Punkten aus einer der Ausgangsdateien von Bladegen aus der Prozesskette aufsummiert. Die Formel zählt für eine Klinge.

Der zweite Graph ist bereits an sich gleichmäßiger. Er stellt die Schruppgänge dar, die als Buffer nahe am Fillet verwendet werden. Da es weniger sind ist auch der Preis verglichen mit dem der restlichen Schruppgänge geringer. Er hängt direkt vom Umfang des Fillets ab. Er steigt gemäß $Durchmesser * \pi$ an. Dementsprechend kann auch für diesen Anteil der Kosten vereinfacht von einem linearen Anstieg ausgegangen werden. Da nur ein Viertel des Kreisumfangs das eigentliche Fillet ausmacht, wird

$$\lceil \frac{r_f * \pi}{2 * \max(AE, AP)} \rceil * \frac{l}{vf}$$

als Formel für die Zeit in Minuten angenommen.

Die beiden verbleibenden Plots zeigen den Verlauf der Schlichtreihen. Der erste steht für die Schlichtbahnen direkt im Radius. Sie werden gleich zur Transferschicht aus Plot 2 berechnet. Der Unterschied zwischen beiden ist der verwendete Kopf. Da dieser hier kleiner ist, kommt bereits bei kleineren Änderungen eine neue Bahn dazu. Wie im Bild auffällt, verlaufen die Graphen in Plot drei und vier relativ ähnlich, wenn auch unterschiedlich schnell. Das liegt daran, dass die zu bearbeitende Länge, welche hier betrachtet wird, sich aus dem Doppelten der Differenz aus r_s und r_f ergibt. Je größer also der Radius wird, desto kleiner wird der dadurch betrachtete Bereich. Die Länge wird dann durch das Maximum aus AE und AP des Schlichtkopfes geteilt. Die Zahl wird aufgerundet, mit der Länge multipliziert, und dann durch die Geschwindigkeit des Kopfes geteilt. Dadurch ergibt sich die Zeit für diesen Teil der Arbeit.

Die Gesamtzeit ergibt sich aus der Summe all dieser Anteile.

$$\begin{aligned} & \lceil \frac{0.5 * r_f^2 + (r_s - r_f) * 2r_s - (r_s - r_f)^2}{AE_1 * AP_1} \rceil * \frac{l}{vf_1} \\ & + \lceil \frac{r_f * \pi}{2 * \max(AE_2, AP_2)} \rceil * \frac{l}{vf_2} \\ & + \frac{l}{vf_s} * (\lceil \frac{2 * (r_s - r_f)}{\max(AE_s, AP_s)} \rceil + \lceil \frac{r_f * \pi}{2 * \max(AE_s, AP_s)} \rceil) \end{aligned}$$

Zur Unterscheidung der verschiedenen Fräsköpfe werden die Indizes verwendet. Die Indizes '1' und '2' stehen für verschiedene Schruppköpfe. Es kann sich dabei um die Gleichen handeln. Der Index 's' steht für den Schlichtkopf. Von diesen wird nur einer genommen. Das Ergebnis der Näherungsgleichung für den Wertebereich von 0.5 bis 2 Millimeter bei einer Abtastrate von einem hundertstel Millimeter ist in Abbildung 3.6 orange dargestellt.

Die Abtastrate wurde so hoch angesetzt, um dem Verlauf besser folgen zu können. Das Ergebnis des programmierten Modells wird im selben Plot blau dargestellt. Als Ausgangsform wurde ebenfalls, damit die Modelle zueinanderpassen, das Quadrat gewählt. Entsprechend höher sind die hier angenommenen Kosten. Da die relativen Kosten zwischen

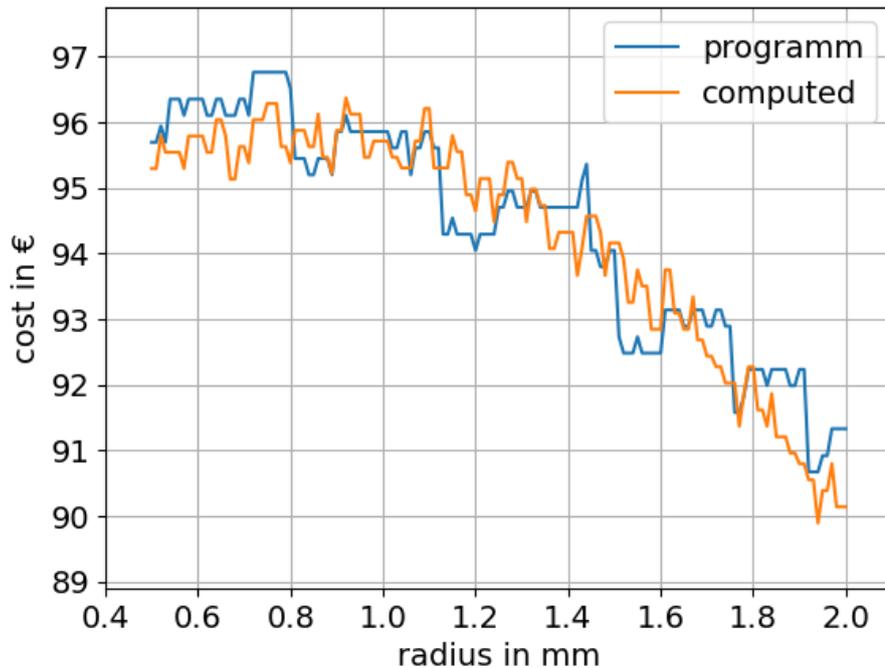


Abbildung 3.6: Kostenverlauf in Abhängigkeit vom Filletradius bei einem angenommenen quadratischen Materialstart von 3mm Seitenlänge. Programmwert in blau, Wert nach Formel berechnet in orange

den verschiedenen Radien der eigentlich wichtige Parameter sind, ist diese Verschiebung weniger wichtig. Während die Kosten sich bei dieser Ausgangsform zwischen rund 90 und 97€ bewegen, sind sie bei einer Dreiecksform zwischen 61 und 54€ zu finden. Beim Viertelkreis liegen sie im Raum zwischen 33 und 40€. Wie erkennbar ist, ist die Größe des Intervalls gleich. Das entspricht der im Kapitel 3.2.3 getroffenen Annahme.

An sich ist die Erwartungshaltung, dass sich bei steigendem Radius ein monoton fallender Graph ergibt. Das das nicht der Fall ist liegt an den verschieden schnell steigenden, bzw. fallenden Kurven aus Abbildung 3.5. Durch die Überlappung kommt es zu dem dargestellten Muster. Die kleinen Änderungen in der Programm-Kurve sind dabei größtenteils auf die Schlichtgänge zurückzuführen. Die größeren auf die Schruppgänge. Die kleineren Fluktuationen werden dabei als verhältnismäßig nahe an der Realität angesehen, da auch die Zahl der Kurven, welche das Fillet direkt betreffen, nicht im gleichen Maße, wie die außerhalb des Radius, abnimmt. Auch die Werte aus dem zweiten Diagramm in Abbildung 3.5 werden bis zu einem gewissen Grad als ausreichend nahe an der Realität angesehen.

Allerdings gibt es bei diesen Überlappungen, welche ein professionelles Programm vermeiden könnte. Die drei Plots zeigen ein gleichmäßiges Muster. Zuletzt bleiben die Bahnen vom Raster. Bei diesen gibt es zwar keine Überlappung, allerdings gibt es wie in Abbildung 3.3 sichtbar ist mehrere Bahnen, welche nur wenig Material abtragen. Dadurch, dass Bahnen schon bei leichter Überdeckung gezählt, bzw. nicht gezählt werden, kommen Bahnen meist auf einmal dazu. Da der Materialabtrag der neuen Bahnen oft durch eine bessere Platzierung bestehender Fräszüge abgedeckt werden könnte und dieser dadurch direkter mit den Radien zusammenhängen dürfte, kommen starke Abweichungen in die Zahl der tatsächlich benötigten Bahnen und den Preis. Die Fräsreihen an der grünen Linie aus Abbildung 3.3 sind dabei weniger wichtig, da sie konstant sind und den relativen Preis nicht beeinflussen. Ohne diese Abweichungen wären alle vier Bahnarten direkter mit dem sich ändernden Radius des Fillets verbunden. Ein gleichmäßigeres Abfallen des Preises wäre das Resultat.

In der Realität sind die Fertigungspreise um einiges höher, da zu den betrachteten Kosten noch die Kosten für die Fräsköpfe dazukommen. Da diese sich abnutzen und auch zum Teil mehrmals pro Fräsung durch neue Köpfe ausgetauscht werden müssen. Im Schnitt lässt sich sagen, dass bei einer Halbierung des Radius sich die Zahl der benötigten Köpfe verdoppelt[22]. Jedes Mal wenn ein Kopfaustausch vorgenommen wird, bleibt außerdem eine Unebenheit zurück, da es bislang nicht möglich ist, wieder genau an der selben Stelle anzusetzen. Abhängig von der geforderten Rauheit kommen entsprechend Kosten für den Ausgleich der Unebenheit dazu[22]. Das größte Problem das die Kosten besonders bei kleinen Radien und bestimmten Geometrien hochtreibt, ist die Zugänglichkeit zu den Frässtellen. Je kleiner die Spitze eines Fräskopfes ist, desto stärker, bzw. näher an der Kontaktstelle zum Material, muss er unterstützt werden, um sich nicht zu schnell abzunutzen. Dadurch wird es bei kleiner werdenden Radius schwerer, an die Geometrie heranzukommen. Problemstellen werden dabei durch die Höhe der Schaufeln, deren Winkel und ihren Abstand zueinander erzeugt[22]. In der aktuellen Simulation wurde lediglich der Zugang der Spitze des Instruments durch einen festen Parameter berücksichtigt. Die tatsächliche Einschränkung setzt dem Radius, der durch normale Fräsinstrumente erreichbar ist, Grenzen. An diesem Punkt müssen spezialisierte Werkzeuge eingesetzt werden, was die Kosten enorm steigert[22]. Dies ist bei kleinen Geometrien und Details wie sie bei der betrachteten Geometrie vorkommen ein starker Faktor. Diese Zusätze können aber im Rahmen dieser Arbeit nicht ausreichend betrachtet werden.

4 Optimierung auf einen Kompromiss aus Betriebs- und Fertigungskosten

4.1 Grundlagen der Optimierung

Das Ziel einer Optimierung ist es, ein gegebenes Problem, im Anwendungsfall die Verdichtergeometrie, auf ein oder mehrere Merkmale zu optimieren. Das heißt es wird versucht, diese Merkmale möglichst nahe an einen bestimmten Wert heranzubringen. Je nach System gibt es außerdem Nebenbedingungen, deren Einhaltung ebenfalls Zielparameter der Optimierung ist. Da die Maschine nicht über die Möglichkeit verfügt, dabei vollständig gezielt vorzugehen, und auch nicht mit der Logik, welche ein Mensch anwendet, um zu einem Schluss zu kommen, urteilen kann, läuft die Optimierung über eine Vielzahl an Versuchen ab. Je nach Ansatz gibt es verschiedene Vorgehensweisen[26]. Im Rahmen der hier betrachteten Optimierung wurde das DLR-interne Programm AutoOpti verwendet. Der grobe Ablauf ist, dass die Grundparameter angegeben werden, bevor eine Anzahl an Testmodellen generiert wird. Diese Testmodelle werden dann über die Prozesskette aus Kapitel 2.2.2 simuliert. Die Ergebnisse werden miteinander verglichen und erhalten eine Wertigkeit je nachdem wie sie die Haupt- und Nebenbedingungen erfüllen. Die besten Modelle werden im Anschluss weiterverwendet. Aus ihnen werden neue Beispiele zum Rechnen generiert. Das kann über eine Vermischung der Parameter verschiedener Ergebnisse oder auch z.B. durch eine zufällige Permutation von Werten erfolgen. Diese werden dann wieder gerechnet, um letztendlich ein möglichst gutes Ergebnis zu erzielen. Das wird fortgesetzt, bis ein bestimmtes Ziel- oder Abbruchkriterium erfüllt ist, oder der Prozess abgebrochen wird[26]. Zum Anfang des Programms wird ein beliebiges Startparameterset übergeben, welches Grundlage für die erste Rechnung ist[27]. Da es aber zu viele Ressourcen verwenden würde, direkt die kompletten Modelle für jeden Versuch zu berechnen, werden Ersatzmodelle verwendet. Diese Modelle spiegeln grob die Eigenschaften des tatsächlichen Modells wieder, sind aber viel schneller auswertbar. Nachdem ein Satz Modelle normal gerechnet wurde, erhalten die Ersatzmodelle die Ergebnisse von diesen. Sie werden dann mit den vorher erwähnten Permutationen und Kreuzungen berechnet. Die Eingangsparametern der Modelle mit den besten Ausgangsparametern, welche entsprechend den größten Erfolg versprechen, werden dann wieder mit der normalen Prozesskette simuliert[28]. Für das Ersatzmodell wird das Kriging-Verfahren verwendet. Bei diesem

werden Methoden der räumlichen Statistik genutzt[26]. Dabei wird die Funktion durch eine Regressionsfunktion zusammen mit einem zufälligen Gaus-Prozess angenähert[28].

4.1.1 Stand der Technik

Die Methoden der Optimierung sind vielfältig z.B. in der Technik anwendbar. Beispiel dafür ist die Optimierung des Wirkungsgrads von Verdichtern[26]. Die Grundlagen der Theorie sind bereits vielfach bearbeitet[27] und in verschiedenen Formen angewandt worden [28][26]. Die Grundlagen der Optimierung wurden bereits vor über 2000 Jahren z.B. von Archimedes angewandt[27]. Praktisch angewandte Algorithmen für den Computer gibt es seit den 1940er Jahren[27]. Der aktuelle Stand ist, dass je nach Aufwendigkeit des Problems viele Ressourcen für gute Ergebnisse benötigt werden. Deshalb werden Ersatzmodelle zur Beschleunigung verwendet[26]. Die Möglichkeiten der Optimierung steigen zusammen mit der verfügbaren Rechenleistung. Auch Entwicklungen bezüglich der künstlichen Intelligenz und neurale Netze finden Einzug in die Optimierung[27].

4.2 Optimierung des Rotors anhand der Filletradien

Für die tatsächliche Optimierung des Rotors mit der in Kapitel 2.2.2 beschriebenen Prozesskette, welche wegen dem in Kapitel 2.2.6 genannten Fehler noch auf dem Gmsh Prozess basiert, wurden zwei Nebenbedingungen gesetzt. Die erste davon war, dass ein Massenstrom von ca. 0.42 kg/s durch die Geometrie möglich ist. Der Wert kann dabei um 5% abweichen. Die Nebenbedingung ist darin begründet, dass ein bestimmter Durchfluss durch den Verdichter erreicht werden muss, um die angestrebte Masse und damit Wärmeübertragung zu gewährleisten. Die zweite Nebenbedingung ist, dass die Spannung im Material 600 MPa nicht überschreiten darf. Höhere Spannungen könnten zu schnelleren Materialschäden führen. Das Maß dabei ist vom Material abhängig. Das Hauptziel auf das optimiert wurde war die isentrope Effizienz. Sie gibt an wie nahe der Prozess an einen idealen verlustfreien Vergleichsprozess herankommt. Als Verlust definiert ist hier eine Temperaturänderung, die nicht durch die Gesetze der Thermodynamik begründet ist[6]. Ein Beispiel wäre Erwärmung durch Reibung, die in diesem Fall nicht vermieden werden kann. Sie liegt immer zwischen 0 und 100%, wobei 100 ideal wäre. Das verwendete Ersatzmodell wird als statisch betrachtet. Die Zahl der simulierten Klingen lag bei 17. Des Weiteren wurde der Radius pro Optimierung auf eine feste Größe gesetzt. Ausgehend von dem Parameter wurden die 36 freien Parameter optimiert. Diese definieren zum Beispiel die Neigung der Rotorscheaufeln, die Form der Schaufeln an der Kontaktstelle zur Hauptplatte, oder auch die Splines welche die Form der Kontaktstelle zum Gehäuse formen. Die Anzahl wäre höher geworden, wenn mit Splitterschaufeln optimiert worden wäre. Um das Modell simpler zu halten, wurde darauf in diesem Fall verzichtet. Mehr freie Parameter

machen die Optimierung komplexer. Das kann zu besseren Ergebnissen führen, aber auch dazu, dass in vertretbarer Zeit keine funktionierende Lösung gefunden wird.

Die Optimierung terminiert unabhängig vom Ergebnis nach zweieinhalb Tagen automatisch. Da sowohl unklar ist, wann, als auch ob, das beste Ergebnis erreicht wird, kann das nicht als Abbruchbedingung hergenommen werden. Ein Abbruch beim ersten akzeptierten Ergebnis würde dagegen mit hoher Wahrscheinlichkeit weiter vom besten erreichbaren Wirkungsgrad entfernt sein. Die genaue Zeitspanne ist frei gewählt. Die Optimierung wur-

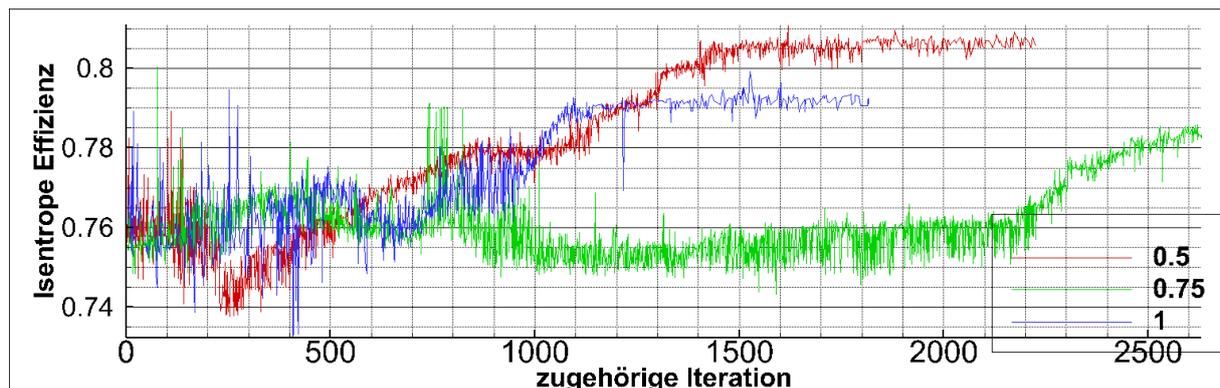


Abbildung 4.1: Darstellung der isentropen Wirkungsgrade der Geometrie über die verschiedenen Abläufe der Optimierung

de nur für eine kleine Menge an Stichwerten ausgeführt anstatt dieselbe Abtastrate wie für das Kostenmodell zu wählen. Ursprünglich sollte die Geometrie für fünf verschiedene Filletradien optimiert werden. Während der Optimierung hat sich aber herausgestellt, dass sich bei einem Radius von zwei oder anderthalb Millimetern die Fillets der verschiedenen Schaufeln schneiden. Entsprechend wurde auf die Optimierung für diese Größen verzichtet, was die Werte für 0.5, 0.75 und 1 übrig ließ. Die Ergebnisse der Optimierung werden in den Diagrammen 4.1 bis 4.4 dargestellt. 4.1, 4.4 und 4.3 zeigen dabei alle Ergebnisse bei denen sinnvolle Werte für den Wirkungsgrad berechnet wurden. Das heißt Werte über Eins werden verworfen. 1 entspricht hier hundert Prozent. Dadurch werden viele Iterationen von der Anzeige ausgeschlossen, da bei einem fehlgeschlagenen Optimierungsversuch der isentrope Wirkungsgrad direkt auf 10.000 gesetzt wird. Die ersten beiden Diagramme in den Abbildungen 4.1 und 4.2 zeigen den Verlauf des isentropen Wirkungsgrads, also den Parameter, nach dem optimiert wird. Der Verlauf für Radius 0.5 ist in den folgenden Diagrammen immer rot, 0.75 ist grün und ein Filletradius von einem Millimeter ist blau eingefärbt. Auf der x-Achse ist die zugehörige Iteration angezeigt. Das heißt, wie viele Verbesserungsschritte es gebraucht hat, um zum jeweils aktuellen Ergebnis zu kommen. Es ist erkennbar, dass der Verlauf des Wertes bei keinem der Radien monoton ist. Das liegt daran, dass nicht strikt nach einer Verbesserung gesucht wird, wie es bei einem Gradientenbasierten verfahren der Fall wäre. Stattdessen werden verschiedene Möglichkeiten zufällig erkundet. Entsprechend verbessert und verschlechtert sich der Zielwert im

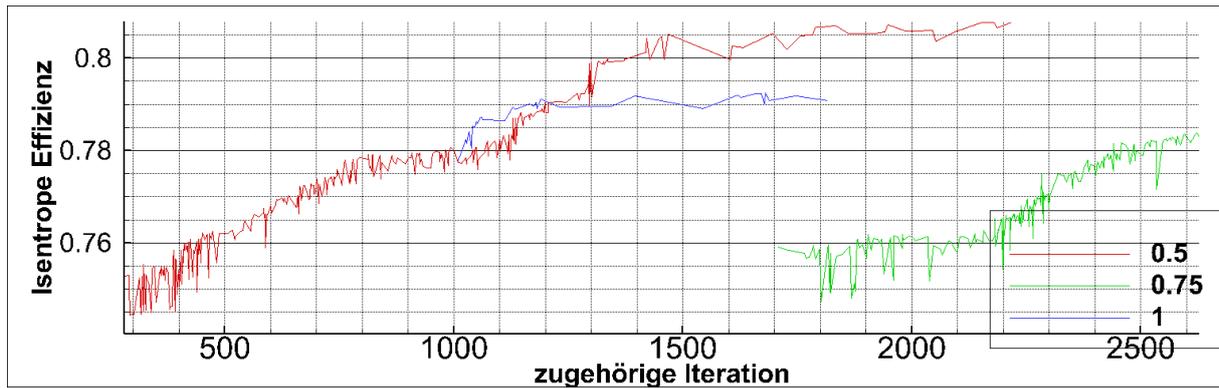


Abbildung 4.2: Darstellung der isentropen Wirkungsgrade der Geometrie über die verschiedenen Abläufe der Optimierung. Gezeigt werden nur Werte mit eingehaltenen Nebenbedingungen

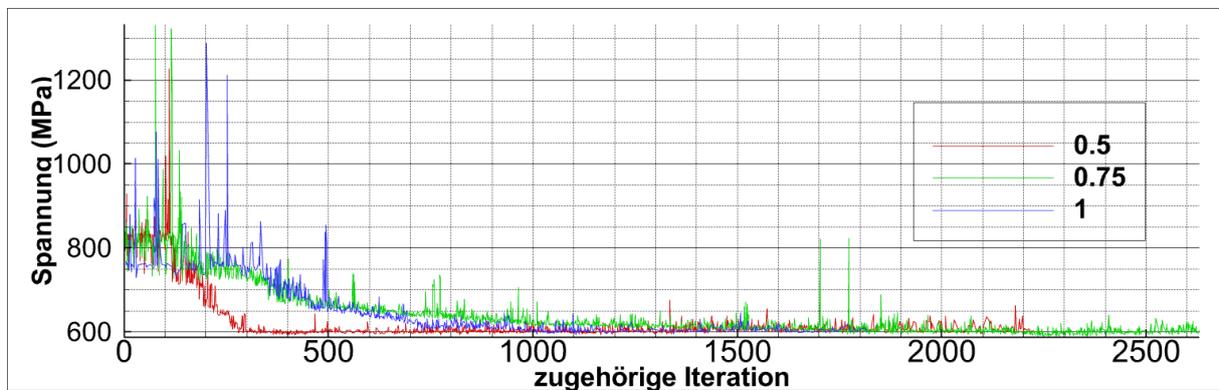


Abbildung 4.3: Darstellung der Spannung der Geometrie über die verschiedenen Abläufe der Optimierung

Verlauf der Optimierung. Es kann auch erkannt werden, dass die gezeigten Reihen am Anfang chaotischer sind, bei höheren Iterationen aber langsam ansteigen, oder bei grob einem Wert bleiben. Der rote Graph beginnt zum Beispiel schon bei nach 250 Iterationen langsam zu steigen. In höheren Iterationen stagniert der Wert und es findet keine oder nur noch eine geringe Verbesserung statt. Der Verlauf ist ähnlich bei blau, startet aber erst bei 650 Iterationen. Grün dagegen schwankt stark, bevor der Verlauf sich nach in etwa 1000 Iterationen stabilisiert. Der Anstieg startet hier erst nach ca. 2200 Simulationen. Er stagniert nicht am Ende sondern scheint noch etwas weiter anzusteigen, auch wenn der Verlauf flacher wird. Es ist hier also möglich, dass bei einer längeren Laufzeit der Optimierung noch ein besseres Ergebnis erreicht worden wäre. Anzusprechen sind auch die Peaks bei ca. Iteration 800. Diese geben gute Werte für den Wirkungsgrad. Nach einem Blick auf 4.3 lässt sich aber sagen, dass dabei die Nebenbedingungen noch nicht erfüllt waren. Der Wert kann entsprechend nicht verwendet werden. Für die nutzbaren Geometrien wird Abbildung 4.3 herangezogen. Da bei den hier dargestellten Werten die Nebenbedingungen eingehalten wurden, können die zugehörigen Geometrien an sich, sofern die

Ergebnisse der Berechnung ausreichend der Realität entsprechen, für Verdichter verwendet werden. Es ist erkennbar, dass für alle drei Filletradien mögliche Geometrien gefunden wurden. Die Optimierung des feinsten Filletradius liefert bereits bei ca. 280 Iterationen verwendbare Werte. Der beste erreichbare Wirkungsgrad für einen Filletradius von 0.5 Millimetern liegt laut der Optimierung bei 80,78%. Als nächstes erreicht der Plot der zu dem ein Millimeter-Radius gehört die Nebenbedingungen. Der höchste dabei berechnete verwendbare Wirkungsgrad liegt bei 79.25%. Wie erkennbar ist, ist der beste Wert nicht zwingend der letzte, der berechnet wurde. Bei der grünen Optimierung werden erst bei 1700 Iterationen nutzbare Inputparameter gefunden. Der Wirkungsgrad schwankt zuerst stark, bevor er langsam ansteigt. Das erreichte Maximum liegt bei 78,38% und ist damit das schlechteste Resultat aus den betrachteten. In der Abbildung ist aber zu erkennen, dass der Graph an dem Punkt, an dem die Optimierung wegen des Zeitlimits abgebrochen ist, noch gestiegen ist. Es ist entsprechend nicht auszuschließen, dass der Wirkungsgrad des grünen Plots noch über den des blauen, welcher bereits stagnierte, steigen kann. Es muss angemerkt werden, dass nicht jede Iteration dieselbe Zeit benötigt. Entsprechend hatte die grüne Optimierung dieselbe Zeit wie z.B. die rote, hat aber darin mehr Simulationen durchgeführt. In den verbleibenden zwei Abbildungen ist der Verlauf der

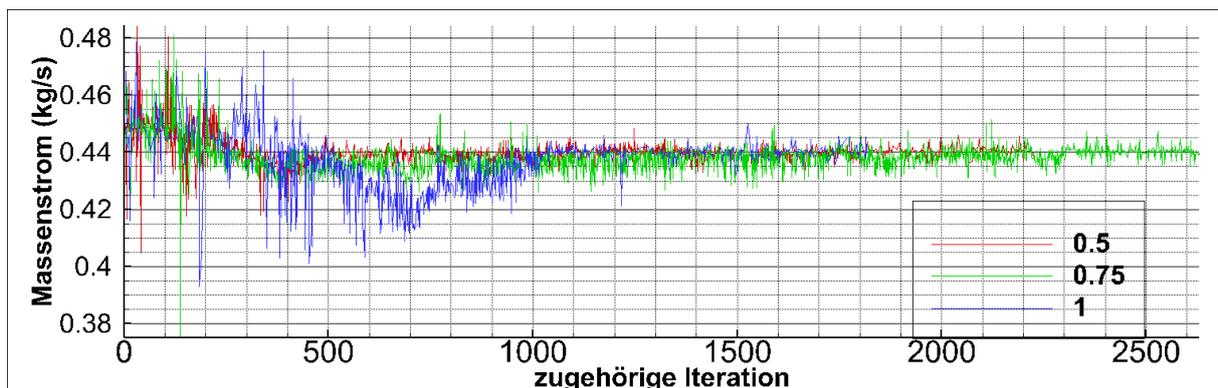


Abbildung 4.4: Darstellung des Massenstroms der Geometrie über die verschiedenen Abläufe der Optimierung

Nebenbedingungen dargestellt. Abbildung 4.3 stellt den Gesamtverlauf der Spannung im System dar. Das Maximum dafür wurde auf 600 MPa angesetzt. Wie erkennbar ist, werden besonders am Anfang sehr hohe Werte für einzelne Iterationen erreicht. Allgemein sind frühe Geometrien zu hohen Spannungen ausgesetzt. Erst nach einigen Optimierungsschritten beginnt die Spannung zu sinken. Zuerst fällt die Spannung, die zum 0.5er Radius gehört. Das könnte darauf hinweisen, dass die zusätzliche Masse außen an der Geometrie einen stärkeren Einfluss auf die Spannung hat als die Stabilisierung durch die Fillets. Danach folgen die Spannungen der beiden anderen. Sie fallen langsamer ab und es gibt öfter Spitzen im Verlauf. Das könnte darauf hinweisen, dass die Geometrien sich bereits durch kleinere Änderungen stark verschlechtern können. Um das zu prüfen müssten aber

die Geometrien und Parameter direkt betrachtet werden. Bei dem 0.5er Radius ist die Spannungsnebenbedingung zum ersten mal bei der Iteration 280 erfüllt. Bei einem Radius von einem Millimeter tritt dieser Punkt erst bei Iteration 1010 auf, bei der letzten bei Iteration 1708. Alle Geometrien die vor diesen Punkten des jeweiligen Graphen erstellt wurden, sind entsprechend nicht in der Realität verwendbar.

Zuletzt wird die zweite Nebenbedingung in Abbildung 4.4 betrachtet. Hier werden Werte zwischen 0.441 und 0.399 kg/s akzeptiert. Das tritt bei allen drei Radien sehr früh auf. Beim kleinsten Radius bereits bei der sechsten Iteration. Für den mittleren Radius wird der Wertebereich bei der 17ten Geometrie erreicht, bei der größten bei der 14ten. Während die Bedingung bei den gewählten Startparametern schnell erreicht wird, ist doch zu sehen, dass die Werte für alle drei Graphen besonders am Anfang stark schwanken und oft aus dem Bereich herausfallen. Gegen Ende stabilisiert sich der Wert um das obere Ende der Begrenzung. Dennoch fallen wegen der Nebenbedingung des öfteren Geometrien weg. Letztendlich lässt sich sagen, dass obwohl diese Nebenbedingung schnell erreicht wird, sie dennoch wichtig ist. Zusammen mit der Spannung schränkt sie die Zahl der nutzbaren Geometrien und damit die erreichbaren Wirkungsgrade ein.

Um genauer sagen zu können, weshalb die Nebenbedingungen und letztlich der Wirkungsgrad diesen Verlauf zeigen, müsste die Strukturmechanik und Strömungsmechanik bei den einzelnen Geometrien genauer betrachtet werden. Das Gewicht am Rand was die Fliehkräfte, welche auf die Geometrie wirken, verstärkt, ist aber als einschränkender Parameter für die Spannung im Bauteil zu sehen. Dem entgegenwirkt die Stabilität, die ein größeres Fillet bringen kann. Der Verlauf allgemein kann mit einer Betrachtung auf das Vorgehen bei der Optimierung ansatzweise erklärt werden. Gestartet wird mit einer kleinen Datenmenge. Dieser Ausgangspunkt kann weit von nutzbaren Geometrien entfernt sein. Das System hat zu diesem Punkt noch keinen Anhaltspunkt, welche Parameter welchen Einfluss haben. Entsprechend werden Werte verändert, um herauszufinden, welche Parameter bessere Ergebnisse bringen. Mit der Zeit werden bessere Geometrien gefunden. Etwas, das bereits näher an einer funktionierenden Geometrie ist, oder aus einer Mischung verschiedener guter Geometrien besteht, wird wie sich am Abflachen des Graphen des Wirkungsgrads zeigt, mit höherer Wahrscheinlichkeit zu ebenfalls guten Geometrien führen. Natürlich kann die Optimierung auch in ein lokales Wirkungsgradmaximum fallen. In diesem Fall würde ein Gradienten basiertes Verfahren das tatsächliche globale Maximum von dieser Startposition ausgehend nie finden. Mit der zufälligen Exploration ist es aber nur eine Frage der Zeit, bis der Weg zu diesem neuen Maximum gefunden wird. Dieses Auffinden eines neuen Weges ist wahrscheinlich bei dem grünen Graphen in Abbildung 4.1 aufgetreten.

Mit der gegebenen Anzahl an Werten für den isentropen Wirkungsgrad kann, besonders da ein Wert mit hoher Wahrscheinlichkeit sein Maximum noch nicht erreicht hat, keine sonderlich aussagekräftige Formel für den Wirkungsgrad in Abhängigkeit vom Filletradius

gebildet werden. Da der mittlere Graph noch im Aufstieg begriffen ist, wird vereinfacht ein geradliniger Verlauf angenommen. Werden die beiden vorhandenen als annähernd fertig angesehene Punkte als Maß genommen ergibt sich der Wirkungsgrad zu

$$\text{Wirkungsgrad}(\text{Filletradius}) = -0.0306 * \text{Filletradius} + 0.8231$$

4.3 Betrachtung der Wirtschaftlichkeit

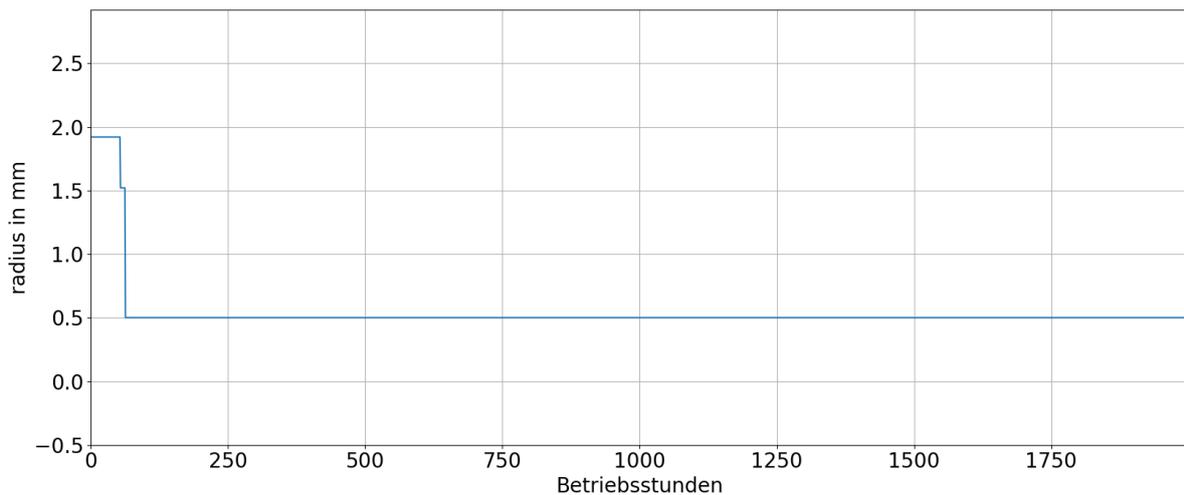


Abbildung 4.5: Darstellung des empfohlenen Radius für eine bestimmte Zahl an Betriebsstunden

Mit dem Programm aus Kapitel 3 können den Wirkungsgraden, die durch die Optimierung gefunden wurden, Kosten pro Stunde zugewiesen werden. Es ist anzumerken, dass diese nur für einen bestimmten Anwendungsfall gelten. Sie sollten dennoch als Maßstab für andere Anwendungsfälle nutzbar sein. Alle nicht extra genannten Parameter sind Freiheitsgrade für das Simulationssystem. Für den Anwendungsfall wird das in Kapitel 2.1.1 beschriebene Zyklus verwendet. Das Ergebnis ist dabei nur vom Wirkungsgrad abhängig. Für den Massenfluss wurden 0.44 kg/s für die Verbindung vom Kompressor zum Kondensator als Massenstrom gesetzt. Des weiteren wird vorausgesetzt, dass das Medium Wasser, wenn es in den Kompressor strömt, komplett verdampft ist, und nach dem Kondensator vollständig flüssig. Nach dem das Medium wieder flüssig ist, soll es eine Temperatur von 150 °C haben. Im Kondensator soll ein Wärmeabfluss von einem Megawatt auftreten. Zuletzt wird für Evaporator und Kondensator ein Druckverlust von 2% angenommen. Für die Startaussage wird ein Zeitraum von 20.000 Betriebsstunden betrachtet. Unter diesen Bedingungen ergibt sich für die drei Wirkungsgrade aus der Optimierung ein Stundenpreis von 10.6792€ für einen Wirkungsgrad von 80.78% , 10.7045€ für 79.25% und 10.7493€ für 78.38% . Auch wenn sich das direkt nicht nach viel anhört, so ergeben sich

für die benannte Stundenzahl Preisunterschiede von 1386€ für einen Größenunterschied von einem halben Millimeter für den Filletradius.

Wenn die Betriebskosten, die über den Wirkungsgrad vom Radius abhängen, auf eine bestimmte Stundenzahl hochgerechnet werden und mit den Fertigungskosten für die verschiedenen Filletradien von 17 Schaufelblättern mit einer Länge von grob 14 Zentimetern addiert werden, um daraus dann das Preisminimum für jede Betriebsstundenzahl zu erhalten, entsteht Abbildung 4.5. Dabei wurden auf der x-Achse die Betriebsstunden bis 2000 angetragen und auf der y-Achse der Filletradius in Millimetern. Der angetragene Radius ist immer der, für den die Summe aus den berechneten Fertigungskosten für das Fillet und den Betriebskosten für die Stunde minimal war. Die Schrittweite wurde auf eine Betriebsstunde gesetzt. Wie zu erkennen ist, sind nur bei sehr geringen Betriebsstunden gröbere Filletradien sinnvoll. Bereits bei geringeren Betriebsstunden, als sie in der Realität bei der Fertigung sinnvoll wären, ist ein minimaler Filletradius laut dem Kostenmodell empfehlenswert. Wie in Kapitel 3.2.3 aber erläutert wurde, kann das Kostenmodell einen Großteil der Kosten nicht sinngemäß widerspiegeln. Wenn die Werkzeugabnutzung und etwaige Spezialanforderungen an Werkzeuge beachtet werden könnten, würde das Ergebnis ein anderes sein.

5 Ergebnis

Punkt	Beschreibung	Erfüllt
1.	Geometrieerstellung in Fidelity	✓
1.1	Fidelity wird automatisch über Konsolenaufruf o.ä. gestartet	✓
1.2	Das ab Kapitel 2.2.3 beschriebene Script wird in Fidelity ausgeführt	✓
1.3	Die Geometrie der Schaufelscheibe wird über das Pythonscript nach Fidelity importiert	✓
1.4	Die Geometrie der Hauptschaufel wird über das Pythonscript nach Fidelity importiert	✓
1.5	Die Geometrie der Leitschaufeln wird über das Pythonscript nach Fidelity importiert und mit der Scheibe verbunden	✓
2.	Vernetzung über Fidelity	
2.1	Die Geometrie wird über das geschriebene Script automatisch vernetzt	✓
2.2	Die Netzkonvergenz für die CSM wird erreicht	x
2.3	Das Netz wird für Calculix lesbar exportiert	x
3.	Optimierung	
3.1	Der Fidelity Scriptaufruf ist in Autoopti integriert	x
3.2	Die Optimierung kann gestartet werden und wirft keine programmbedingten Fehler	x
4.	Kostenmodell	✓
4.1	Die Anzahl der Fräsbahnen kann näherungsweise berechnet werden	✓
4.2	Das Kostenmodell in Abhängigkeit von der Filletradiengröße ist aufgestellt und grafisch dargestellt	✓
4.3	Für das Kostenmodell wurde eine Näherungsgleichung gefunden	✓
5.	Der Einfluss von Wirkungsgrad und Kostenmodell ist dargestellt und erklärt	✓

6 Zusammenfassung und Ausblick

Die Frage, ob es einen optimalen Filletradius zu einer bestimmten Anzahl an Betriebsstunden gibt, konnte letzten Endes nicht ausreichend beantwortet werden. Dennoch wurde erfolgreich ein Programm geschrieben, das die Geometriepunkte, welche die Prozesskette liefert, in Fidelity umsetzt. Aufgrund der Neuheit des Programms hat dieser Schritt allerdings bedeutend länger gedauert, als erwartet. Erschwerend kam dazu, dass die Dateistrukturen, welche eingelesen werden sollten, ursprünglich für das vorher genutzte Programm Gmsh angefertigt worden waren, welches sich von der Herangehensweise in Teilen stark von Fidelity unterscheidet. Nach diesen anfänglichen Schwierigkeiten wurde aber sowohl die Grundscheibe als auch die Schaufeln eingelesen. Beides wurde miteinander verbunden. Der erste Oberpunkt aus Kapitel 5 konnte also, wenn auch mit Verzögerung, komplett erfüllt werden.

Der zweite Schritt konnte nur teilweise abgearbeitet werden. Die Geometrie konnte zwar ebenfalls mit Verzögerungen durch das neue Programm vernetzt werden. Dabei führte zuerst der Hexa-dominante Ansatz zum Erfolg, welcher mit Calculix aber nicht kompatibel war. Ein neues Netz war nötig. Aufgrund der Möglichkeiten in Fidelity wurde dabei von einem reinen Tetraeder-Netz ausgegangen und unter dieser Annahme weitergearbeitet. Das an diesem Punkt temporäre Netz wurde im OpenFoam Format exportiert. Darauf folgte die Anbindung an die Schnittstelle zum Rest der Kette. Diese schlug allerdings aus bislang unbekanntem Grund fehl. Die exportierte Datei lieferte ein Netz, in dem die Elementdefinition Fehler aufwies. Die Kombination verschiedener Elemente des Fidelity-Netzes lieferte unabhängig von der Restausgangsdatei weiterhin kein korrektes Ergebnis. Der Export war entsprechend in Calculix nicht verwendbar.

Das Ziel der Netzkonvergenz war an diesem Punkt vorerst durch feste Anforderungen ersetzt worden. Da das Programm durch den Exportfehler nicht in die Prozesskette integriert werden konnte, wurde auf eine Netzkonvergenzanalyse verzichtet. Die tatsächliche Optimierung konnte entsprechend nicht wie geplant mit der Fidelity-Komponente gestartet werden, weshalb das Programm nicht in Autoopti integriert wurde. Damit konnte auch die Optimierung nicht wie geplant gestartet werden, weshalb Punkte 3.1 und 3.2 als fehlgeschlagen angesehen werden.

Um dennoch einen Kostenvergleich zwischen Fertigung und Betrieb herstellen zu können, wurde die Optimierung mit der ursprünglichen Prozesskette und Gmsh durchgeführt. Dabei ergaben sich, dass sich bei den getroffenen Einstellungen Geometrien mit einem

Filletradius von 1.5 und zwei Millimetern die verschiedenen Blätter berühren, weshalb diese Optimierungsgänge abgebrochen wurden. Die Optimierungen für Radien von einem, einem halben und einem dreiviertel Millimeter liefen bis zum Zeitlimit durch. Dadurch wurden Wirkungsgrade pro Radius gefunden. Die Ergebniswerte der Optimierung für den mittleren Radius stiegen allerdings noch an, als die Optimierung aufgrund des Zeitlimits terminierte. Das Ergebnis konnte entsprechend nicht für einen Werteverlauf verwendet werden. Aus den verbleibenden beiden Werten wurde ein erwarteter Verlauf gebildet, welcher aber aufgrund der wenigen Werte sehr ungenau ist. Die erreichten Werte lagen bei 80.78% für den kleineren Radius und bei 79.25% für den größeren.

Unabhängig davon konnte ein grundlegendes Kostenmodell erstellt werden, das die Zahl der Schlicht- und Schruppbahnen bei dem Fräsen eines Filletradius näherungsweise betrachtet. Durch eine Verrechnung der Zeit, welche pro Fräsbahn benötigt wird, mit der Zahl der Bahnen und einem Beispielpreis konnte ein Preis für den reinen Fräsprozess unter optimalen Bedingungen gefunden werden. Anhand der Logik des Programms konnte eine Näherungsgleichung für den Preis der Radien im betrachteten Bereich von 0.5 bis zwei Millimetern gefunden werden. Punkt Vier wurde entsprechend komplett erfüllt. Allerdings war Teil des Ergebnisses auch, dass nur ein Teil der Kosten durch die reine Fräszeit anfällt. Spezialwerkzeuge und nicht beachtete Randbedingungen haben einen größeren Einfluss und müssen öfter verwendet werden, als vorher angenommen. Anhand der vorliegenden Daten konnte eine Radiumempfehlung für beliebige Betriebsstunden getroffen werden. Da die Daten aber nur die reine Fräszeit betrachtet, ist die Empfehlung in der Wirtschaft kaum nutzbar.

Nachfolgende Arbeitsschritte ergeben sich aus den übrigen Punkten. Die Behebung des Fehlers beim Calkulix-Export stellt die größte Hürde dar. Weiterführend kann auch geprüft werden, ob das Netz für die Beispielgeometrie konvergiert, bzw. es kann versucht werden, es allgemein zu verbessern. Auch ergibt sich aus den erwähnten weiteren Kosten bei der Fertigung der Bedarf nach einem umfassenderem Kostenmodell. Zuletzt stellt die Optimierung auf den Filletradius nur einen Teil des Verdichters da. Eine komplette Kostenberechnung zusammen mit einer vollständigen Optimierung wäre das Endziel, um der Wirtschaft tatsächlich den jeweils lohnendsten Verdichter für die Zielprozesskette aufzeigen zu können.

Literaturverzeichnis

- [1] Mayer, N.: *Klimawandel: Wie lange dauert es noch bis zum Point of no Return?*: https://www.futura-sciences.com/de/klimawandel-wie-lange-dauert-es-noch-bis-zum-point-of-no-return_6931/, 2021, (abgerufen 9.07.2024)
- [2] Umwelt Bundesamt: *Internationale Klimapolitik, Temperatur- und Emissionsziele*: <https://www.umweltbundesamt.de/themen/klima-energie/internationale-klimapolitik#temperatur-und-emissionsziele>, 2024, (abgerufen am 9.07.2024)
- [3] Europäische Umweltagentur: *Treibhausgasemissionen in der EU nach Schadstoffen*: <https://www.europarl.europa.eu/topics/de/article/20180301ST098928/treibhausgasemissionen-nach-landern-und-sektoren-infografik>, 2018 mit Aktualisierung 2023, (abgerufen am 9.07.2024)
- [4] Arpagaus, C.: *Hochtemperatur-Wärmepumpen: Marktübersicht, Stand der Technik und Anwendungspotenziale*, Berlin, Deutschland: VDE Verlag GmbH, 2019
- [5] Bräunling, W.: *Flugzeugtriebwerke: Grundlagen, Aero-Thermodynamik, Ideale und reale Kreisprozesse, Thermische Turbomaschinen, Komponenten, Emissionen und Systeme*, Berlin, Deutschland: Springer, 2015
- [6] Casey M., Robinson C.: *Radial Flow Turbocompressors: Design, Analysis and Applications*, Cambridge, England: Cambridge University Press, 2021
- [7] Dehli, M; Doering, E; Schedwill, H: *Grundlagen der Technischen Thermodynamik*, Wiesbaden, Deutschland: Springer, 2023
- [8] Witte, F.: "Heat Pump". [tespy.readthedocs.io: https://tespy.readthedocs.io/en/main/basics/heat_pump.html](https://tespy.readthedocs.io/en/main/basics/heat_pump.html), 2024, (abgerufen 29.07.2024)
- [9] Kristen, Y.: Entropie: https://www.uni-ulm.de/fileadmin/website_uni_ulm/nawi.inst.251/Didactics/thermodynamik/INHALT/ENTROPIE.HTM, ohne Datum, (abgerufen 29.07.2024)

- [10] Kriese, M.; Klöppel, S.; Setzepfand, N.; Schaffrath, R.; Nicke, E.: *Part-load behavior and start up procedure of a reverse rankine hightemperature heat pump with water as its working medium*, Boston, New England, USA: Turbo Expo: Power for Land, Sea, and Air. Vol. 86984. American Society of Mechanical Engineers, 2023
- [11] Schobeiri, M.: *Turbomachinery Flow Physics and Dynamic Performance*, Berlin, Deutschland: Springer, 2005
- [12] Gülich, J.: *Centrifugal Pumps*, Cham, Schweiz: Springer, 2020
- [13] LEADRP: *Verrundung vs. Fase: Was ist der Unterschied?*, <https://leadrp.net/de/blog/fillet-vs-chamfer-whats-the-difference/>, 2022, (abgerufen am 24.08.2024)
- [14] DLR: *Institut für CO₂-arme Industrieprozesse*, <https://www.dlr.de/de/di/ueber-uns/das-institut>, 2024, (abgerufen am 24.08.2024)
- [15] Nicke, E.: *Industrieprozesse mittels Hochtemperatur-Wärmepumpen effizient dekarbonisieren*, Berlin, Deutschland: 21. Forum Wärmepumpen, 2023
- [16] Dhondt, G.: *CalculiX CrunchiX USERS'MANUAL version 2.20*, Kein Verlag, 2022
- [17] Klein, B.: *FEM: Grundlagen und Anwendungen der Finite-Element-Methode im Maschinen- und Fahrzeugbau*, Wiesbaden, Deutschland: Springer, 2015
- [18] Shikin, E.; Plis, A.: *Handbook on Splines for the User*, New York, USA: CRC Press, 1995
- [19] Hoffmann, A.: *Hausarbeit: Einfluss des Wirkungsgrads von Turbomaschinen auf Hochtemperaturwärmepumpen*, Cottbus, Deutschland: nicht veröffentlicht (einsehbar DHBW-Manheim), 2024
- [20] González-Barrio, H.; Celleja-Ochoa, A.; Lamikiz, A.; Lacalle, L.: *Manufacturing Processes of Integral Blade Rotors for Turbomachinery, Processes and New Approaches*, Applied Science 10, 2020
- [21] Dreibati, O.; Sommer, S.; Marder, C.: *Prozessbegleitendes dynamisches Spannen zur Verzugs- und Eigenspannungsreduzierung beim Schweißen von Bauteilen*, Freiburg, Deutschland: Fraunhofer IWM, 2015
- [22] Motz, M.: Online-Meeting zur Besprechung der Verdichterfertigung ab dem 9. Juli 2024 mit DLR internem Fertiger
- [23] Fraisa: *FRAISA ToolExpert 2.0*: <https://www.fraisa.com/toolexpert/2.0.22/#!/de/mode/expert/tooltype/F/norm/all/normid/tab/zk/usecase/>

articleNo/d1/d2/d3/11/12/13/z/r/b/H/L/deg45/CutMat/ShaftType/ToolType/
performance_class/coating//beta/worktype/group/toolid/json/append/
material, 2020 (abgerufen 14.08.2024)

- [24] Schierbock, P.: *Formeln und Tabellen für metalltechnische Berufe*, Braunschweig, Deutschland: Westermann, 2019
- [25] Newman, M.: *Milling Machine*, U.S. Patent 1 689 635, Oct. 30, 1928
- [26] Krüger, F.: *Entwicklung von parallelisierbaren Gradienten-basierten Verfahren zur automatisierten, Ersatzmodell-gestützten Optimierung unter Nebenbedingungen für CFD-FEM-Verdichterdesign*, Berlin, Deutschland: Technische Universität Berlin, 2012
- [27] Martins, J.; Ning, A.: *Engineering Design Optimization*, Cambridge, England: Cambridge University Press, 2021
- [28] Schaffrath, R.; Nicke, E.; Fortshofer N.; Kunc, O; Voß, C.: *Gradient-Free aerodynamic optimization with structural constraints and surge line control for radial compressor stage*, Boston, Massachusetts, USA: Turbo Expo: Power for Land, Sea, and Air. American Society of Mechanical Engineers, 2023. S. V13DT34A005, 2023