

Masterarbeit

# Simulation von Orgelpfeifen

Rico Feist

Alma Mater Lipsiensis  
Institut für Informatik

30. Januar 2019

betreuender Hochschullehrer: Prof. Mario Hlawitschka  
Zweitkorrektor: Prof. Gerik Scheuermann

ᚱᚿᚿ ᚱᚾᚾ ᚾᚾᚾ ᚾᚾᚾ:  
ᚱᚾᚾ ᚾᚾᚾ ᚾᚾᚾ::

Ennyn Durin Aran Moria;  
pedo mellon a minno.

- John Ronald Reuel Tolkien, Inschrift am Tor von Moria

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
1.1. Geschichtlicher Abriss . . . . .	2
1.1.1. Lichttonorgel . . . . .	2
1.1.2. Hammondorgel . . . . .	3
1.1.3. Vox Continental . . . . .	4
1.1.4. Dr.-Böhm-Organ . . . . .	5
1.2. Derzeitige Situation . . . . .	5
1.2.1. Arten der Tonerzeugung . . . . .	5
1.2.2. Hard- und Software zur Simulation von Organen . . . . .	7
1.3. Problemstellung . . . . .	9
<b>2. Technischer und physikalischer Hintergrund</b>	<b>10</b>
2.1. Funktionsweise einer Orgel . . . . .	10
2.1.1. Aufbau . . . . .	10
2.1.2. Spieltisch . . . . .	10
2.1.3. Traktur . . . . .	11
2.1.4. Windlade . . . . .	11
2.1.5. Pfeifenwerk . . . . .	12
2.2. Die Physik der Tonerzeugung in Orgelpfeifen . . . . .	13
2.2.1. Tonerzeugung in Labialpfeifen . . . . .	13
2.2.2. Abstraktion für die Simulation . . . . .	23
2.3. Technischer Hintergrund der Simulation . . . . .	23
2.3.1. Hintergrund Numerische Strömungsberechnung . . . . .	24
2.3.2. Der Solver rhoPimple . . . . .	25
2.4. Ausgabe der Ergebnisse . . . . .	27
2.4.1. Technischer Hintergrund Audioverarbeitung . . . . .	27
2.4.2. Zusammenhang zwischen Pfeifenlänge, Pfeifenfrequenz und benötigter Samplingrate . . . . .	34
<b>3. Aufbau der Simulation</b>	<b>36</b>
3.1. Beschreibung des Testaufbaus . . . . .	36
3.2. Beschreibung des Ablaufes . . . . .	37
3.2.1. Ansatz 1 . . . . .	37
3.2.2. Ansatz 2 . . . . .	41
3.2.3. Ansatz 3 . . . . .	51

3.3. Bewertung der Ergebnisse . . . . .	54
3.3.1. Gedackte Pfeife . . . . .	58
3.3.2. Offene Pfeife . . . . .	67
3.3.3. Vergleich offen und gedackt . . . . .	75
3.3.4. Vergleich Simulation und Realität . . . . .	77
<b>4. Nachbetrachtung</b>	<b>85</b>
4.1. Einschränkungen der Simulation . . . . .	85
4.1.1. Zweidimensionale Darstellung . . . . .	85
4.1.2. Beschränkung auf die Simulation der Luft. . . . .	86
4.2. Zukünftige Entwicklung . . . . .	87
<b>A. Listings</b>	<b>88</b>
<b>Abbildungsverzeichnis</b>	<b>116</b>
<b>Verzeichnis der Quelltexte</b>	<b>120</b>
<b>Literaturverzeichnis</b>	<b>121</b>

# 1. Einleitung

Die Orgel ist ohne Zweifel das größte, das kühnste und das herrlichste aller von menschlichem Geist erschaffenen Instrumente, Sie ist ein ganzes Orchester, von dem eine geschickte Hand alles verlangen, auf dem sie alles ausführen kann.

---

*Honoré de Balzac*

Orgeln sind monumentale Instrumente, denen eine einzigartige Klangfülle innewohnt. Seit Jahrhunderten haben sie ihren Platz in Kirchen und Konzertsälen und sind in der Lage den ganzen Raum in Klang zu hüllen. Dabei reicht das Spektrum von leisen, zarten Tönen bis hin zum donnernden Tutti, welches durch die großen Pedalpfeifen nicht nur hör- sondern auch spürbar ist.

Doch diese Klanggewalt bringt mit sich, dass die Orgel selbst eine Größe erreicht, die alle anderen Musikinstrumente übertrifft. Majestätisch thront sie über dem Kirchenschiff oder bildet den Abschluss des Konzertsaaes. Mit ihren hunderten oder tausenden von Pfeifen übertrifft sie in ihrer Komplexität jedes andere Instrument. Jede Orgel ist ein Unikat, das genau für den Raum, in dem sie klingt, erbaut wurde.

All dies führt dazu, dass Orgeln zu den teuersten Musikinstrumenten gehören. Es erscheint daher nur logisch, dass immer wieder versucht wurde, den Klang einer Orgel nachzubilden. Die ersten erfolgversprechenden Versuche fanden mit der deutschen Lichttonorgel und der US-amerikanischen Hammondorgel in den 1930er Jahren statt. Ab da wurden kontinuierlich neue, elektronische Orgeln, entwickelt.

## 1.1. Geschichtlicher Abriss

### 1.1.1. Lichttonorgel

Eine Vorreiterrolle der samplingbasierten Simulation nimmt die von Edwin Welte in den dreißiger Jahren des 20. Jahrhunderts entwickelte Lichttonorgel ein. Das Prinzip der Tonerzeugung beruht auf rotierenden Glasscheiben, auf die mit einer Photoemulsion zuvor aufgezeichnete Schwingungskurven aufgetragen werden. Diese rotierten vor Photozellen, welche die Informationen auslasen, über Verstärker wurden diese dann an die Lautsprecher weitergeleitet und somit wiedergegeben. Ein Ausschnitt aus einer dieser Scheiben ist in Abbildung 1.1 abgebildet, die Wellenform ist deutlich zu sehen. 1935 meldete Edwin Welte ein Patent für die Lichttonorgel (DRP 712570) an. Am 17. August wurde der bereits Ende 1935 fertiggestellte, serienreife Prototyp in der Berliner Philharmonie vorgestellt. Ein weiteres Konzert folgte. Bei beiden Konzerten war die Presseresonanz sehr positiv. Die Orgel hatte bereits 16 Stimmen und 24 Register, spielbar auf zwei Manualen und Pedal<sup>1</sup>.



Abbildung 1.1.: Ausschnitt aus einer der bei der Lichttonorgel verwendeten Scheiben. Bild aus Dangel (2006).

Edwin Welte legte mit dieser Orgel den Grundstein für die Entwicklung der samplingbasierten Sakralorgeln. Das hier angewendete Samplingverfahren ermöglichte zur damaligen Zeit bereits einen Klang, der einer Pfeifenorgel wesentlich ähnlicher war als bei der synthetischen Erzeugung durch andere Instrumente, wie zum Beispiel der nachfolgend beschriebenen, etwa zeitgleich erfundenen Hammondorgel.

---

<sup>1</sup>Siehe dazu Dangel (2007).

### 1.1.2. Hammondorgel

Die Hammondorgel wurde 1933 von Laurens Hammond, einem US-amerikanischen Uhrmacher, erfunden. Die Orgel basiert auf einem Tongenerator, der Sinuswellen unterschiedlicher Frequenz erzeugt und diese anschließend miteinander mischt. Die Lautstärke der einzelnen Frequenzen wird über 9 Zugriegel bestimmt, die den Fußlagen von Orgelpfeifen entsprechen und in der folgenden Reihenfolge angeordnet sind:

$$16', 5\frac{1}{3}', 8', 4', 2\frac{2}{3}', 2', 1\frac{3}{5}', 1\frac{1}{3}', 1'$$

Diese Töne können in acht Lautstärkestufen gemischt werden, um Pfeifenregister zu simulieren. Ein „Diapason“ (deutsch: Prinzipal) entspricht dabei beispielsweise der folgenden Zusammenstellung<sup>2</sup>: 00 5756 254. Die Ziffern bezeichnen die Lautstärkestufen in Reihenfolge der Zugriegel. Jeder der Zugriegel steht dabei für einen Oberton. Werden diese zusammengemischt, entsteht ein ähnliches Frequenzspektrum wie bei der entsprechenden Orgelpfeife. Dieses Verfahren wird additive Synthese genannt.

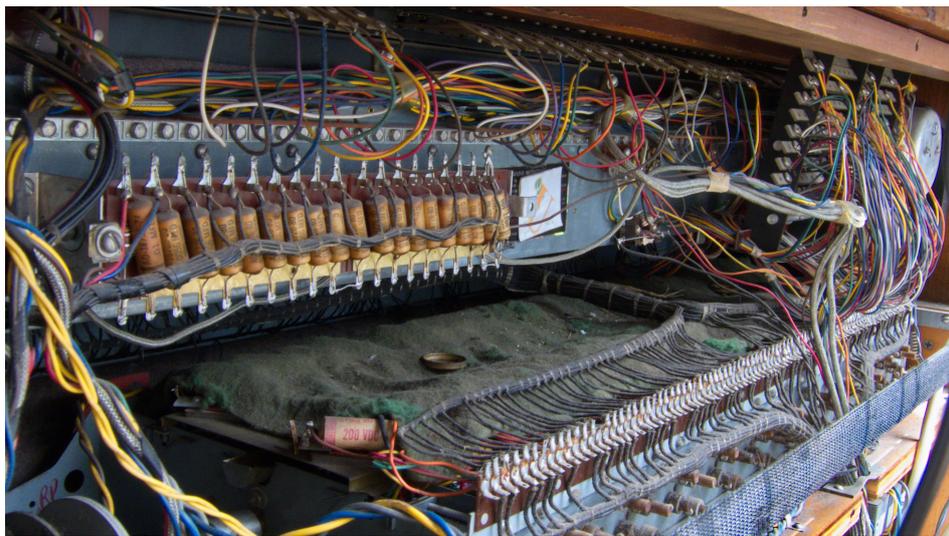


Abbildung 1.2.: Blick auf den Tongenerator einer Hammond B3. Bild aus Zbyszynski (2012).

---

<sup>2</sup>Siehe Hammond (1955)

Der Tongenerator der Hammondorgel besteht aus insgesamt 91 sogenannten Tonrädern. Die Erzeugung der Sinuswellen geschieht über elektromagnetische Tonabnehmer vor denen das Tonrad rotiert. Durch das sich verändernde magnetische Feld wird von dem Tonabnehmer eine Sinuswelle abgenommen. Über eine entsprechende Verdrahtung, die in Abbildung 1.2 zu sehen ist, werden dann diese 91 Frequenzen den jeweiligen Tasten und Fußlagen zugeordnet.

Zusammen mit einem sogenannten Leslietonkabinett ist die Hammondorgel aus der Jazzmusik, aber auch aus der Rockmusik nicht mehr wegzudenken. Auch war ihr einzigartiger Klang jahrzehntelang Teil der Fanfare zu Beginn der Tagesschau. Inzwischen gibt es auch zahlreiche Ansätze, den Klang der Hammondorgel nachzubilden. Die Nachbildung ist somit selbst zum Gegenstand von Nachbildungen geworden.

### 1.1.3. Vox Continental

Mit der flächendeckenden, erschwinglichen Verfügbarkeit von Transistoren gab es ab den 1950er Jahren Versuche, den elektromechanischen Tongenerator der Hammondorgel durch eine rein elektronische Lösung zu ersetzen<sup>3</sup>.

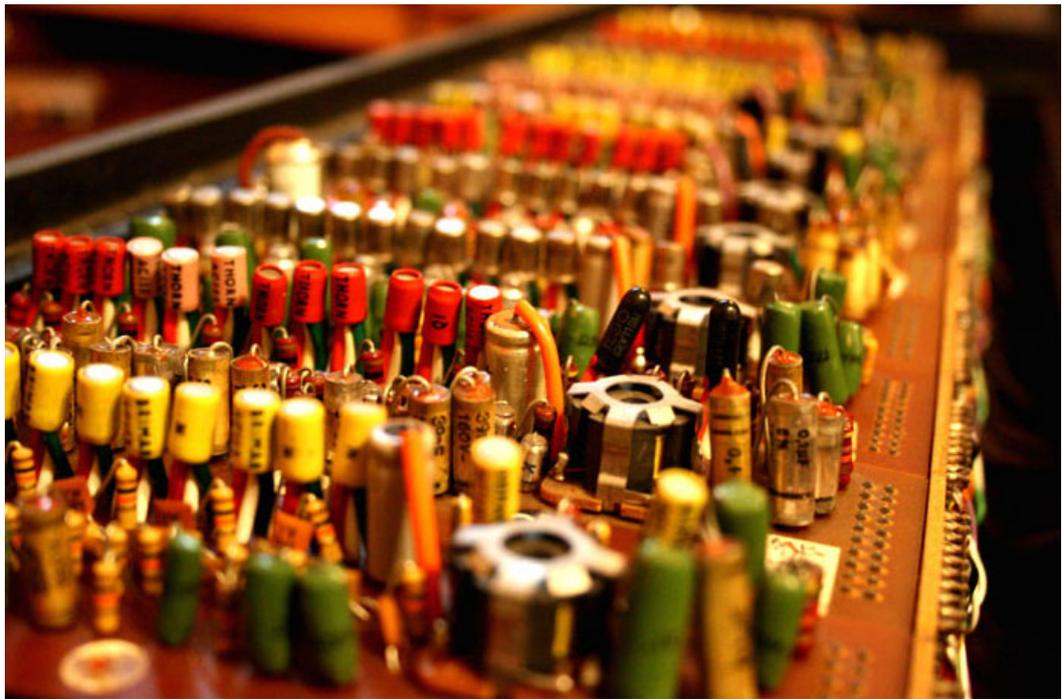


Abbildung 1.3.: Blick auf den Tongenerator Vox Continental. Bild aus Starling (2009).

---

<sup>3</sup>Siehe Combo-Organ (2019).

Vorreiter dieser Technik war die britische Firma Vox mit der Orgel Vox Continental. Der Tongenerator nutzt Germanium-Transistoren, die zu 12 LC-Oszillatoren - entsprechend 12 Halbtönen einer Oktave - zusammengeschaltet sind. Über weitere Vibratoren und Tiefpassfilter werden 84 Frequenzen erzeugt die auf jeder Taste 4 Töne bereitstellen, deren Lautstärke wie bei der Hammondorgel über Zugriegel geregelt werden kann. Der Blick auf den Tongenerator in Abbildung 1.3 zeigt deutlich die Unterschiede zur Hammondorgel.

#### **1.1.4. Dr.-Böhm-Orgel**

Aufbauend auf der Transistortechnik entwickelte der Physiker Rainer Böhm ab 1960 eine Baureihe elektronischer Orgeln. Zunächst standen die Schaltungen nur in einem Lehrbuch zur Klangsynthese zur Verfügung, später wurden diese Orgeln als Bausätze vertrieben. So wie bei der Vox Continental wird der Klang bei dieser Art der Tonerzeugung über subtraktive Synthese erzeugt<sup>4</sup>. Dabei werden die vom Oszillator erzeugten Schwingungen über Tiefpassfilter gefiltert und so unerwünschte Frequenzen entfernt.

Die Orgeln wurden als Bausätze vertrieben und alle Schaltungen mussten vom Endkunden selbst gelötet werden. Dies ermöglichte es, die Orgeln preisgünstig zu vertreiben. Somit fanden diese Orgeln weite Verbreitung als Heimorgel. Versierten Elektronikern war es auch möglich, die Schaltungen selbst zu individualisieren.

## **1.2. Derzeitige Situation**

### **1.2.1. Arten der Tonerzeugung**

Bei der Art der Tonerzeugung kann unterschieden werden zwischen dem Abspielen und Zusammensetzung zuvor aufgezeichneter Töne einerseits und der Berechnung des Tones in dem Moment, in dem der Ton erklingen soll.

#### **Samplingbasierte Tonerzeugung**

Bei einer samplingbasierten Simulation werden die Pfeifen einzeln aufgenommen und aus diesen Aufnahmen anschließend der Gesamtklang generiert. Dazu wird die aufgenommene Datei in einen Anfang, einen Mittelteil und ein Ende unterteilt. Ein einzelnes Sample führt jedoch dazu, dass die jeweilige Pfeife stets gleich klingt. Daher werden heutzutage Pfeifen mehrfach aufgenommen, um nachbilden zu können, dass Pfeifen nicht immer gleich klingen.

---

<sup>4</sup>Siehe Böhm (1986).

Ein weiterer Schritt zu mehr Realismus in einer samplingbasierten Orgelsimulation ist das Aufnehmen der Pfeifen mit Tremulant. Vor 10 Jahren begrenzte vorrangig die Größe des verfügbaren Hauptspeichers die Größe der virtuellen Orgel - nicht zuletzt aufgrund der weiten Verbreitung von Windows XP mit 32Bit im semiprofessionellen Bereich. Eine zusätzliche Aufnahme der Samples mit Tremulant war oft nicht sinnvoll, da sich dadurch die Menge der Samples und somit auch der Speicherbedarf weiter vergrößert. Inzwischen ist diese Beschränkung nicht mehr vorrangig, sodass der Tremulant heute durch zusätzliche Samples nachgebildet wird.

Zur Speicherung der Samples haben sich verschiedene Formate etabliert. Seit Anfang der 1990er Jahre sind **Soundfonts** als offener Standard definiert<sup>5</sup>. Eine Soundfontdatei ist ein Container, in dem Samples und weitere Einstellungen zu diesen Samples, wie z.B. Filter oder Hüllkurven, gespeichert sind. Diese Samples können dann von einem Hard- oder Softwaresampler wiedergegeben werden. Spezialisierte Software wie **Hauptwerk** nutzt ein eigenes Format. Dabei wird jedes Sample einzeln gespeichert und die Informationen über die Bedeutung der einzelnen Samples werden in einer eigenen Datei gespeichert. Dies hat den Vorteil, dass spezifische Eigenheiten der Simulation einer Orgel besser nachgebildet und konfiguriert werden können.

Dennoch lassen sich nicht alle physikalischen Effekte, die in einer Orgel auftreten, dadurch nachbilden. Dazu gehört auch der Effekt der überblasenden Pfeifen bei nachlassendem Winddruck. Wird die Windversorgung der Orgel abgeschaltet, entleeren sich die Magazinbälge, die für eine gleichmäßige Windversorgung unerlässlich sind, langsam und der Winddruck sinkt. Bleiben dabei weiter Pfeifen aktiv, ergibt sich ein interessanter Effekt. Zunächst wird die Pfeife leiser, verstummt dann und nach kurzer Zeit ertönt sie erneut, jedoch nun im nächsthöheren Oberton. Je nach Register und Kapazität der Bälge geschieht dies mehr als einmal. Dieser Effekt wird auch in zeitgenössischen Kompositionen wie „Volumina für Orgel“ von György Ligeti verwendet. Somit können derartige Kompositionen nicht mit samplingbasierten Orgeln aufgeführt werden.

### Physical Modelling

Ein anderer Ansatz, Töne zu erzeugen, ist Physical Modelling. Diese Art der Klangerzeugung baut darauf, die physikalischen Effekte der Tonerzeugung nachzubilden. Früher wurde dies durch die Zusammensetzung des zuvor analysierten Klanges entsprechend seiner Zusammensetzung aus einzelnen Schwingungen erreicht. So stellt die Hammondorgel eine frühe Form der additiven Klangsynthese aus Sinuswellen dar. Heutzutage werden die für die Klanger-

---

<sup>5</sup>Definition siehe Soundfont (1996).

zeugung verantwortlichen direkt als Algorithmen modelliert<sup>6</sup>.

Voraussetzung dafür ist, dass bekannt ist, welche physikalischen Vorgänge an der Tonerzeugung beteiligt sind und wie diese modelliert werden können. So hat beispielsweise die Form der Wandung eines Pfeifenkörpers bei bestimmten Materialien einen Einfluss auf die Klangcharakteristik einer Orgelpfeife<sup>7</sup>, auch wenn der Einfluss dieses Effektes nur sehr klein ist.

Problematisch bei Physical Modelling ist der sehr hohe Rechenaufwand. Auch wenn heutzutage mit FPGAs leistungsfähige Simulationen möglich sind<sup>8</sup>, so bleibt das grundlegende Problem, dass eine solche Simulation nur bis zu einer bestimmten Komplexität in Echtzeit berechnet werden kann<sup>9</sup>.

Andererseits bietet Physical Modelling die Möglichkeit, Instrumente außerhalb physikalischer Rahmenbedingungen zu simulieren und somit neue Klänge zu erzeugen. Dies ist möglich, da die Modelle, anhand derer berechnet wird, parametrisiert werden können.

## 1.2.2. Hard- und Software zur Simulation von Orgeln

### Miditzer

Die **Miditzer** ist eine Programm, das eine virtuelle Theaterorgel bereitstellt. Der Name ist ein Kofferwort und leitet sich aus der für ihre Theaterorgeln bekannte Firma Wurlitzer und dem MIDI genannten Standard für den Austausch von musikalischen Steuerinformationen her. Als Basis der Tonerzeugung dient eine gesampelte Theaterorgel die in Form eines Soundfonts bereitgestellt wird. Die weiteren Möglichkeiten, auf die Klangerzeugung Einfluss zu nehmen, sind jedoch stark beschränkt.

### Hauptwerk

Das Programm **Hauptwerk** wurde ab 2004 von Martin Dyde entwickelt. Es stellt die Funktion für die Darstellung eines Spieltisches auf dem Desktop zur Verfügung und bildet die Plattform zur Verarbeitung der Audiodateien. Diese liegen in Form von Samples vor, die zu einem sogenannten Sampleset zusammengefasst sind. Ein Sampleset bildet üblicherweise eine Orgel ab und besteht neben den Samples auch noch aus Graphiken zur Darstellung des virtuellen Spieltisches und weiteren Konfigurationsdateien in denen die Merkmale der Orgel abgebildet sind. Damit kann ein Sampleset als Erweiterung von Soundfonts angesehen werden.

---

<sup>6</sup>Siehe Hagenow (2001), Seite 24 f.

<sup>7</sup>Siehe Bergweiler (2005), Seite 35.

<sup>8</sup>So beispielsweise eine auf FPGAs basierende Simulation der Hammond B3, siehe Kraus (2012).

<sup>9</sup>Siehe dazu auch den Stand von 2001 bei Hagenow (2001), Seite 25.

## GrandOrgue

Das Programm **GrandOrgue** wurde ab 2006 unter dem Namen MyOrgan als freie Alternative zu Hauptwerk entwickelt. Es ist OpenSource und bietet grundlegend die gleiche Funktion wie Hauptwerk, das Spielen von gesampelten Orgeln. Unterschiede betreffen die zusätzlichen Funktionen. Während GrandOrgue aus dem Programm heraus die Möglichkeit bietet, Faltungshall zu benutzen, fehlt Hauptwerk diese Funktion. Im Gegensatz dazu bietet GrandOrgue keine Simulation der Windversorgung der Orgel, sodass Effekte wie Windstößigkeit nicht nachgebildet werden können<sup>10</sup>.

## Aeolus

**Aeolus**, benannt nach dem griechischen Gott der Winde, ist eine freie Software zur Simulation von Orgeln auf Basis von Physical Modelling. Das Programm benutzt eine Mischung aus Sampling und additiver Klangsynthese<sup>11</sup>. Dabei werden die Register zunächst über additive Synthese generiert und als Wavetable abgespeichert. Aus diesen Wavetables wird dann der Klang anhand der gespeicherten Klangparameter generiert. Der dabei erzeugte Klang wird anschließend in einem weiteren Prozessschritt um das Klangverhalten des Raumes, in dem die virtuelle Orgel steht, ergänzt<sup>12</sup>.

## Sakralorgeln

Außerhalb der PC-basierten Lösungen sind seit Jahrzehnten sogenannte **Sakralorgeln** der Standard. Die Bezeichnung weist dabei auf die Nutzung in Kirchen hin und dient der Abgrenzung gegenüber popularmusikalisch orientierten Heim- und E-Orgeln. Im Gehäuse eines freistehenden Spieltisches sind alle elektrischen Komponenten für die Klangerzeugung untergebracht. Sakralorgeln für kleine Räume oder für die Benutzung als Instrument zum Üben in Haus oder Wohnung enthalten meist Stereo-Lautsprecher für Wiedergabe der erzeugten Klänge. Geräte für größere Kirchen und Räume hingegen haben üblicherweise Verstärker integriert und nutzen für die Klangabstrahlung mehrere Stereo-Kanäle.

---

<sup>10</sup>Siehe Kleine (2019).

<sup>11</sup>Siehe Adriaensen (2004), Seite 19.

<sup>12</sup>Siehe Adriaensen (2004), Seite 25.

Die Klangerzeugung geschieht jedoch auch bei diesen Orgeln über die Wiedergabe von Samples. Während frühere Sakralorgeln dabei lediglich verschiedene Dispositionen mit generischen Stilbezeichnungen wie z.B. „romantisch“, „barock“ oder „symphonisch“ boten, ist derzeitiger Stand der Technik die Annäherung an Hauptwerk mit der Möglichkeit, real existierende Orgeln zu spielen<sup>13</sup>.

### **1.3. Problemstellung**

Die Frage, welche Möglichkeiten es gibt, eine Pfeifenorgel zu simulieren, beschäftigt mich schon seit meiner Schulzeit. In einer Besonderen Lernleistung im Rahmen des Abiturs habe ich mich damit beschäftigt, welche Möglichkeiten der damalige Stand der Technik bot. Obwohl seitdem mehr als 10 Jahre vergangen sind gibt es im Hinblick auf die Möglichkeiten der Simulation von Orgeln keine neuen Impulse.

---

<sup>13</sup>Siehe Johannus (2018).

## 2. Technischer und physikalischer Hintergrund

### 2.1. Funktionsweise einer Orgel

#### 2.1.1. Aufbau

Die Pfeifenorgel ist in der Regel ein sehr großes, in Gebäude fest eingebautes Instrument. Besonders die zur Tonerzeugung erforderlichen Pfeifen benötigen sowohl aufgrund ihrer großen Zahl als auch der eigenen Größe sehr viel Platz.

Die Orgel kann grob in vier Teile unterteilt werden:

1. Spieltisch
2. Traktur
3. Windlade
4. Pfeifenwerk

Wichtig für diese Arbeit ist nur die Tonerzeugung in Labialpfeifen. Der generelle Aufbau und die Funktionen der einzelnen Teile sollen dennoch kurz in den nachfolgenden Abschnitten beschrieben werden<sup>1</sup>.

#### 2.1.2. Spieltisch

Die Orgel wird vom Spieltisch (oder je nach Bauart auch Spielschrank) gesteuert. In der Mitte befinden sich die Manuale (von lat. Manus=Hand), damit werden die einzelnen Töne angesteuert. Um die Manuale herum sind die Schalter oder Züge für die Register und Koppeln angebracht, je nach Bauform auch noch weitere Anzeigen.

---

<sup>1</sup>Standardwerk für Aufbau und Funktion von Orgel ist Klotz (1998). Diese Ausführungen folgen im Aufbau der früheren Arbeit Feist (2008).

### 2.1.3. Traktur

Als Traktur bezeichnet man die Verbindungselemente von den Registerzügen und Tasten zu den Pfeifen, um die Orgel zu steuern. Sie kann mechanisch, pneumatisch oder elektrisch ausgeführt sein, wobei auch Mischformen Anwendung finden. Die älteste Form ist die mechanische Traktur. Bei ihr besteht eine direkte Verbindung zwischen Taste und Ventil, sodass der Organist spürt, wie die Ventile geöffnet werden. Bei der pneumatischen und elektrischen Traktur fehlt diese Rückkopplung, da die Ventile entweder mit Luft (pneumatisch) oder von Magneten/Motoren (elektrisch) geöffnet werden.

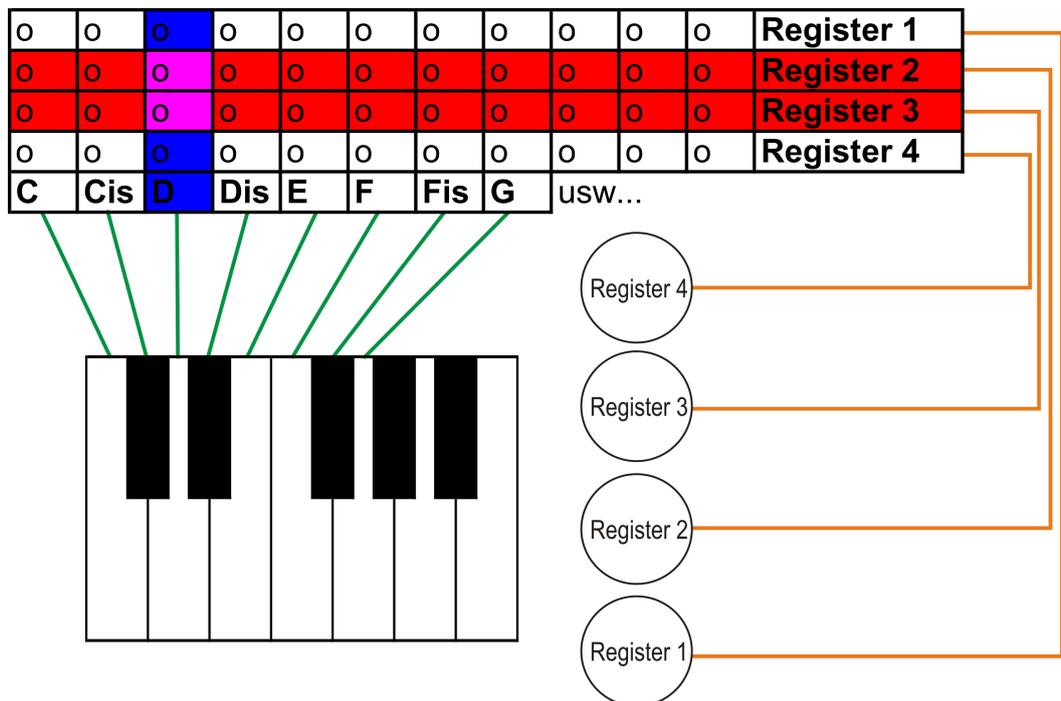


Abbildung 2.1.: Vereinfachte schematische Zeichnung des Prinzips der Windlade in der Draufsicht (blau=gedrückte Taste, rot=gezogenes Register). Jedes „o“ steht für eine Pfeife. Die grünen Striche stellen die Spieltraktur dar, die orangenen Striche die Registertraktur.

### 2.1.4. Windlade

Die Töne werden in den Pfeifen erzeugt, diese sind auf der Windlade aufgestellt. Die Abbildung 2.1 zeigt schematisch das Prinzip der Windlade. In der Skizze sind die Töne in der x-Achse angeordnet, die Register entlang der y-Achse. Wird durch das gezogene Register die Luftzufuhr zu der jeweiligen

Pfeifengruppe freigegeben, kann die Pfeife erklingen, wenn die zugehörnde Taste auf dem Spieltisch gedrückt wird. Sind mehrere Register gezogen, können gleichzeitig mehrere Pfeifen beim Drücken einer Taste erklingen.

In der Abbildung 2.1 sind Register 2 und 3 gezogen (rot markiert), die Taste D ist gedrückt (blau markiert). Die Luft strömt nur zu den Pfeifen, die in der Zeichnung violett markiert sind.

### 2.1.5. Pfeifenwerk

Als Pfeifenwerk werden alle klingenden Pfeifen einer Orgel bezeichnet. Pfeifen gleicher Klangfarbe und Tonhöhe sind zu so genannten Registern zusammengefasst. Üblicherweise sind diese auf der Windlade in einer Reihe angeordnet.

Es gibt unterschiedliche Bauformen von Orgelpfeifen. Man unterscheidet:

- Labialpfeifen (von lat.: labium=Lippen<sup>2</sup>), bestehend aus einem Pfeifenfuß, einem Kern, einem Labium und bei offenen Pfeifen einer Mündung oder bei gedackten Pfeifen einem Deckel<sup>3</sup>.
- Lingualpfeifen (von lat.: lingua=Zunge<sup>4</sup>), bestehend aus einem Pfeifenfuß (Stiefel) mit der Zunge und einem Becher, der auf dem Stiefel aufgesetzt ist.

Die Lingualpfeifen erzeugen einen schnarrenden Ton, die Labialpfeifen einen feineren, flötenartigeren Ton. Während Lingualpfeifen ausschließlich aus Metall gebaut werden, können Labialpfeifen sowohl aus Metall als auch aus Holz hergestellt werden. Ob die Wahl des Materials einen Einfluss auf den Klang hat, ist Gegenstand eines jahrzehntelangen Disputs unter Orgelbauern. Nachgewiesen ist, dass die stehende Luftsäule auch den Pfeifenkörper in Schwingung bringt und die dafür eingesetzte Energie nicht mehr für die Abstrahlung des Klanges zur Verfügung steht<sup>5</sup>. Der Effekt ist jedoch vernachlässigbar mit Blick darauf, dass andere Parameter der Pfeife, wie ihre Form oder ihre Querschnittsfläche den Klang wesentlich stärker beeinflussen..

Das gesamte Pfeifenwerk ist in so genannte Teilwerke unterteilt. In diesen Werken werden mehrere Register zusammengefasst, die dann zusammen auf einer Windlade stehen. Die Zusammenstellung und Benennung der Werke ist von der musikalischen Epoche abhängig, die wichtigsten epochenunabhängigen Werke sind:

---

<sup>2</sup>Hau (2003), Seite 270.

<sup>3</sup>Siehe dazu Zscherpel (2001), Seite 2.

<sup>4</sup>Hau (2003), Seite 282.

<sup>5</sup>Siehe Bergweiler (2005), Seite 96ff.

- das Hauptwerk als das zentrale Teilwerk, es enthält üblicherweise die meisten Register mit einem gut ausgebauten Prinzipalchor, relativ viele Zungenpfeifen und weitere gemischten Stimmen,
- das Pedalwerk, das mit den Füßen gespielt wird, enthält als Bassfundament der Orgel die tiefsten Register,
- das Brustwerk, das sich etwa in Kopfhöhe über dem Spieltisch befindet und
- das Oberwerk, als der höchste Punkt der Orgel.

Entsprechend der Größe einer Orgel sind nicht immer alle aufgeführten Werke vorhanden. Es gibt zudem noch weitere Einrichtungen, den Klang zu variieren. Dazu gehören der Tremulant und die Jalousien des Schwellkastens.

Mit Hilfe des Tremulanten wird der Luftstrom, der zum Zwecke der Klangzeugung in die Pfeifen geleitet wird, periodisch variiert. Dies hat zur Folge, dass sich sowohl die Amplitude als auch die Frequenz des in der Pfeife erzeugten Tones ändern.

Der Schwellkasten ist ein nur an einer Seite durch Jalousien geöffneter Kasten, in dem Pfeifen untergebracht sind. Über einen Hebel oder Tritt am Spieltisch können die Jalousien geöffnet oder geschlossen werden. Sind die Jalousien geschlossen, wird der Ton gedämpft.

## 2.2. Die Physik der Tonerzeugung in Orgelpfeifen

Um eine Pfeife möglichst exakt in der Simulation nachzubilden, ist es notwendig, deren Aufbau zu verstehen. Die Tonerzeugung einer Pfeifenorgel ist ein komplexer Vorgang. Jede Pfeife hat ein eigenes, ihrem Register entsprechendes Obertonspektrum. Einzelne Obertöne werden durch andere Register verstärkt, Aliquotregister färben den Klang durch neue Obertöne.

### 2.2.1. Tonerzeugung in Labialpfeifen

Die Luft strömt durch den Pfeifenfuß und wird vom Kern gegen das Labium, eine dünne Kante, gelenkt. Nach dem Einschwingen pendelt der Luftstrom stetig zwischen der Innen- und der Außenseite des Labiums und regt dadurch die Luftsäule in der Pfeife zum Schwingen an<sup>6</sup>.

---

<sup>6</sup>Vgl. dazu Janke (1997).



Abbildung 2.2.: Einige Pfeifen der Rohrflöte aus der St. Paul's Catholic Church, Danville, Illinois, USA. Quelle: Stauff (2005)

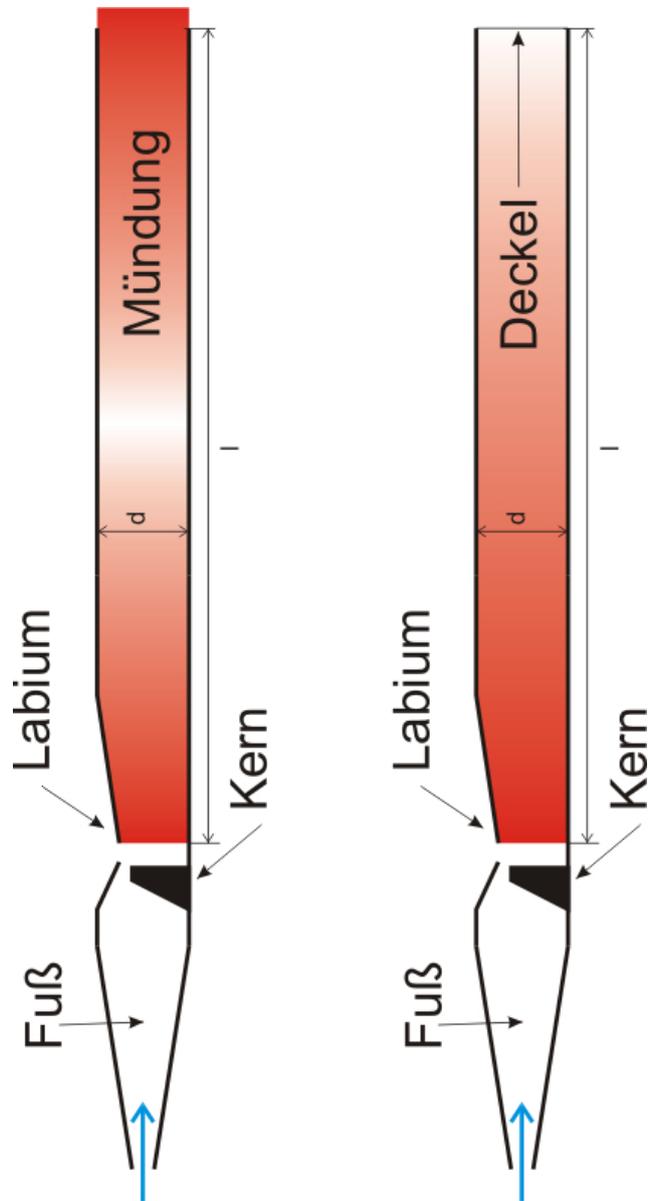


Abbildung 2.3.: Schematischer Aufbau von Labialpfeifen. Links eine offene, rechts eine gedackte Pfeife. Der blaue Pfeil zeigt die Strömungsrichtung an. Die stehende Welle des Grundtones ist rot gekennzeichnet.

Die Frequenz  $f$  mit der diese Luftsäule schwingt ist von zwei Faktoren abhängig:

- Die Pfeifenlänge ( $l$ ), entspricht dem Abstand zwischen Unterkante des Labiums und Oberkante des vollständig geschlossenen Bereiches der Pfeife.
- Die Schallgeschwindigkeit des Stoffes ( $c$ ), in dem sich die Pfeife befindet.

Daraus ergibt sich die normalerweise hinreichend genaue Formel:

$$f = \frac{c}{2l}$$

Da die schwingende Luftsäule jedoch etwas über das Ende der Pfeife hinausragt, muss diese Differenz ( $\Delta l$ ) noch dazu addiert werden. Diese sogenannte Mündungskorrektur ist abhängig vom Durchmesser der Pfeife und wird nach der Formel  $2\Delta l = 2r\alpha$  ( $l$ =Länge der Pfeife,  $r$ =Radius der Pfeife) berechnet<sup>7</sup>. Der Wert für  $\alpha$  muss experimentell ermittelt werden und unterscheidet sich. Zusammengefasst ergibt sich also folgende Formel:

$$f = \frac{c}{2(l + 2r\alpha)}$$

Diese Formel ist, ein entsprechend exakt bestimmter  $\alpha$  vorausgesetzt, genau genug um die Frequenz (bzw. die anderen Parameter) einer Labialpfeife auszurechnen.

Die gedackten Pfeifen sind zum Ende hin geschlossen, deswegen wird die schwingende Luftsäule am Deckel reflektiert. Dies hat zur Folge, dass sich die Frequenz, mit der die Luftsäule schwingt, im Verhältnis zu einer gleich langen offenen Pfeife, halbiert. Zudem kann die Mündungskorrektur vernachlässigt werden<sup>8</sup>. Für gedackte Labialpfeifen gilt somit folgende Formel:

$$f = \frac{c}{4l}$$

In der Abbildung 2.3 ist neben dem Aufbau auch die Schwingung des Grundtones verdeutlicht. Je intensiver das Rot, desto größer ist die maximale Amplitude der stehenden Welle an dieser Stelle. Der Schwingungsknoten (weiß) befindet sich bei der offenen Pfeife nahe der Mitte, bei der gedackten Pfeife am Deckel.

Neben dem Grundton sind für den Klang der Orgelpfeife auch die Obertöne, also die ganzzahligen Vielfachen der Frequenz des Grundtones von Bedeutung. Durch unterschiedliche Zusammensetzung der Obertöne ergeben sich unterschiedliche Klangfarben. Die Frequenzanalyse in den Abbildungen 2.4 und 2.5

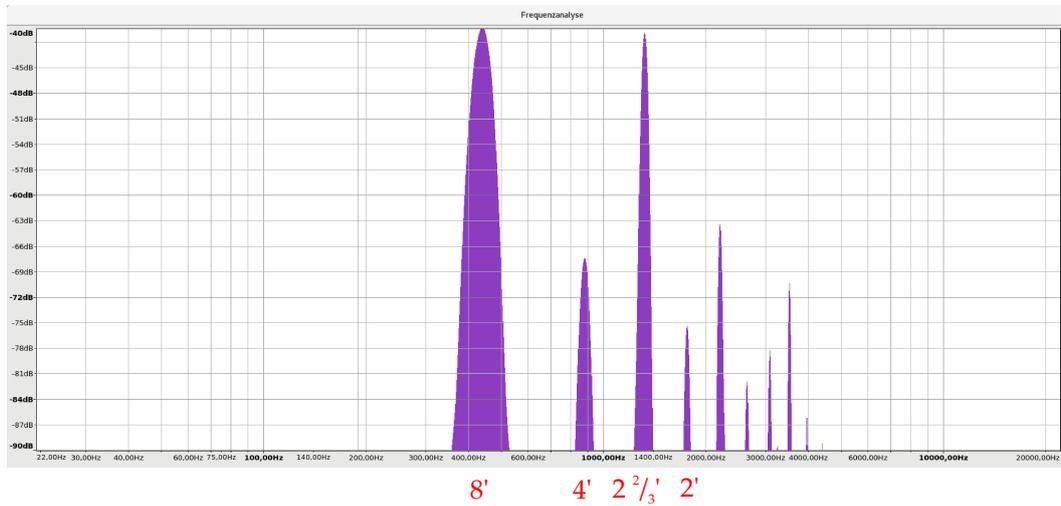


Abbildung 2.4.: Frequenzspektrum einer Prinzipalflöte auf dem Ton a1.

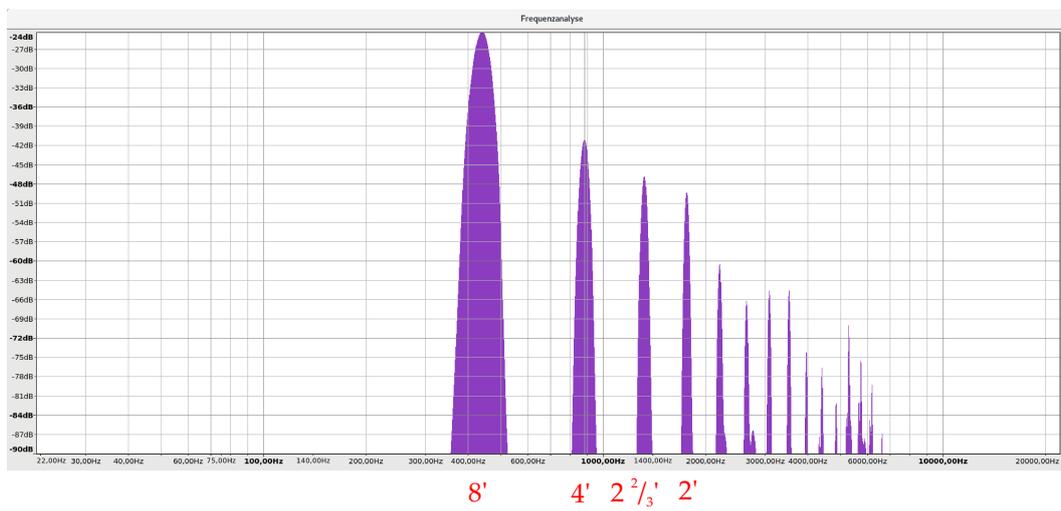


Abbildung 2.5.: Frequenzspektrum einer Rohrflöte auf dem Ton a1.

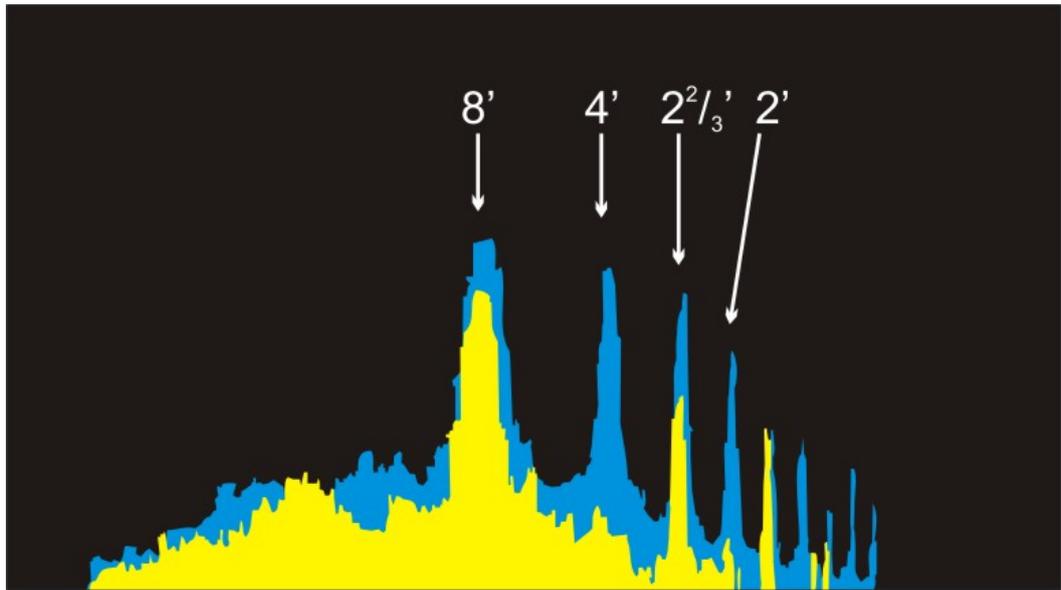


Abbildung 2.6.: Vergleich der Frequenzspektren auf dem Ton a1. Prinzipal: blau, Rohrflöte: gelb. Der Prinzipal ist obertonreicher, deutlich sind die blauen Spitzen zu sehen, die bei der Rohrflöte fehlen.

verdeutlicht dies. Während bei einer Prinzipalpfeife alle Obertöne vorhanden sind, fehlen bei einer Rohrflöte die ungeraden Obertöne.

Erzeugt wird die Schwingung in der Pfeife dadurch, dass die aus der Kernspalte gegen das Labium geblasene Luft periodisch zwischen Pfeifeninnerem und -äußerem pendelt. Dieses Pendeln regt die Luftsäule zum Schwingen an und erzeugt vor und in der Pfeife einen Wirbel. Bei Versuchen an der Fachhochschule Emden wurden 1997 diese Strömungen bildlich festgehalten<sup>9</sup>. Dazu wurde eine Holzpfeife im Bereich des Pfeifenmundes mit gläsernen Seitenwänden ausgestattet und mit Hilfe eines Lasers durch diese Wände hindurch ein Bild der Strömung erzeugt. Abbildung 2.7 zeigt eines dieser Bilder. Links ist der Bereich vor der Pfeife, rechts der Pfeifenkörper. Es ist deutlich zu sehen, dass die Strömung einen Wirbel erzeugt. Das Schwingungsverhalten ist in Abbildung 2.8 zu sehen. Hier wird deutlich, dass die Luftströmung zwischen Innerem und Äußerem pendelt. Im Rahmen dieser Untersuchungen wurden auch Messungen durchgeführt um zu den Bildern Messwerte zu erhalten. Eine Visualisierung dieser Messwerte ist in Abbildung 2.9 abgedruckt. Deutlich ist auch in den Messwerten der Wirbel zu sehen.

<sup>7</sup>Vgl. Zscherpel (2001), Seite 3.

<sup>8</sup>Vgl. Zscherpel (2001), Seite 3.

<sup>9</sup>Siehe Janke (1997).

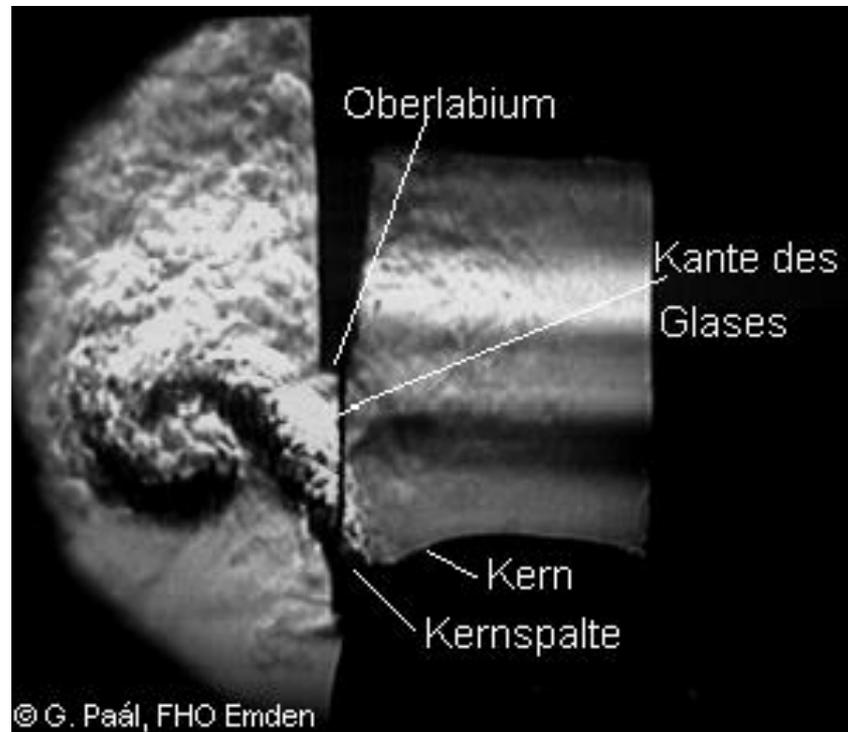


Abbildung 2.7.: Visualisierte Strömung am Pfeifenmund. Bild aus Janke (1997)

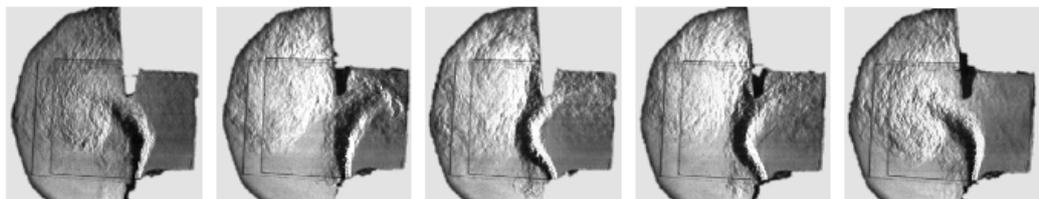


Abbildung 2.8.: Bildfolge der visualisierten Strömung am Pfeifenmund. Bild aus Janke (1997)

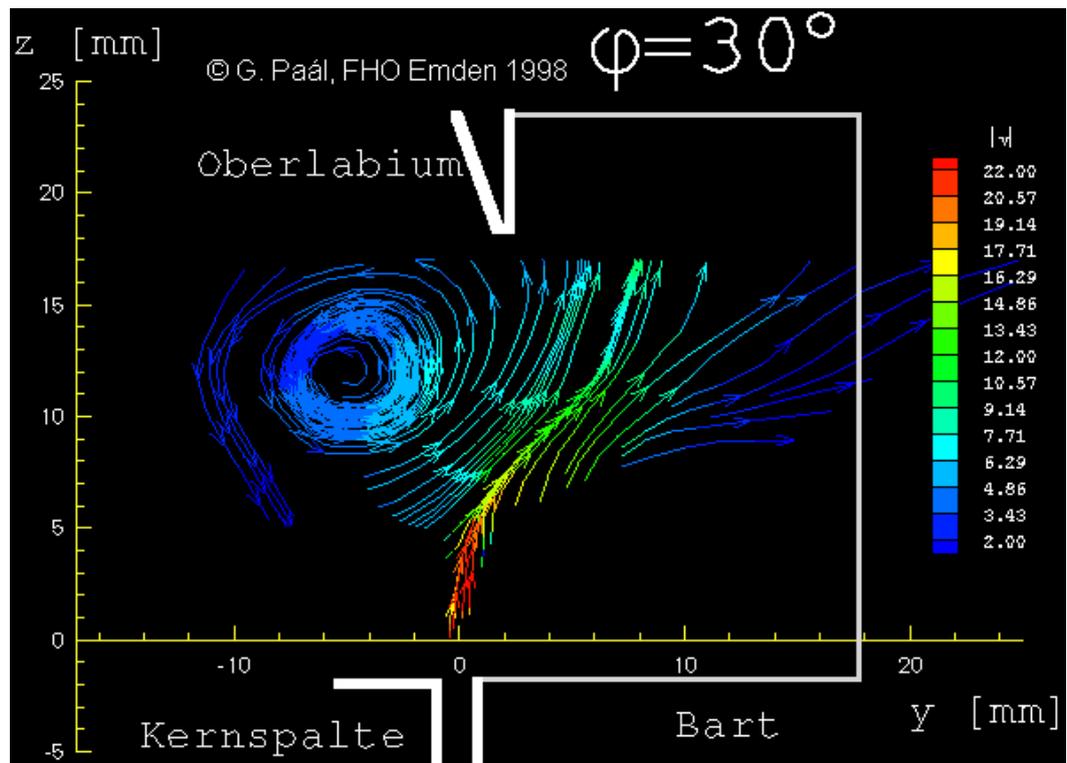


Abbildung 2.9.: Visualisierte Messwerte der Strömung am Pfeifenmund. Bild aus Janke (1997)

Bis eine Pfeife stabil schwingt benötigt sie einige Zeit, diese Zeit ist der sogenannte Einschwingvorgang. Wird eine Taste losgelassen und somit die Pfeife nicht mehr angeblasen klingt sie noch kurz nach. Je nach Register sind Einschwingen und Nachklingen unterschiedlich ausgeprägt und werden auch entsprechend intoniert. So ist ein ausgeprägter Einschwingvorgang bei Prinzipalpfeifen generell unerwünscht, während ein sogenanntes „Spucken“ bei gedackten Pfeifen charakteristisch ist. Auch ein kurzer, perkussiver Ton beim Nachklingen ist charakteristisch für Register wie Gedackt oder Rohrflöte.

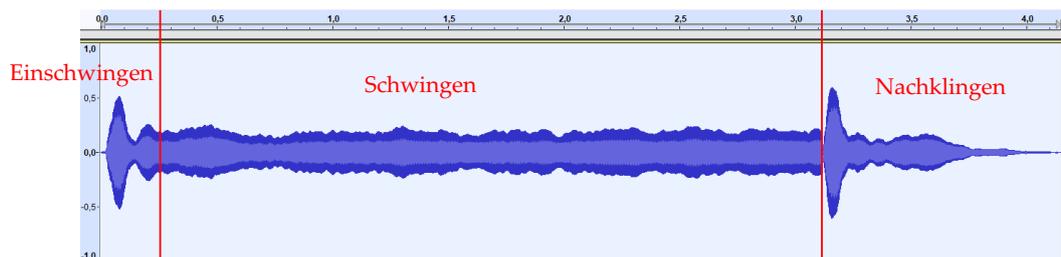


Abbildung 2.10.: Graphische Darstellung des Tons einer Rohrflöte auf dem Ton a1.

In Abbildung 2.10 ist diese Aufteilung sichtbar und bezeichnet. Der charakteristische, leicht perkussive Einschwingvorgang und das genauso eingeleitete Nachklingen sind gut sichtbar. Es ist jedoch zu beachten, dass der Einschwingvorgang erst dann beendet ist, wenn die Pfeife stabil schwingt. Die in der Abbildung zu sehende, leicht unregelmäßige Schwankung in der Amplitude ist auf das Alter der Orgel zurückzuführen und im Allgemeinen als lebendiger Klang gewünscht. Betrachtet man die Visualisierung des Einschwingvorganges in Abbildung 2.11 so wird deutlich, wie die Wellenform des Einschwingvorganges entsteht.

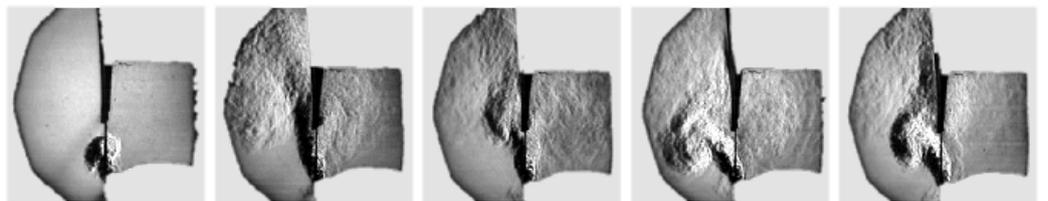


Abbildung 2.11.: Bildfolge des Einströmvorgangs am Pfeifenmund. Bild aus Janke (1997)

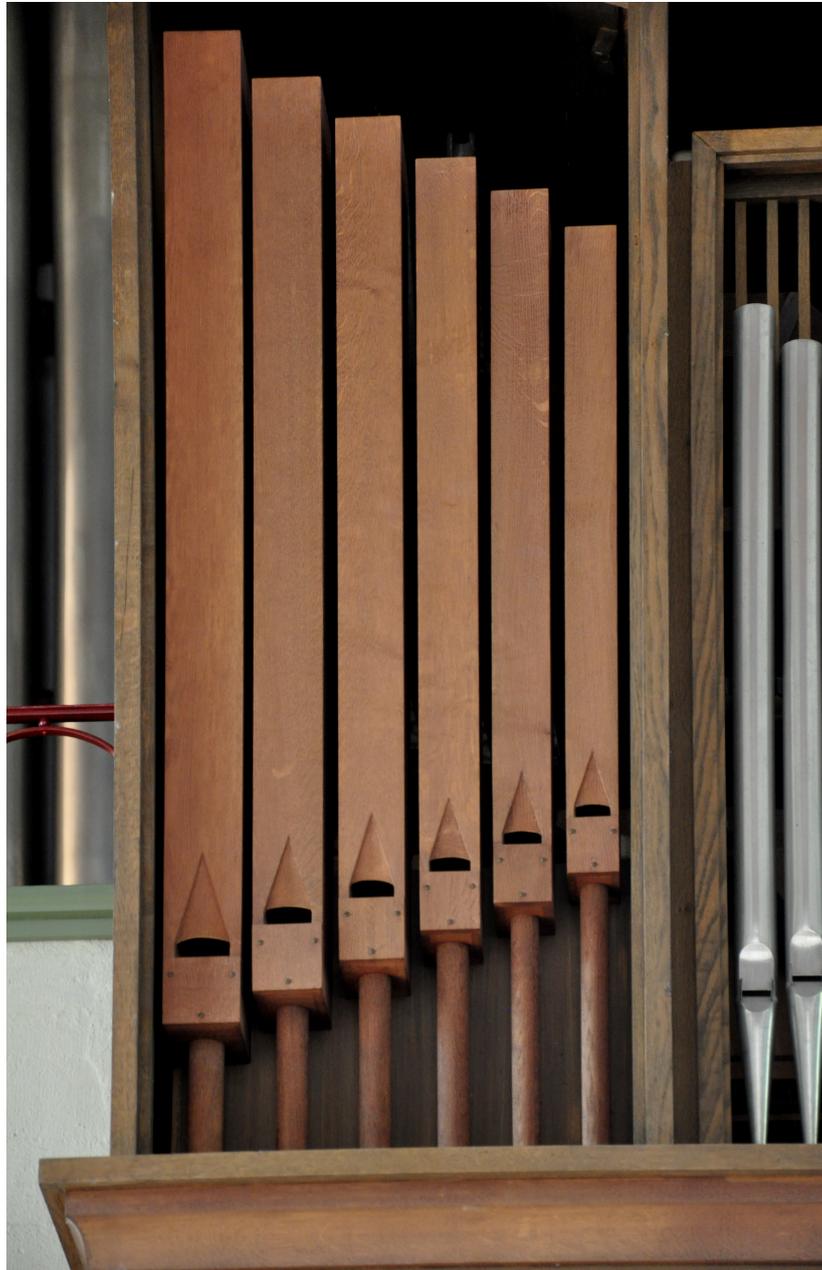


Abbildung 2.12.: Detail vom Prospekt der Orgel der Christuskirche Stuttgart mit Holzpfifen. Quelle: Praefcke (2014)

### 2.2.2. Abstraktion für die Simulation

In der Simulation sollen zunächst nicht alle Aspekte simuliert werden. Zudem soll die Simulation optimiert werden, sodass möglichst wenig Rechenaufwand entsteht. Da in der Simulation ausschließlich Luftbewegungen simuliert werden sollen, kommen nur Labialpfeifen für die Simulation in Betracht, daher werden Lingualpfeifen nicht weiter betrachtet. Die wichtigste Vereinfachung nach der Reduktion der möglichen Register ist dabei der Schritt von einer dreidimensionalen Pfeife zu einer zweidimensionalen Simulation.

Wie im Abschnitt 2.1.5 beschrieben hat die Wahl des Materials nur einen sehr geringen Einfluss auf den Klang einer Pfeife der für die Simulation vernachlässigt werden kann. Um die Fertigung zu vereinfachen werden Metallpfeifen jedoch mit rundem Querschnitt, Holzpfeifen mit quadratischem Querschnitt gefertigt. Vergleichend zu den runden Pfeifen in Abbildung 2.2 ist die rechteckige Form in Abbildung 2.12 deutlich sichtbar. Innerhalb eines Orgelregisters können sowohl Metall- als auch Holzpfeifen verbaut sein solange die Querschnittsfläche gleich bleibt<sup>10</sup>. Dies führt zu der These dass die dritte Dimension für die Simulation vernachlässigbar ist. Die von Janke (1997) veröffentlichten Bilder zeigen deutlich die Strömung im Querschnitt und stützen damit diese These. Die dort untersuchten Unterschiede bei Pfeifen mit und ohne Kernstichen<sup>11</sup> zeigen, dass Kernstiche wie auch Pfeifenbärte<sup>12</sup> den Einschwingvorgang zwar beeinflussen, jedoch für ein Schwingen der Pfeife nicht nötig sind.

Für die Simulation im Rahmen der Forschung kann daher die Orgelpfeife zweidimensional vereinfacht werden. Nach erfolgreicher Durchführung der zweidimensionalen Simulation kann anschließend eine dreidimensionale Simulation untersucht werden in der die Einflüsse von Kernstichen und Bärten untersucht wird.

## 2.3. Technischer Hintergrund der Simulation

Der Duden definiert **simulieren**, als Verb zu Simulation, wie folgt:<sup>13</sup>

Sachverhalte, Vorgänge (mit technischen, naturwissenschaftlichen Mitteln) modellhaft zu Übungs-, Erkenntniszwecken nachbilden, wirklichkeitsgetreu nachahmen

---

<sup>10</sup>Vgl. Brüner (2002)

<sup>11</sup>Eine Aufweitung des Kernspaltes an einigen Stellen um den Einschwingvorgang der Pfeife und damit den Klang zu verändern

<sup>12</sup>Kleine Plättchen an den Rändern des Labiums um die strömende Luft zu leiten

<sup>13</sup>Siehe Duden (2017), Stichwort **simulieren**.

Um den Vorgang der Luftströmung in einer Orgelpfeife nachzubilden bedarf es zunächst dem Verständnis, wie diese Luftströmung modelliert werden kann. Das zentrale Prinzip diese Modellierung ist die physikalische Erkenntnis, dass sich u.a. Masse, Energie und Impuls in bestimmten physikalischen Prozessen nicht ändern<sup>14</sup>. Diese Erkenntnis führt zu den sogenannten Erhaltungsgleichungen der Strömungsmechanik, deren Berechnung die Grundlage der numerischen Strömungsberechnung bildet.

### 2.3.1. Hintergrund Numerische Strömungsberechnung

Für die numerische Strömungsberechnung sind insgesamt fünf Erhaltungsgleichungen relevant<sup>15</sup>:

- Massenerhaltung
- Impulserhaltung in x-Richtung
- Impulserhaltung in y-Richtung
- Impulserhaltung in z-Richtung
- Energieerhaltung

Die verwendeten Gleichungen gehen zurück auf den französischen Physiker und Ingenieur **Claude Louis Marie Henrie Navier** und den britischen Mathematiker und Physiker **Sir George Gabriel Stokes** und werden daher Navier-Stokes-Gleichungen genannt. Für einfachste Anwendungen lassen sich diese Gleichungen theoretisch lösen. Für alle anderen Anwendungen müssen Näherungen gefunden werden die dem jeweiligen Zweck angepasst sind.

Eine Näherung ist dabei die Berechnung in Zeitschritten, während in der Physik die Umwelt als zeitkontinuierlich angenommen wird<sup>16</sup>. Dadurch werden alle Simulationen iterativ berechnet und es ist nicht nötig, die Gleichungen exakt zu berechnen. Das bedeutet auch, dass es möglich ist, in der Simulation den Zeitschritt selbst zu definieren und somit ein Optimum zu finden zwischen der Genauigkeit der Simulation und der für die Berechnung dieser Simulation benötigten Zeit.

Eine weitere Näherung ist die Reduktion auf wesentlich Teile der Berechnung. So ist besonders bei turbulenten Strömungen der Rechenaufwand sehr hoch, da auch kleinste Turbulenzen exakt berechnet werden müssen<sup>17</sup>. Um

---

<sup>14</sup>Siehe Lecheler (2008), Seite 9 ff.

<sup>15</sup>Siehe Lecheler (2008), Seite 9 ff.

<sup>16</sup>Auch wenn die Quantenphysik mit der Planck-Zeit eine kleinste Zeiteinheit definiert, siehe Lesch (2008), Abschnitt „Als der Zeitpfeil das Weite suchte“.

<sup>17</sup>Siehe Lecheler (2008), Seite 32.

dieses Problem zu lösen werden hochfrequente Turbulenzen nicht mehr direkt berechnet, sondern über Modelle. Bei diesen sogenannten **Reynoldsgemittelten** Navier-Stokes-Gleichungen hängt das Ergebnis dann jedoch wesentlich vom verwendeten Turbulenzmodell ab.

Um diese Modelle richtig einsetzen zu können ist es wichtig, die Art der Strömung zu kennen. Es wird unterschieden zwischen<sup>18</sup>

**laminarer Strömung** bei der sich zwei Strömungen mit unterschiedlich Geschwindigkeiten nicht vermischen, also keine sichtbaren Turbulenzen sichtbar sind

**turbulenter Strömung** bei der sich zwei Strömungen mit unterschiedlichen Geschwindigkeiten in Form von Wirbeln vermischen.

Zur Unterscheidung dient dabei die nach dem britischen Physiker Osborne Reynolds benannte **Reynolds-Zahl**. Diese Zahl ist abhängig von

$\rho$  Dichte des Stoffes in dem die Strömung ist

$v$  Geschwindigkeit der Strömung

$d$  Charakteristische Länge des um- oder durchströmten Objektes

$\eta$  Dynamische Viskosität des Stoffes in dem die Strömung ist

und wird nach der Formel

$$Re = \frac{\rho v d}{\eta} = \frac{v d}{\nu}$$

berechnet.

Anhand der Reynolds-Zahl kann dann das am Besten geeignete Turbulenzmodell ermittelt werden.

### 2.3.2. Der Solver rhoPimple

Bei der Auswahl des Solver fiel die Wahl auf den Solver **rhoPimple**. Im OpenFOAM User Guide wird der Solver wie folgt beschrieben<sup>19</sup>:

Transient solver for turbulent flow of compressible fluids for HVAC [Heating, Ventilation and Air Conditioning] and similar applications, with optional mesh motion and mesh topology changes

Da im vorliegenden Fall eine Luftströmung simuliert werden sollte ist dieser Solver geeignet um die Berechnung durchzuführen.

Der Solver basiert auf den beiden Algorithmen **PISO**<sup>20</sup> und **SIMPLE**<sup>21</sup>.

<sup>18</sup>Siehe Versteeg (2007), Seite 40 ff.

<sup>19</sup>Siehe OpenCFD (2018) Kapitel A.1 Standard Solvers.

<sup>20</sup>Pressure-Implicit with Splitting of Operators

<sup>21</sup>Semi-Implicit Method for Pressure Linked Equations

## SIMPLE

Der SIMPLE-Algorithmus bietet eine einfache Möglichkeit zum Lösen der Strömungsgleichungen und arbeitet iterativ folgende Schritte<sup>22</sup> ab:

1. Löse die diskreten Impulsgleichungen
2. Löse die Gleichung für die Korrektur des Drucks
3. Korrigiere Druck und Geschwindigkeit
4. Löse alle anderen diskreten Transportgleichungen
5. Wenn die Berechnung konvergiert, beende. Sonst beginne wieder bei 1.

## PISO

Der PISO-Algorithmus erweitert den SIMPLE-Algorithmus um einige zusätzliche Schritte. So werden zuerst die Schritte 1 bis 3 des SIMPLE-Algorithmus ausgeführt sodass der gesamte Algorithmus wie folgt abläuft<sup>23</sup>:

1. Löse die diskreten Impulsgleichungen
2. Löse die Gleichung für die Korrektur des Drucks
3. Korrigiere Druck und Geschwindigkeit
4. Löse die zweite Gleichung für die Korrektur des Drucks
5. Korrigiere Druck und Geschwindigkeit
6. Löse alle anderen diskreten Transportgleichungen
7. Wenn die Berechnung konvergiert, beende. Sonst beginne wieder bei 1.

## Kombination der Algorithmen

Die Arbeitsweise des Solvers rhoPimple verdeutlicht ein Blick in den zugehörigen Quelltext, der in Listing A.1 abgedruckt ist. Das Grundprinzip ist dabei die mehrfache Ausführung des PISO-Algorithmus pro Zeitschritt.

Die Schleife von Zeile 86 bis 190 beinhaltet die Berechnung, die jeweils pro Zeitschritt ausgeführt wird. Die mehrfache Ausführung des PISO-Algorithmus wird in Zeile 121 durchgeführt. Der Quelltext dazu ist in Listing A.2 zu finden. Die Anzahl der Iterationen pro Zeitschritt wird durch den Parameter

---

<sup>22</sup>Siehe Versteeg (2007), Seite 190 ff.

<sup>23</sup>Siehe Versteeg (2007), Seite 193 ff.

`nCorrPIMPLE_` festgelegt, der in Zeile 198 ausgewertet wird und damit eine Abbruchbedingung für die Schleife aus dem dem `rhoPimpleFoam-Solver` darstellt.

## 2.4. Ausgabe der Ergebnisse

Da das Ziel der Arbeit eine Ausgabe der Simulation auf dem Audioausgang ist soll hier erläutert werden, was dabei zu beachten ist. Die Samplingrate ist nicht nur durch die Ausgabe relevant sondern auch für die Simulation in direktem Zusammenhang mit der Frequenz der Pfeife.

### 2.4.1. Technischer Hintergrund Audioverarbeitung

Schallwellen sind ein kontinuierliches Signal das zur Verarbeitung zunächst in ein diskretes Signal umgewandelt werden muss. Dabei ist neben der Qualität der verwendeten Analog-Digital-Wandler auch wichtig, in welcher Auflösung die Signale weiterverarbeitet werden. Das betrifft sowohl die zeitliche Auflösung als auch die Auflösung der Amplitude. Diese Parameter sind auch für die Simulation relevant, denn die Simulation erzeugt direkt ein digitales Signal, das jedoch durch die Anzahl der Simulationsschritte in seiner zeitlichen Auflösung und durch die Genauigkeit der Berechnung in seiner Wertauflösung festgelegt ist.

#### Grundlagen digitaler Audioverarbeitung

Um ein analoges Signal in ein digitales Signal umzuwandeln muss dieses abgetastet und quantisiert werden. Beide Prinzipien sind in der Abbildung 2.13 dargestellt. Gemeinsam bilden sie die Puls-Code-Modulation.

Bei der Abtastung wird zunächst das zeitkontinuierliche Signal in ein zeitdiskretes Signal umgewandelt. Theoretisch wird dies durchgeführt indem das Eingangssignal mit dem Dirac-Kamm multipliziert wird. Ein Dirac-Kamm ist eine Folge von Dirac-Stößen, dabei befinden sich diese Dirac-Stöße genau an den Zeitpunkten, an denen abgetastet werden soll. In Abbildung 2.13 ist das Ergebnis dieser Multiplikation als PAM (Puls-Amplituden-Modulation) eingezeichnet.

In der Praxis kann die Abtastung nicht auf diese Weise durchgeführt werden, da das Dirac-Signal ein theoretisches, ideales Signal ist. Stattdessen stehen zwei Verfahren zur Verfügung die eine Messung über einen kurzen Zeitraum durchführen. Bei einer Messung mit einer Sample-and-Hold-Schaltung wird durch einen Kondensator das Signal zum Zeitpunkt der Messung kurzzeitig auf einem konstanten Wert gehalten. Diese Zeit reicht dann für eine Messung

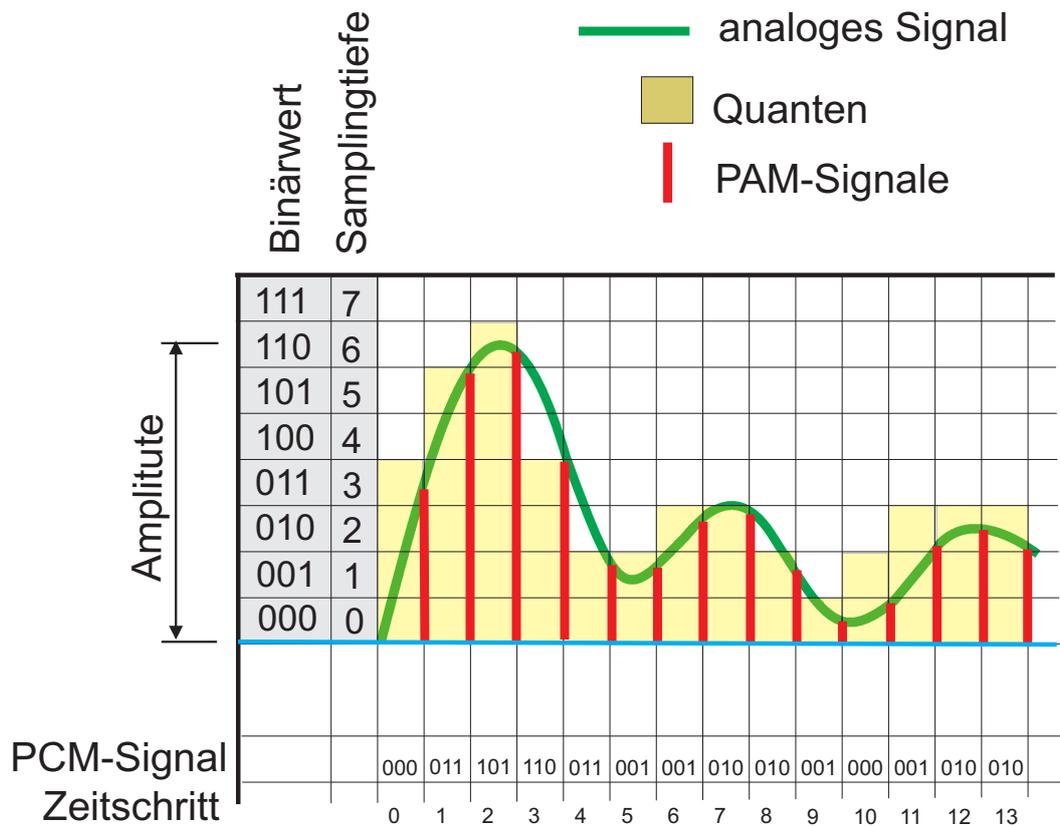


Abbildung 2.13.: Darstellung des Prinzips der Puls-Code-Modulation und der Quantisierung

aus. Alternativ dazu wird die Messung ohne diese Schaltung über einen kurzen Zeitraum um den Messpunkt gemessen und anschließend daraus der Wert ermittelt.

Nach der Abtastung muss das nun zeitdiskrete Signal auch in ein wertdiskretes Signal umgewandelt werden. Dazu muss der Wert an der Abtaststelle auf ein Raster abzählbar vieler möglicher Werte abgebildet werden. In der Praxis wird dazu jedoch eine endliche Menge genutzt die je nach Anwendungsfall zwischen 8 und 32 bit liegt. In Abbildung 2.13 erfolgt diese Abbildung, indem stets auf die nächsthöhere Ganzzahl abgebildet wird. Der ermittelte Wert ist blassgelb dargestellt und unter dem Diagramm eingetragen.

Dabei ist jedoch zu beachten, dass es durch eine solche Umwandlung zu Differenzen zwischen dem abgetasteten Signal und dem Ausgangssignal kommt und zwar immer dann, wenn das Ausgangssignal nicht genau einem Wert aus der Menge möglicher Werte entspricht. Dieser sogenannte Quantifizierungsfehler ist abhängig von der verwendeten Quantifizierungsmethode. Auch dies ist in Abbildung 2.13 ersichtlich. So wird bei Zeitschritt 2 der quantifizierte Wert

höher als die Kurve, bei den Zeitschritten 6 und 7 bleibt der quantifizierte Wert gleich, obwohl sich die Kurve verändert.

### Auflösung der Zeit - Abtasttheorem

Die Unterteilung des zeitkontinuierlichen Signals in ein zeitdiskretes Signal führt dazu, dass nur Schwingungen bis zu einer bestimmten Frequenz erfasst werden können. Das Problem ist in Abbildung 2.14 dargestellt. Sowohl die rote, als auch die blaue Schwingung liefern an den ganzzahligen Abtastpunkten den gleichen Wert. Dies geschieht dann, wenn die Frequenz einer Schwingung größer ist als die doppelte Abtastfrequenz. Um ein solches Signal eindeutig rekonstruieren zu können muss die Abtastfrequenz doppelt so groß sein wie die Frequenz der Es können also nur Schwingungen erfasst werden, deren Frequenz maximal halb so groß ist wie die Samplefrequenz:

$$f_s \geq 2f_{max}$$

Diese Formel wird **Nyquist-Shannon-Abtasttheorem** genannt.

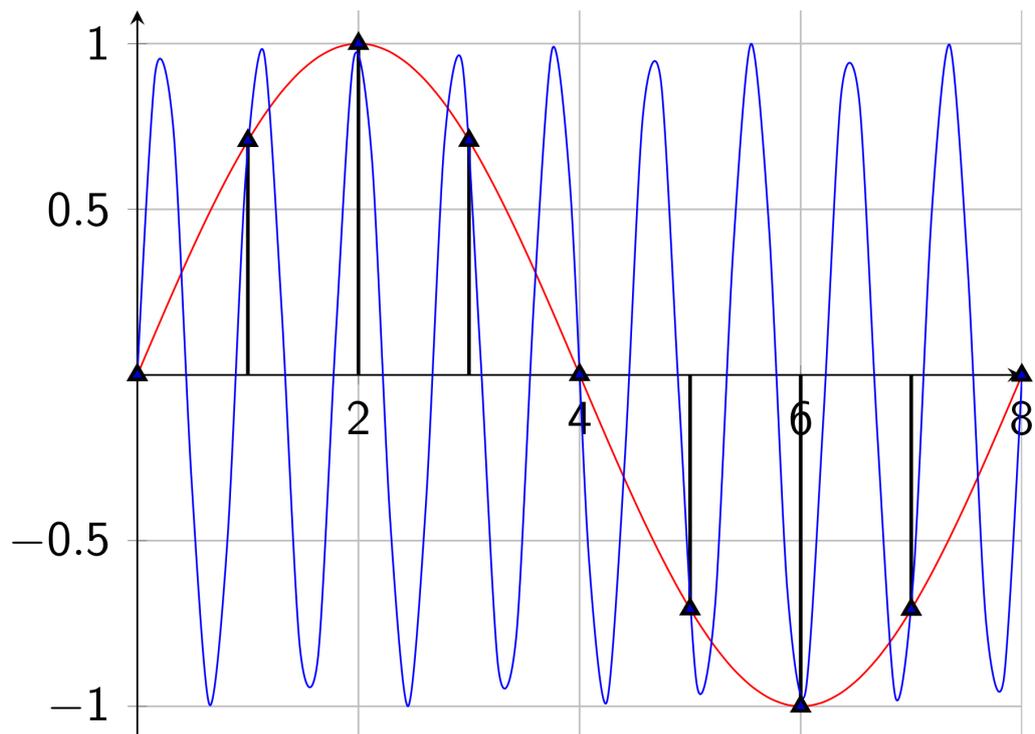


Abbildung 2.14.: Darstellung zweier Signale mit Abtastung. Abbildung aus Hlawitschka (2015)

In der Theorie bedeutet eine Abtastung mit der doppelten Samplefrequenz jedoch einen unendlich hohen Aufwand zur Rekonstruktion des ursprünglichen Signals. Daher werden in der Praxis höhere Abtastraten verwendet und zu hohe Frequenzen mit einem Tiefpassfilter herausgefiltert. Es hat sich dafür die Formel

$$f_{\text{sampling}} \geq 2,2f_{\text{max}}$$

eingebürgert<sup>24</sup>. Um zeitnah Frequenzen bis 20 kHz rekonstruieren zu können wird daher bei Audio-CDs eine Abtastrate von 44,1 kHz verwendet. Zu beachten ist hierbei, dass dabei nicht nur die Grundfrequenz eines Tones wichtig ist, sondern auch die den Klang formenden Obertöne.

### Auflösung der Amplitude

Neben der zeitlichen Auflösung ist für die Qualität des Signals auch die Auflösung der Quanten von Bedeutung. Dies ist besonders dann wichtig, wenn aufgenommene Signale nicht den vollen Umfang der Auflösung ausnutzen. Die Gründe dafür sind üblicherweise die Mischung lauter und leiser Töne, bei denen die lauten Töne nicht übersteuern dürfen. In Orgeln tritt dieses Problem ganz selbstverständlich auf, denn es gibt sowohl leise Register wie z.B. Gedackt oder Aeoline als auch laute Register wie die Prinzipale oder die Chormaden. Dazu kommt, dass bei simulierten Orgeln auch Nebengeräusche wie das Betriebsgeräusch des Orgelmotors wiedergegeben werden um den Realismus zu steigern. Diese Geräusche sind noch leiser als die leisesten Orgelregister<sup>25</sup>. Bei der Zusammenstellung eines Samplesets hat es sich dennoch als praktikabel erwiesen, das Lautstärkeverhältnis der einzelnen Register untereinander beizubehalten, sodass leise Register nur einen Bruchteil der zur Verfügung stehenden Auflösung nutzen. Dies ist kein Nachteil im Rahmen der samplingbasierten Simulation, denn dieses Vorgehen reduziert die benötigten Rechenschritte beim Zusammenfügen des Klangs.

Nichtsdestoweniger muss beachtet werden, welche Effekte eine zu geringe Auflösung mit sich bringt. In Abbildung 2.16 ist der Ton einer Prinzipalpfeife auf dem Ton a1 graphisch dargestellt. Es ist ersichtlich, dass der Ton nur ca.  $\frac{1}{20}$  der zur Verfügung stehenden Auflösung ausnutzt. Bei 16 bit entspricht dies ca. 3276 verschiedenen Werten. Reduziert man die Auflösung auf 8 bit so ergeben sich bei sonst gleicher Welle und Amplitude nur noch 13 verschiedene Werte. Bei extremer Vergrößerung der Welle erscheint diese wie in Abbildung 2.17 abgebildet treppenförmig. Die dadurch entstehenden Artefakte sind auch in der Frequenzanalyse gut sichtbar. Während im Spektrum der Pfeife mit 16 bit Auflösung (siehe Abbildung 2.4) nur die Grundfrequenz und die Ober-

---

<sup>24</sup>Siehe Hlawitschka (2015), Seite 83.

<sup>25</sup>Siehe Zurek (2013).

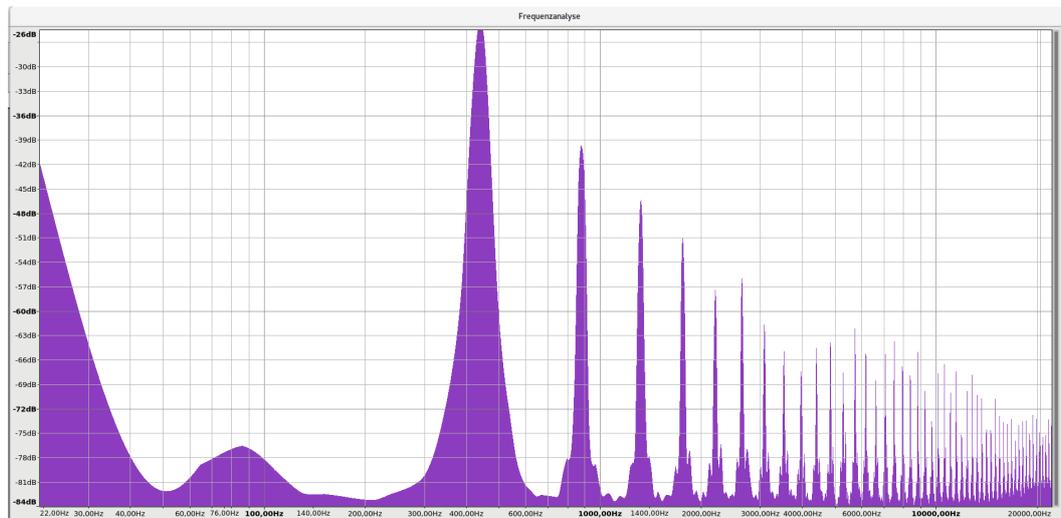


Abbildung 2.15.: Frequenzspektrum einer Prinzipalpfeife auf dem Ton a1 wenn sie mit 8 bit Auflösung verarbeitet wurde.



Abbildung 2.16.: Ansicht der Waveform einer Prinzipalpfeife auf dem Ton a1.

töne sichtbar sind zeigt das Spektrum der mit 8 bit aufgelösten Pfeife (siehe Abbildung 2.15) ein anderes Bild. Diese Artefakte sind nicht nur sichtbar, sondern auch gut hörbar.

Eine Auflösung von 16 bit hat sich schon lange als Mindeststandard etabliert. Geringere Auflösungen werden nur noch dort genutzt, wo die Datenmengen möglichst klein gehalten werden müssen, beispielsweise bei Voice over IP. Das hier gezeigte Beispiel mit 8 bit Auflösung stellt deshalb für den Bereich der professionellen Audioverarbeitung einen Extremfall dar. Bei aktuellen samplingbasierten Orgelsimulationen liegt der Fokus stattdessen bei den Unterschieden zwischen 16 bit und 24 bit. Das Grundproblem bleibt jedoch, ist aber nicht mehr so extrem wie bei diesem illustrativen Beispiel.

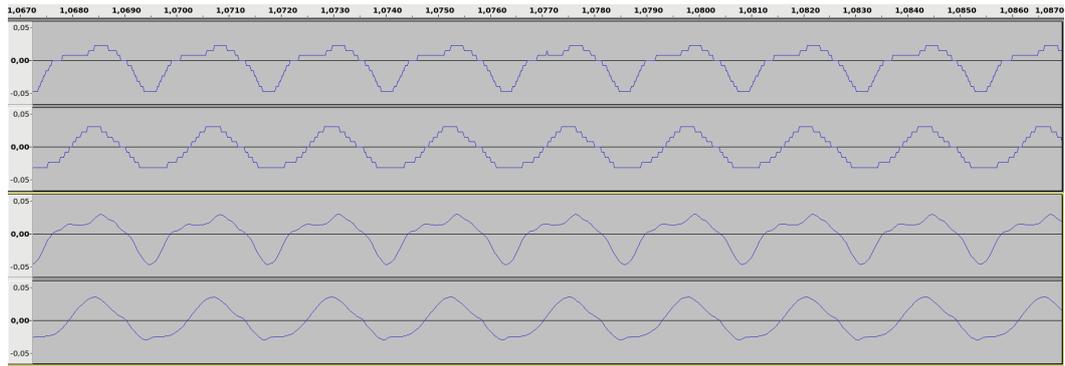


Abbildung 2.17.: Vergleich der Wellenform einer Prinzipialpfeife zwischen 8 bit und 16 bit Auflösung.

## Formate

Wenn das Audiosignal umgewandelt wurde muss es noch als Datei gespeichert werden. Dafür stehen eine Reihe von Dateiformaten zur Verfügung die unterschiedliche Eigenschaften und damit verschiedene Vor- und Nachteile mit sich bringen. Diese lassen sich insbesondere danach unterscheiden, ob sie verlustbehaftet sind oder nicht.

Heutzutage werden vor allem folgende Audioformate benutzt:

**PCM (WAV)** Im PCM-Format (Microsoft Pulse Code Modulation) werden die Daten unkomprimiert in eine auf Microsoft-RIFF<sup>26</sup> aufbauende WAV-Datei gespeichert.

**AIFF** steht für **Audio Interchange File Format** und basiert ebenso auf PCM, wurde jedoch durch Apple standardisiert.

**FLAC** steht für **Free Lossless Audio Codec** und ist ein freier Codec zur verlustfreien Speicherung von Audiodaten. Im Gegensatz zu PCM oder AIFF werden die Daten komprimiert.

**MP3** steht für **MPEG-1 Audio Layer III** und ist ein Codec zur verlustbehafteten Speicherung von Audiodaten. Der Algorithmus nutzt aus, dass das menschliche Gehör nicht exakt arbeitet. Beispielsweise werden zwei verschiedene Frequenzen erst dann als unterschiedliche Töne wahrgenommen, wenn sie sich um einen gewissen Mindestabstand unterscheiden. Der Algorithmus entfernt dann Töne, die zu nah beieinander liegen und komprimiert somit das Audiosignal verlustbehaftet.

<sup>26</sup>Ressource Interchange Format.

**AAC** steht für **Advanced Audio Coding** und ist ein Codec zur verlustbehafteten Speicherung von Audiodaten. Der Codec wurde als Nachfolger von MP3 entwickelt.

**OGG-Vorbis** ist ein freie Codec zur verlustbehafteten Speicherung von Audiodaten. Der Algorithmus transformiert das Signal zunächst und komprimiert es anschließend auf Basis eines Codebuchs.

**WMA** steht für **Windows Media Audio** und ist ein proprietärer Codec zur verlustbehafteten Speicherung von Audiodaten. Der Codec wurde von Microsoft entwickelt und basiert auf dem selben Ansatz wie MP3.

Es ist offensichtlich, dass eine verlustbehaftete Speicherung für die professionelle Audioverarbeitung nicht in Frage kommt. Für die Auswahl eines geeigneten Formates müssen zudem weitere Kriterien herangezogen werden. Für die Verarbeitung mit Samples sind daher zwei Kriterien besonders wichtig:

1. Der Prozessor soll bei der Verarbeitung so wenig wie möglich belastet werden.
2. Es muss möglich sein, in der Datei einen Beginn, einen Loop und ein Ende zu definieren.

Während bei einer unkomprimierten Datei der Prozessor die Daten bereits so liest, wie sie in der Verarbeitung benötigt werden, müssen diese bei einer komprimierten Datei vor der weiteren Verarbeitung entpackt werden. Dies benötigt zusätzliche Rechenleistung, die in dem Moment nicht zur Verarbeitung anderer Audiodaten zur Verfügung steht. Daher ist ein Format wie FLAC für diesen Zweck nur bedingt geeignet.

Die zweite Anforderung bezüglich Beginn, Loop und Ende zur Kennzeichnung innerhalb des Samples erfüllen sowohl WAV als auch AIFF. Die Verbreitung der beiden Formate ist durch die dahinter stehenden Firmen bedingt. In Umgebungen, in denen vorrangig mit Apple-Computern gearbeitet wird, ist AIFF das verbreitete Format. Demgegenüber ist in Windowsdominierten Umgebungen WAV verbreitet.

Da die Software Hauptwerk zunächst für Windows entwickelt wurde und erst mit Version 2 auch auf Mac OS portiert wurde, ist es daher leicht zu erklären, dass sich das WAV-Format in diesem Bereich als Standard etabliert hat.

### 2.4.2. Zusammenhang zwischen Pfeifenlänge, Pfeifenfrequenz und benötigter Samplingrate

Wie in Abschnitt 2.2.1 erläutert, wird der Klang von Orgelpfeifen durch die Obertöne charakterisiert. Durch das Abtasttheorem ist bekannt, wie groß die Zeitabschnitte zwischen zwei Simulationsschritten mindestens sein müssen, damit die Schwingung erkennbar ist. Für die Simulation ergibt sich nun der interessante Zusammenhang, dass die Länge der Pfeife zur gewünschten Frequenz indirekt proportional ist, zu  $\Delta T$  hingegen direkt proportional ist. Das bedeutet, dass sich bei einer Pfeife mit niedriger Frequenz bei gleichbleibender Auflösung des Gitters die Anzahl der Knoten erhöht, gleichzeitig jedoch die Anzahl der benötigten Rechenschritte pro Sekunde sinkt.

Tabelle 2.1.: Übersicht der Obertöne über dem Kammerton  $a^1$

Bezeichnung	Frequenz	Notenname	Fußzahl
Grundton	440 Hz	$a^1$	8'
1. Oberton	880 Hz	$a^2$	4'
2. Oberton	1320 Hz	$e^3$	$2\frac{2}{3}'$
3. Oberton	1760 Hz	$a^3$	2'
4. Oberton	2200 Hz	$cis^4$	$1\frac{3}{5}'$
5. Oberton	2640 Hz	$e^4$	$1\frac{1}{3}'$
6. Oberton	3080 Hz	$g^4$	$1\frac{1}{7}'$
7. Oberton	3520 Hz	$a^4$	1'

In obiger Tabelle 2.1 sind die ersten 7 Obertöne für den Kammerton  $a^1$  mit ihren Frequenzen und der Fußlage eines entsprechenden Registers aufgeführt. Bisher wurde nur der Grundton betrachtet, für die charakteristischen Klangunterschiede von Orgelpfeifen sind jedoch die Obertöne verantwortlich.

Es ist nun für die Betrachtung wichtig, welcher höchste Oberton noch in der Simulation dargestellt werden soll. Soll wie im Beispiel der 7. Oberton simuliert werden, so ist

$$f_{max} = 3520Hz$$

und damit gilt für die Samplingfrequenz

$$\begin{aligned} f_{sampling} &\geq 2,2f_{max} \\ f_{sampling} &\geq 2,2 \cdot 3520Hz \\ f_{sampling} &\geq 7744Hz \end{aligned}$$

Daraus ergibt sich für den Zeitschritt

$$\begin{aligned}\Delta T &= \frac{1}{f} \\ \Delta T &= \frac{1}{7744Hz} \\ \Delta T &= 1,2^{-04}\end{aligned}$$

Dies bedeutet aber nur, dass die erzeugten Obertöne eindeutig rekonstruiert werden können. Es muss dabei noch beachtet werden, dass die zwischen innerem und äußerem der Pfeife pendelnde Luftströmung möglicherweise einen noch kleineren Zeitschritt benötigt. Ergebnisse aus der Simulation zur Frage nach einem möglichst großen Zeitschritt werden in Abschnitt 3.2.3 präsentiert.

# 3. Aufbau der Simulation

## 3.1. Beschreibung des Testaufbaus

Simuliert werden soll eine Orgelpfeife im Raum, der von der Orgelpfeife erzeugte Klang soll dabei gespeichert und analysiert werden. Insgesamt wurden im Verlauf dieser Arbeit drei Ansätze durchgeführt. Zur Simulation wurde in Ansatz 1 und 2 FreeFOAM, in Ansatz 3 OpenFOAM genutzt. Beide Programme unterscheiden sich prinzipiell nur in Details, die Entwicklung von FreeFOAM wurde jedoch Ende 2017 eingestellt<sup>1</sup>, sodass ein Wechsel auf OpenFOAM sinnvoll erschien.

Um optimale Ergebnisse zu erzielen wurde entschieden, dass folgende Teile der Orgelpfeife und ihrer Umgebung simuliert werden müssen:

- Der Pfeifenkörper, in dem die stehende Luftsäule schwingt.
- Der Pfeifenfuß mit Fußloch.
- Der Bereich direkt vor der Pfeife um den vor der Pfeife entstehenden Wirbel aufzunehmen.
- Freier Raum im Bereich vor dem Pfeifenkörper und unterhalb des Labiums um die von der Pfeife ausgestoßenen Druckschwankungen aufzunehmen
- Bei offenen Pfeifen der freie Raum über dem Pfeifenkörper, um die bei offenen Pfeifen über den Pfeifenkörper hinausgehende stehende Luftsäule aufzunehmen.

Die Annahmen basieren dabei auf den in Janke (1997) gezeigten Ergebnissen die als Anhaltspunkte für die Simulation dienten.

Weitere Parameter der Simulation ergeben sich aus daraus, dass die simulierte Pfeife in einer normalen Umgebung steht. Das bedeutet, sie wird bei Zimmertemperatur mit Luft angeströmt. Die Umsetzung ist in den Listings A.3 und A.4 beispielhaft angegeben. Das Dateiformat unterscheidet sich zwischen FreeFOAM und OpenFOAM, die Werte selbst sind jedoch gleich. Daher wird an dieser Stelle darauf verzichtet, die Werte und Dateien für alle 3 Ansätze wiederzugeben.

---

<sup>1</sup>Siehe Wild (2017).

## 3.2. Beschreibung des Ablaufes

Orgelbau beruht auf hunderten Jahren Erfahrung und Entwicklung. Insbesondere beim auf den Bau der Orgel folgenden Intonation muss sehr viel Zeit aufgewendet werden damit die Pfeifen genau so klingen, wie vom Orgelbauer vorgesehen. Dabei entscheiden oft Bruchteile von Millimetern über guten oder schlechten Klang. Es war daher nicht zu erwarten, dass die Simulation einer Orgelpfeife beim ersten Versuch gelingt.

Im Rahmen dieser Arbeit wurden drei verschiedene Pfeifen modelliert und bei diesen verschiedene Parameter der Pfeife und der Simulation variiert, um ein zufriedenstellendes Ergebnis zu erreichen.

Während die Länge der Pfeife (im Zusammenhang mit dem gewählten Simulationsschritt) anfangs so festgelegt wurde, dass sie mit 1000 Hz schwingt wurden andere Parameter der Pfeife wie die Position des Labiums, die Breite des Aufschnitts oder der Druck, mit dem die Pfeife angeblasen wird, verändert. Weitere Parameter der Simulation, die besonders zwischen den Ansätzen verändert wurden, waren das Verhalten der begrenzenden Wände und das Verhalten der Luft, in der sich die Pfeife befindet.

### 3.2.1. Ansatz 1

Der erste Ansatz war eine sehr einfach aufgebaute Pfeife, die als Ausgangspunkt der Forschung von Professor Hlawitschka zur Verfügung gestellt wurde.

Diese erste Variante ist sehr einfach aufgebaut und reduziert die Orgelpfeife auf die wesentlichen Bestandteile. Das Labium ist spitz, die Pfeife ist außen glatt, innen wird das Labium aus der Pfeifenwand geformt. Der Pfeifenfuß als solches wird durch ein schmales Rohr nachgebildet, an dessen linken Ende die Luft einströmt. Zur Untersuchung der Schallausbreitung ist wie in Abbildung 3.1 vor der Pfeife sehr viel Raum nachgebildet.

Durch diesen sehr einfachen Aufbau ist auchb wie in Abbildung 3.2 zu sehenb das Gitter im Bereich des Labiums sehr einfach aufgebaut. Eine durchgeführte Simulation führte nach wenigen Schritten zu einem Abbruch.

Als Ursache für den Abbruch der Simulation wurde vermutet, dass die Pfeife in ihren Abmessungen und Parametern nicht berechnet ist. So wurde zunächst anhand eines Messrechners<sup>2</sup> der vorgesehene Durchmesser der Pfeife nach Normmessur berechnet und in die Simulation übernommen. Mit diesen übernommenen Parametern brach die Simulation nicht mehr ab, die Pfeife begann jedoch auch nicht wie erwartet zu schwingen.

---

<sup>2</sup>Siehe Brüner (2002).

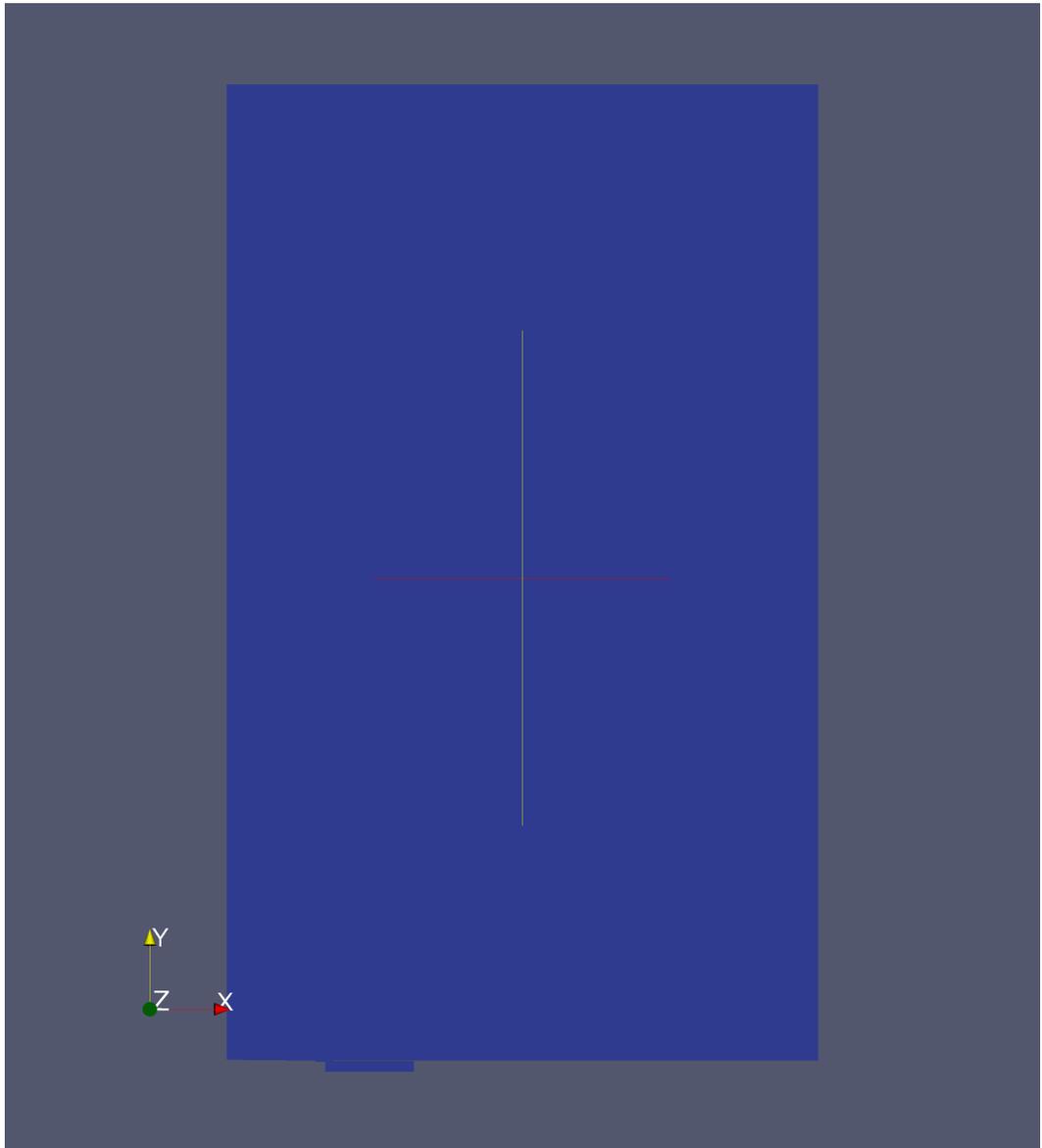


Abbildung 3.1.: Screenshot aus ParaView - Originales Gitter, Ansatz 1.



Abbildung 3.2.: Screenshot aus ParaView - Detailansicht der Pfeife, Ansatz 1.

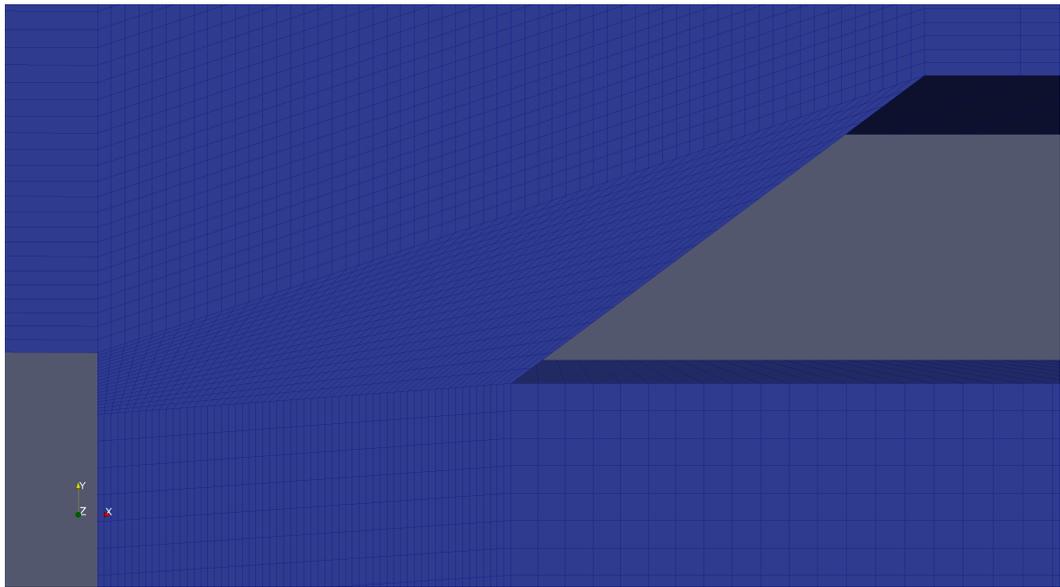


Abbildung 3.3.: Screenshot aus ParaView - Detailansicht des Bereichs des Labiums, Ansatz 1.

So wurde als weitere Maßnahme die Position der Kernspalte verändert. Ausgangspunkt war eine Bauanleitung mit exakten Maßen zu den Pfeifen<sup>3</sup>. Aus dieser ist ersichtlich, dass die Kernspalte prinzipiell unter der Spitze des Labiums ist und nicht, wie im Gitter, außerhalb.

Durch diese Änderung ergab sich jedoch im Bereich des Labiums die in Abbildung 3.3 sichtbare starke Scherung des Gitters.

Im Rahmen der Arbeit mit diesem Gitter wurden mehrere Versuchsreihen durchgeführt und dabei versucht, experimentell geeignete Werte für die Anströmung der Pfeife zu ermitteln. In Abbildung 3.4 ist beispielhaft der Druck vor und in der Pfeife visualisiert. Es ist sichtbar, dass sich die abgestrahlten Schallwellen konzentrisch um den Pfeifenmund ausbreiten. Der Druck innerhalb der Pfeife entspricht jedoch nicht dem erwarteten Bild und führt nicht zu einer korrekten Schwingung, auch wenn die Verteilung außerhalb der Pfeife sinusförmig aussieht.

Die Graphiken 3.5 und 3.6 zeigen Visualisierungen der Strömung bei Versuchen mit diesem Gitter. Wie schon bei der Druckverteilung entspricht die Strömung nicht der Erwartung.

---

<sup>3</sup>Siehe Beijer (2010) Seite 7ff.

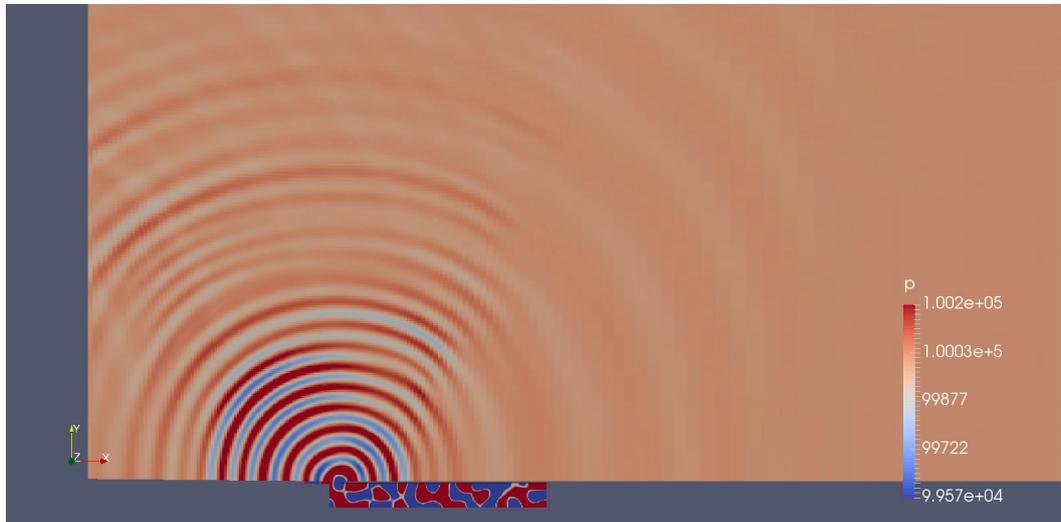


Abbildung 3.4.: Screenshot aus ParaView - Druckverteilung vor und in der Pfeife, Ansatz 1.

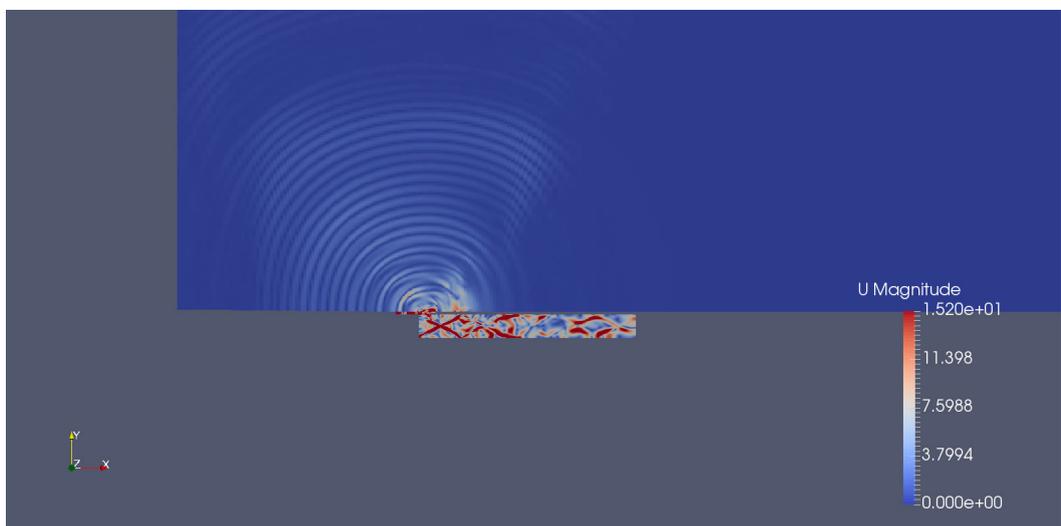


Abbildung 3.5.: Screenshot aus ParaView - Strömung vor und in der Pfeife, Ansatz 1.



Abbildung 3.6.: Screenshot aus ParaView - Strömung vor und in der Pfeife, Ansatz 1.

### 3.2.2. Ansatz 2

Da vermutet wurde, dass die starke Scherung im Bereich des Labiums die Simulation beeinträchtigt, wurde das Gitter neu aufgebaut. Dabei wurde darauf geachtet, dass im Bereich des Labiums keine Scherungen auftreten. Die Dimensionierung der Pfeife folgte dabei wieder Beijer (2010), Seite 7ff.

Die Simulation zeigte dabei jedoch ähnliche Probleme wie die ursprüngliche Version. Im Wesentlichen bereitete dabei nach wie vor das Labium Probleme, da es je nach Winkel des Labiums zu den gleichen Fehlermeldungen kam wie zuvor.

Kam es nicht zu Fehlermeldungen zeigten die berechneten Simulationen ein recht eigenartiges Verhalten im freien Raum. In Abbildung 3.9 ist der Winddruck im freien Raum dargestellt. Wie in der Abbildung ersichtlich kommt es im Verlauf der Simulation zu einer massiven Verringerung des Druckes. Aus dem Graph ist auch kein Einschwingvorgang der Pfeife zu erkennen. Ignoriert man die generelle Verringerung des Druckes und betrachtet nur lokale Minima und Maxima des Graphen ist dennoch keine gleichmäßige Welle zu erkennen. Insbesondere müsste nach 0,3 Sekunden der Einschwingvorgang abgeschlossen sein. Das ist im Graph nicht zu erkennen.

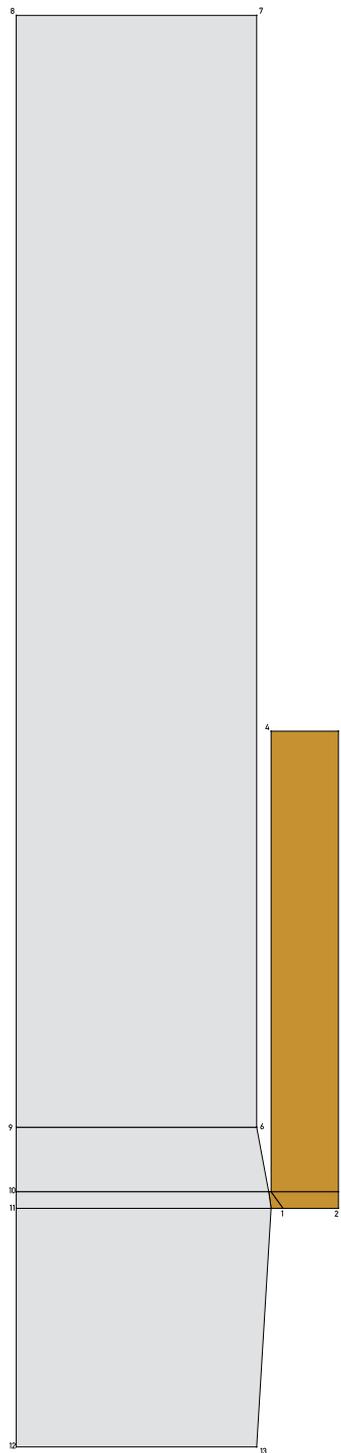


Abbildung 3.7.: Graphik des zweiten Versuches.

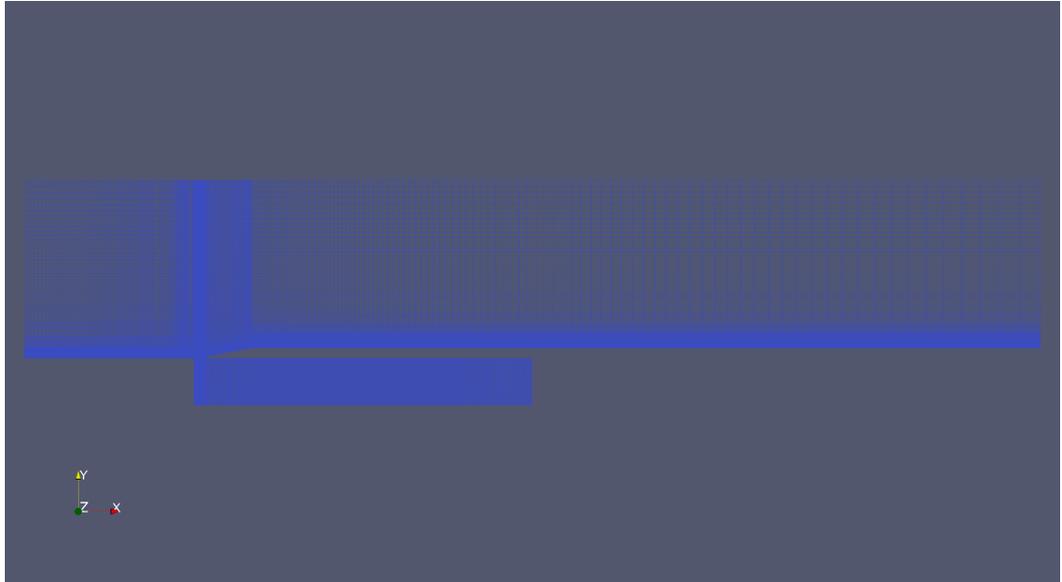


Abbildung 3.8.: Screenshot aus ParaView - Gitter des zweiten Versuchs.

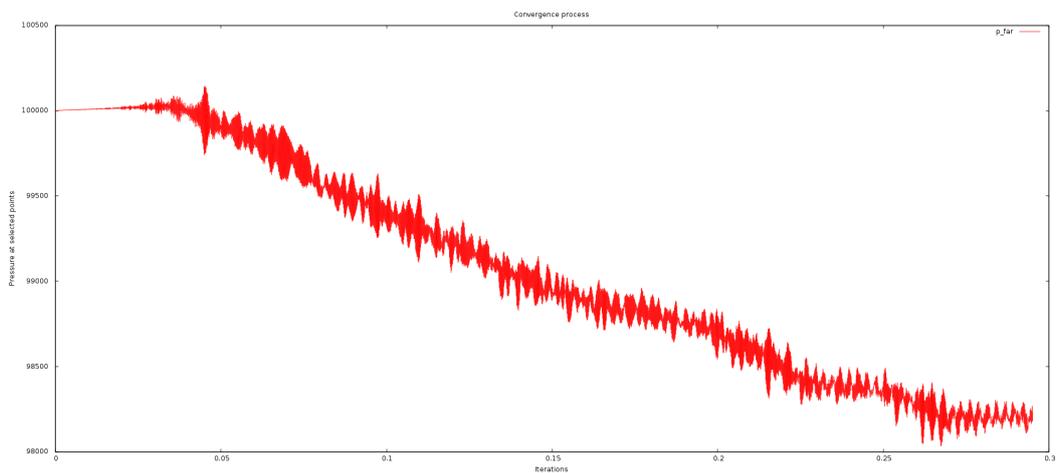


Abbildung 3.9.: Druckabfall in der Luft vor der Pfeife

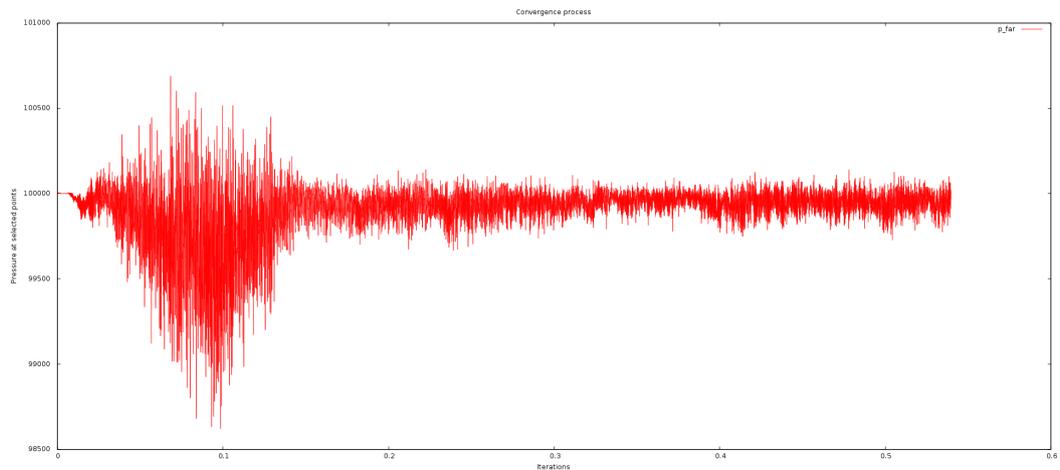


Abbildung 3.10.: Kein Druckabfall in der Luft vor der Pfeife

Eine Veränderung der Begrenzung des Raumes vor der Orgelpfeife führt zu einem anderen Verhalten. War die Begrenzung zuvor als wall, also feste Wand modelliert wurde sie nun als space, also freier Raum mit bestimmten Eigenschaften, modelliert. Wie in Abbildung 3.10 ersichtlich gibt es nun eine kurze Phase mit sehr hohen Druckunterschieden, der eine Phase von Schwingungen geringerer Amplitude folgt. Diese kurze Phase zu Anfang kann als Einschwingvorgang interpretiert werden, die folgende Schwingung folgt jedoch keiner festen Frequenz und erzeugt nicht die gewünschten Töne.

In der graphischen Darstellung der Ergebnisse ist ersichtlich, warum die Pfeife nicht die erwünschten Schwingungen erzeugte. Es sind deutlich zwei sich überlagernde, falsche Schwingungen zu sehen. In den Abbildungen 3.11 bis 3.16 ist zu sehen, wie sich in der Pfeife sehr viele Schwingungsknoten ausbilden. Diese sind regelmäßig angeordnet, bilden jedoch keine stehende Welle in der Form, in der sie gewünscht ist. Die Abbildungen 3.18 und 3.19 zeigen, dass sich eine Schwingung sogar längs der Orgelpfeife entwickelt und nicht quer. Einige Zeitschritte weiter ist in Abbildung 3.20 zu sehen, dass die Druckunterschiede in der Pfeife keine für eine gewünschte Schwingung notwendige Regelmäßigkeit aufweisen.

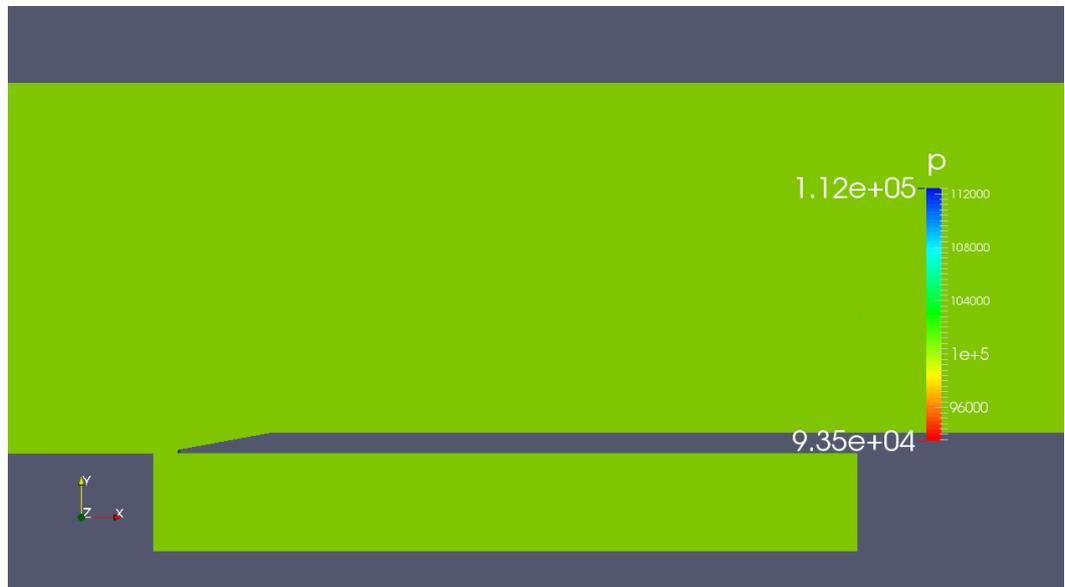


Abbildung 3.11.: Druck in der Pfeife bei Zeitschritt 0.0, Version 2

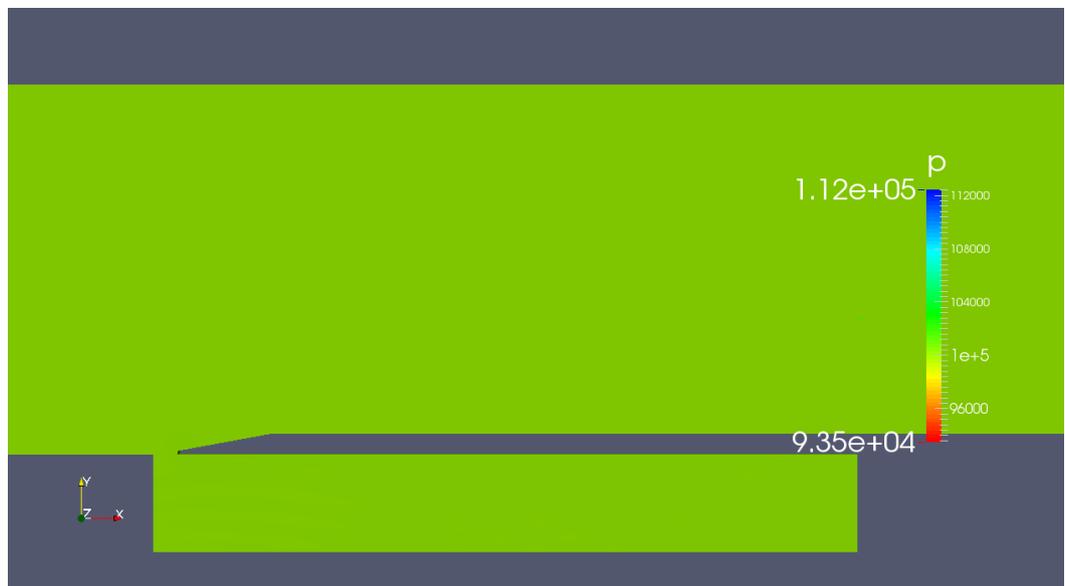


Abbildung 3.12.: Druck in der Pfeife bei Zeit 0,001 ms, Version 2



Abbildung 3.13.: Druck in der Pfeife bei Zeit 0,002 ms, Version 2



Abbildung 3.14.: Druck in der Pfeife bei Zeit 0,003 ms, Version 2

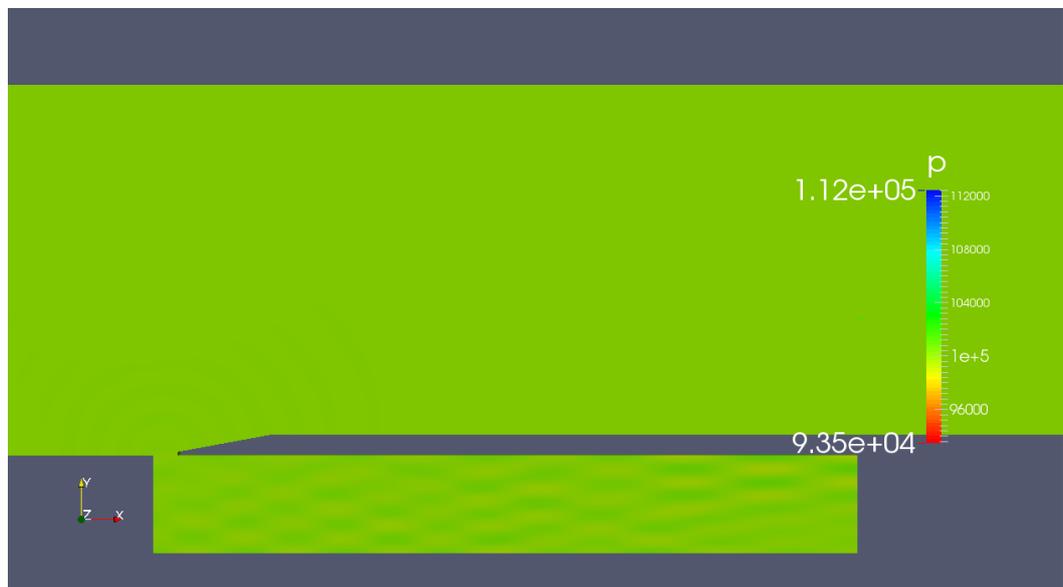


Abbildung 3.15.: Druck in der Pfeife bei Zeit 0,004 ms, Version 2



Abbildung 3.16.: Druck in der Pfeife bei Zeit 0,005 ms, Version 2

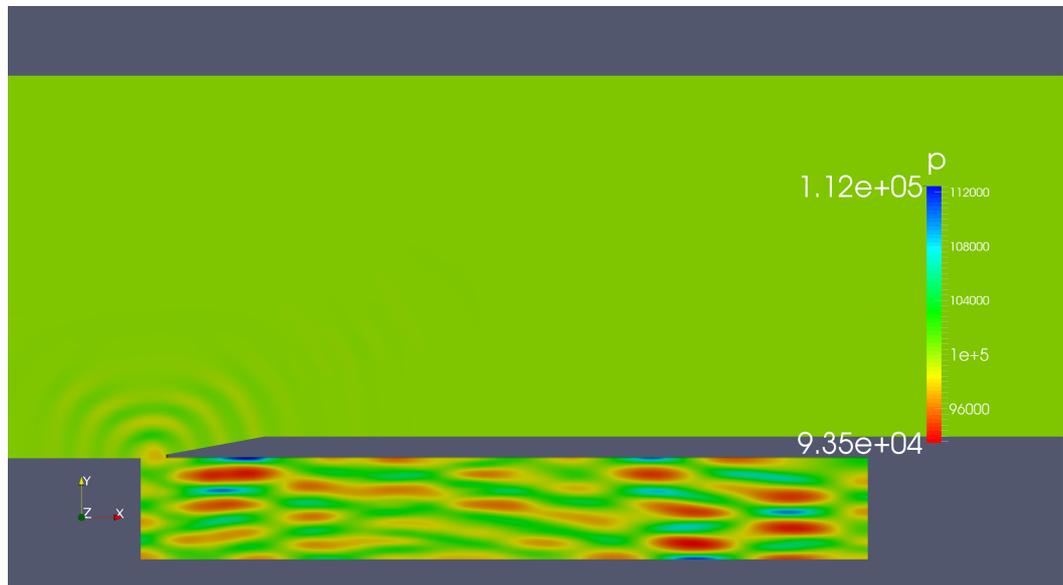


Abbildung 3.17.: Druck in der Pfeife bei Zeit 0,01 ms, Version 2

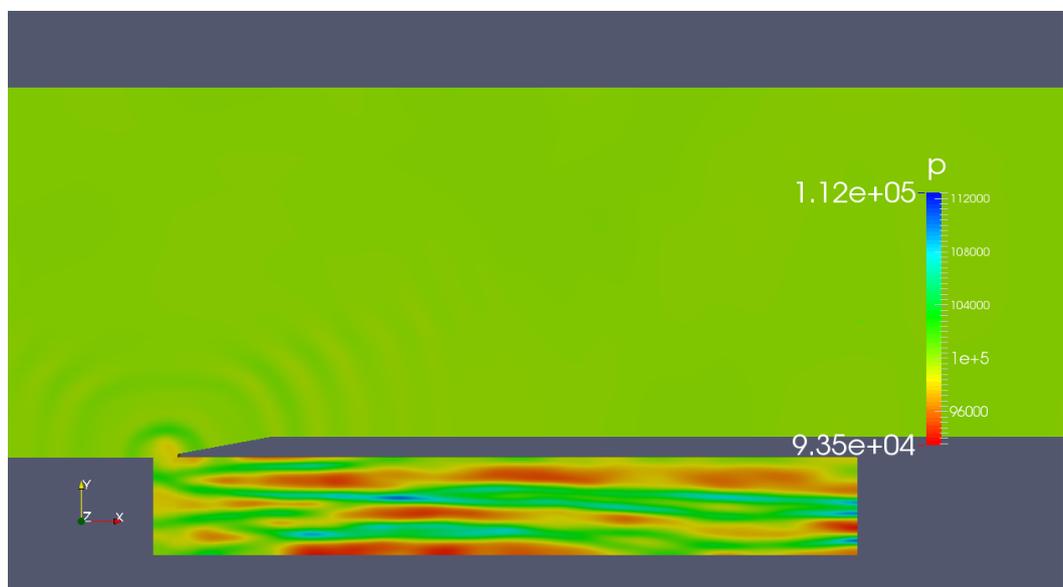


Abbildung 3.18.: Druck in der Pfeife bei Zeit 0,032 ms, Version 2

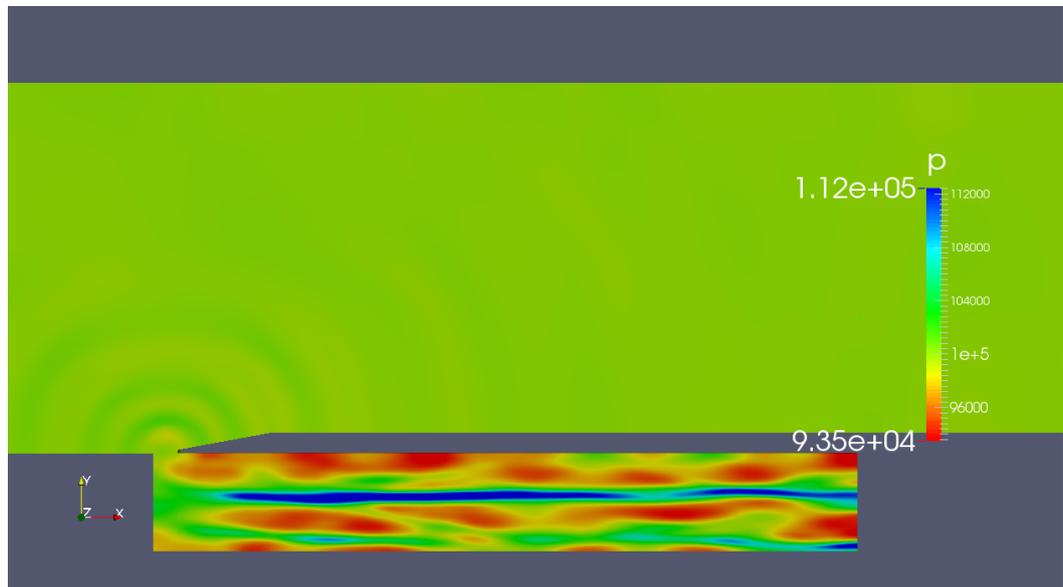


Abbildung 3.19.: Druck in der Pfeife bei Zeit 0,041 ms, Version 2

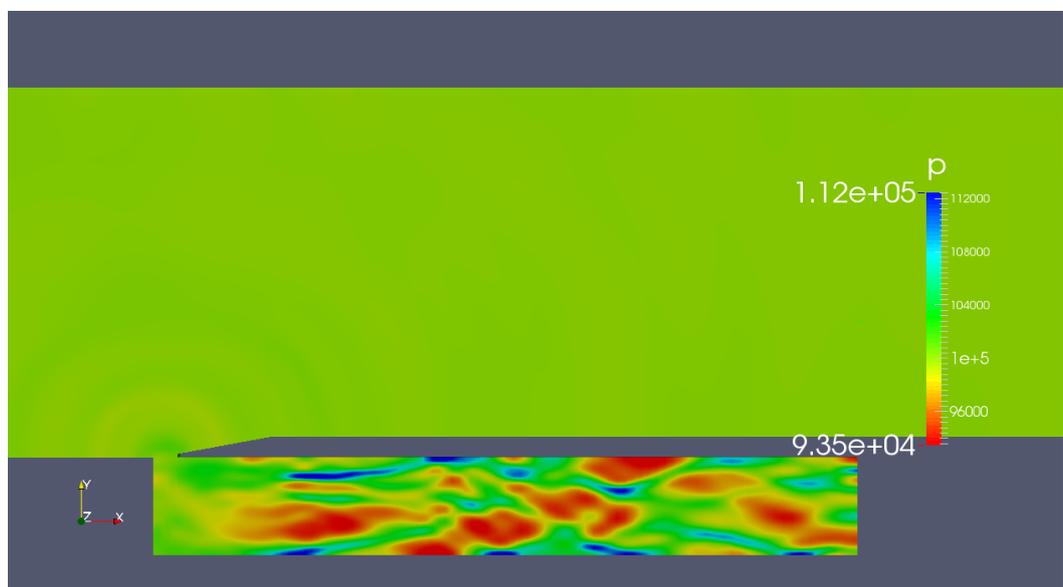


Abbildung 3.20.: Druck in der Pfeife bei Zeit 0,25 ms, Version 2



Abbildung 3.21.: Luftströmung in der Pfeife bei Zeit 0,05 ms, Version 2

Diese Unregelmäßigkeit wird ebenso sichtbar, wenn man die Strömung in der Pfeife betrachtet. In den Abbildungen 3.21 und 3.22 ist diese abgebildet. In beiden Graphiken ist zu erkennen, dass im Pfeifenkörper Schwingungen mit sehr hoher Frequenz entstehen. In Abbildung 3.21 ist zudem zu sehen, wie die Luft am Labium sowohl nach innen, als auch nach außen gelenkt wird und ein Wirbel entsteht. Daraus kann geschlussfolgert werden, dass die Pfeife und die Simulation prinzipiell korrekt sind, denn für die Entstehung der Schwingungen in der Pfeife muss die Luft zwischen Pfeifeninnerem und Raum schwingen. Somit konnte als Erkenntnis für die weitere Arbeit an der Simulation gewonnen werden, dass ein Schwingen prinzipiell möglich ist. Es müssen offensichtlich nur die richtigen Parameter gefunden werden und die Simulation mit der nötigen Genauigkeit berechnet werden.

Die unterschiedliche Anströmgeschwindigkeit von  $10 \frac{m}{s}$  und  $18 \frac{m}{s}$  - vgl. die Listings A.6 und A.9 - widerspricht zwar dem Prinzip des *ceteris paribus*, ist hier aber nicht relevant. Das gezeigte Verhalten änderte sich stets nur in Abhängigkeit der Patch-Typen, nicht jedoch bei Änderung der Anströmgeschwindigkeit.

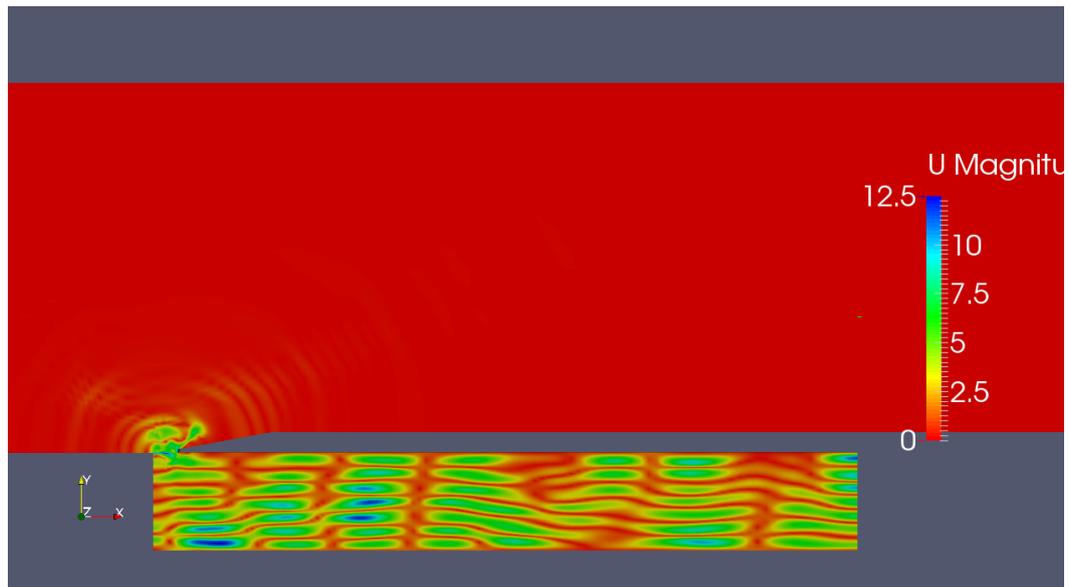


Abbildung 3.22.: Luftströmung in der Pfeife bei Zeit 0,07 ms, Version 2

### 3.2.3. Ansatz 3

Aufgrund der anhaltenden Probleme in Ansatz zwei wurde entschieden, das Gitter noch einmal, wesentlich detaillierter aufzubauen. Inspiriert durch die in Wand (2018) vorgestellten Pfeifen wurde das Labium bei diesem Ansatz mittig über der Kernspalte angeordnet. Des Weiteren wurde der simulierte Raum um die Pfeife gegenüber den vorherigen Ansätzen reduziert.

#### Aufbau der Simulation

Zunächst wurde die Pfeife als gedackte Pfeife modelliert wie in Abbildung 3.23 dargestellt und mit  $20 \frac{m}{s}$  angeblasen<sup>4</sup>. Um Einflüsse der verschiedenen Parameter einer Pfeife zu überprüfen wurden anschließend zwei weitere Simulationen berechnet. In einer ersten Variante wurde der Pfeifendurchmesser verkleinert um darzustellen, wie dieser den Klang beeinflusst. In einer weiteren Variante wurde der Winddruck verringert und die Pfeife nur noch mit  $10 \frac{m}{s}$  angeblasen.

Anschließend wurde die Pfeife darauf aufbauend als offene Pfeife modelliert. Dabei war es wichtig, über der Pfeife einen Bereich zu modellieren um die über den Pfeifenkörper hinausgehende stehende Luftsäule<sup>5</sup> in Gänze zu si-

<sup>4</sup>In der Simulation wurde die austretende Luft über ihre Geschwindigkeit festgelegt. Daher ist die Einheit für den Winddruck an dieser und an weiteren Stellen nicht wie zu erwarten mmWS, sondern  $\frac{m}{s}$ .

<sup>5</sup>Siehe dazu Kapitel 2.2.1.

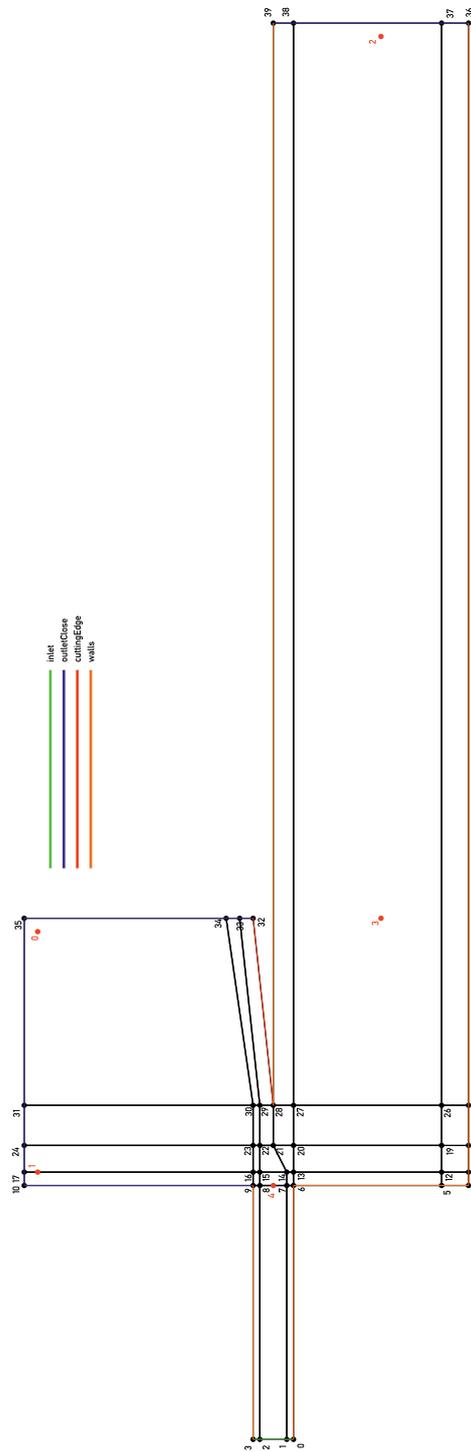


Abbildung 3.23.: Graphik des dritten Versuches als gedackte Pfeife.

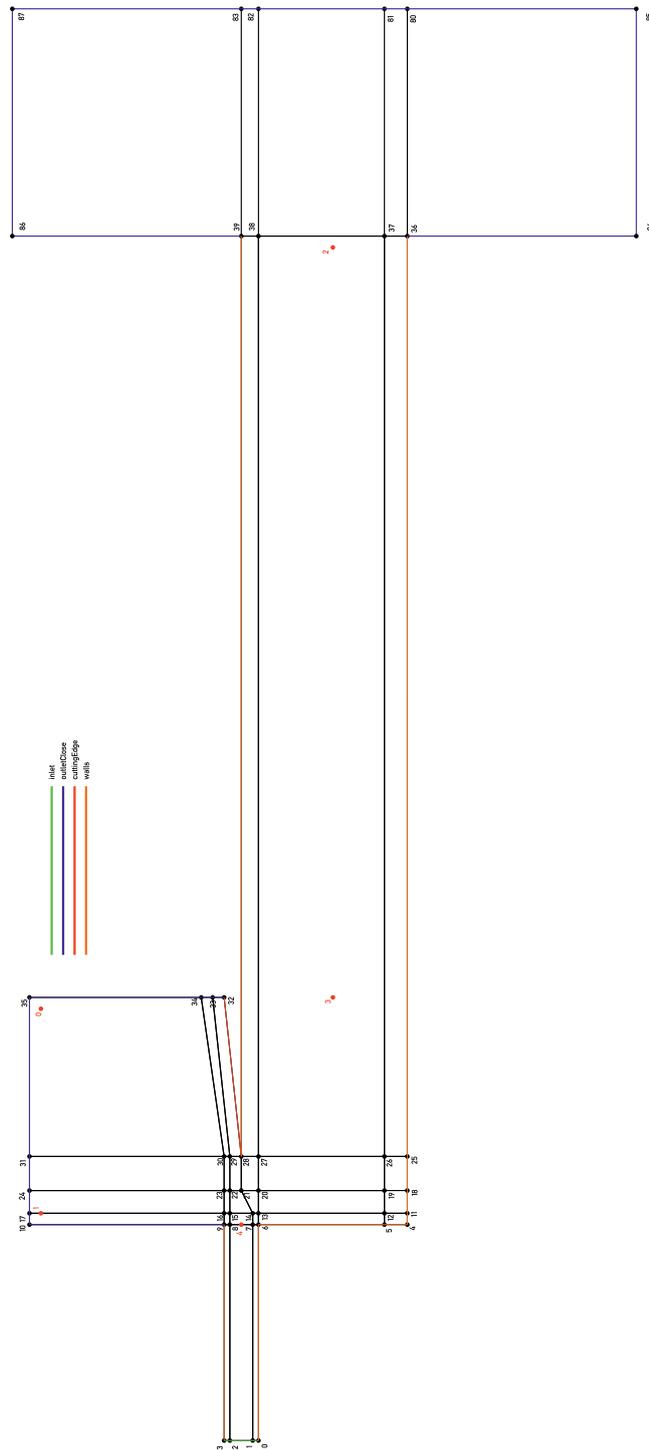


Abbildung 3.24.: Graphik des dritten Versuches als offene Pfeife mit zusätzlichem Raum über der Pfeife.

mulieren. Dies wurde durch den in Abbildung 3.24 sichtbaren, offenen Raum über der Pfeife erreicht. Dieser Platz reicht aus, um die stehende Luftsäule aufzunehmen. Mehr Raum zu simulieren ist mit Blick auf die dafür nötige Rechenleistung nicht sinnvoll.

### Optimierung der Simulation

Um die Simulation bei gleichbleibender Rechenzeit schneller zu berechnen und so schneller zu sehen, ob die Simulation die gewünschten Ergebnisse hervorbringt wurde als Ansatz die Vergrößerung des Zeitschrittes gewählt. In der ursprünglichen Version wurde als Zeitschritt  $1^{-07}$  s gewählt, eine Verringerung auf  $1^{-06}$  s würde somit die zur Berechnung nötige Zeit auf ein Zehntel verkürzen und damit immer noch weit den Anforderungen des Abtasttheorems genügen. Experimentell zur Bestimmung geeigneter Parameter wäre sogar eine Vergrößerung auf  $1^{-05}$  s möglich.

Diese Optimierung brachte jedoch nicht die gewünschten Ergebnisse. Wie in Listing A.11 ersichtlich bricht die Simulation bei einer Vergrößerung des Zeitschrittes auf  $1^{-06}$  s ab. Die Fehlermeldung zeigt eine extrem niedrige Temperatur in der Simulation an was auf eine plötzliche Entspannung hindeutet. In der graphischen Darstellung ist gut zu sehen an welchem Punkt es zu der problematischen Berechnung kommt. Da sonst alle Parameter gleich blieben ist ein optischer Vergleich gut möglich. In den Abbildungen 3.25 und 3.26 ist die Temperatur in der Simulation bei Zeit 0,07 ms bei kleinem respektive großen Zeitschritt dargestellt. Es fällt direkt auf, dass die Welle im Bereich des Labiums bei großem Zeitschritt weniger scharf abgegrenzt ist. Dies wird auch deutlich bei Betrachtung des Druckes zu diesem Zeitschritt, siehe Graphiken 3.27 und 3.28. Zudem zeichnet sich am Ende des Unterlabiums ein Minimum ab. Dies wird noch deutlicher sichtbar bei Betrachtung des letzten Zeitschritts vor dem Abbruch der Simulation der in Abbildung 3.29 abgebildet ist. Eine Detailaufnahme mit veränderter Skala zeigt dann an dieser Stelle eine große Temperaturdifferenz, sichtbar in Abbildung 3.30. Damit ist ersichtlich, an welcher Stelle die zum Abbruch führende negative Temperatur auftritt.

## 3.3. Bewertung der Ergebnisse

Nach der durchgeführten, zeitintensiven Simulation sind die Ergebnisse von besonderem Interesse. Dabei ist zu überprüfen, ob die Ergebnisse der Erwartung entsprechen und welchen Effekt die in der Simulation durchgeführten Änderungen haben. Dies beinhaltet sowohl Analysen des Frequenzspektrums, als auch Analysen der Strömungen und Druckverhältnisse über Visualisierungen.

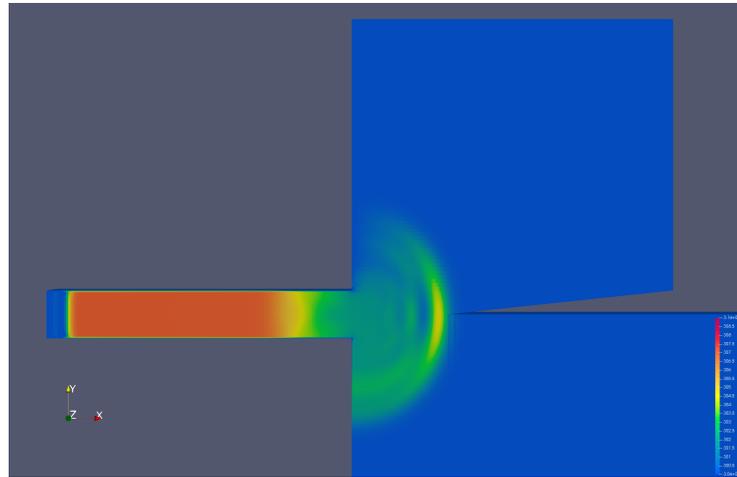


Abbildung 3.25.: Temperatur in der Pfeife bei Zeit 0,00007 s, Version 2, Zeitschritt  $1^{-07}$  s

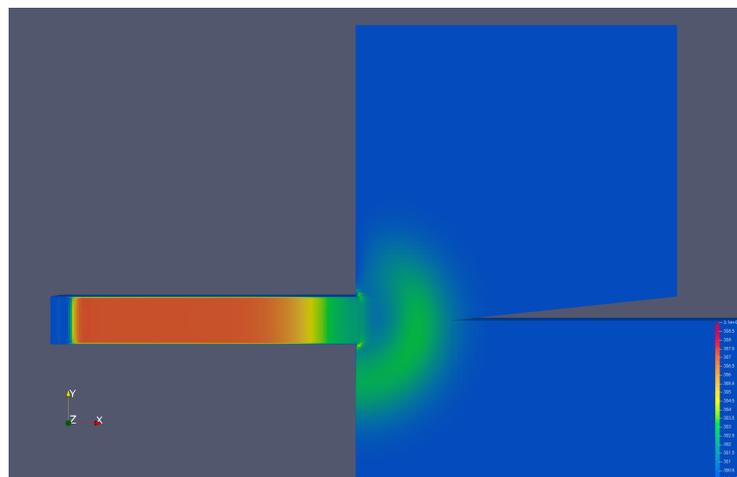


Abbildung 3.26.: Temperatur in der Pfeife bei Zeit 0,00007 s, Version 2, Zeitschritt  $1^{-06}$  s

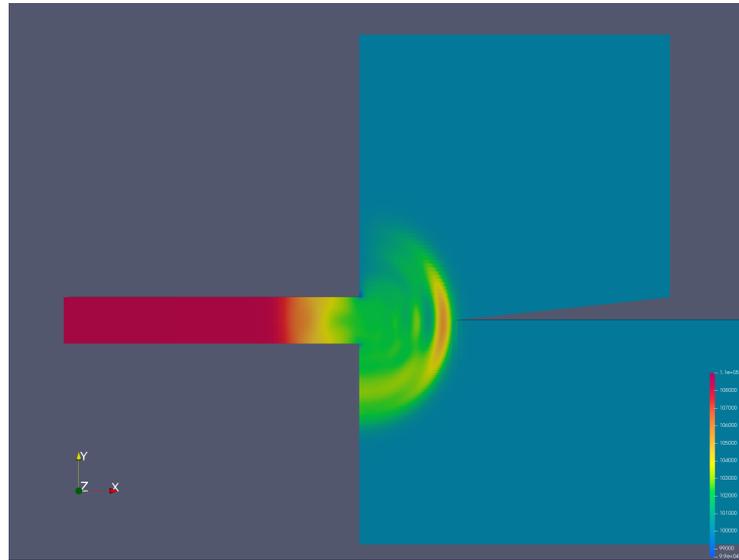


Abbildung 3.27.: Druck in der Pfeife bei Zeit 0,00007s, Version 2, Zeitschritt  $1^{-07}$  s

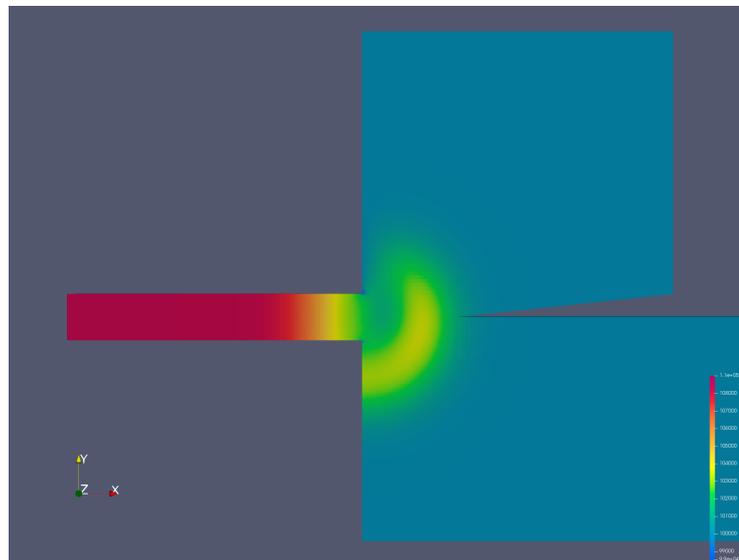


Abbildung 3.28.: Druck in der Pfeife bei Zeit 0,00007s, Version 2, Zeitschritt  $1^{-06}$  s

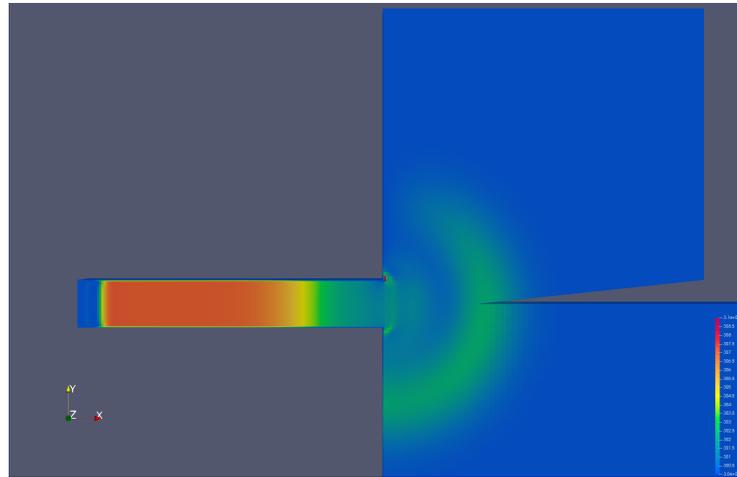


Abbildung 3.29.: Temperatur in der Pfeife bei Zeit 0,000077 s, Version 2, Zeitschritt  $1^{-06}$  s



Abbildung 3.30.: Temperatur in der Pfeife bei Zeit 0,000077 s, Version 2, Zeitschritt  $1^{-06}$  s, vergrößert und mit größerer Farbskala

### 3.3.1. Gedackte Pfeife

Die erzeugten Schwingungen sind in der vergleichenden Abbildung 3.31 zu erkennen. Es ist auch ersichtlich, dass alle drei Schwingungen unterschiedliche Ausprägungen haben.

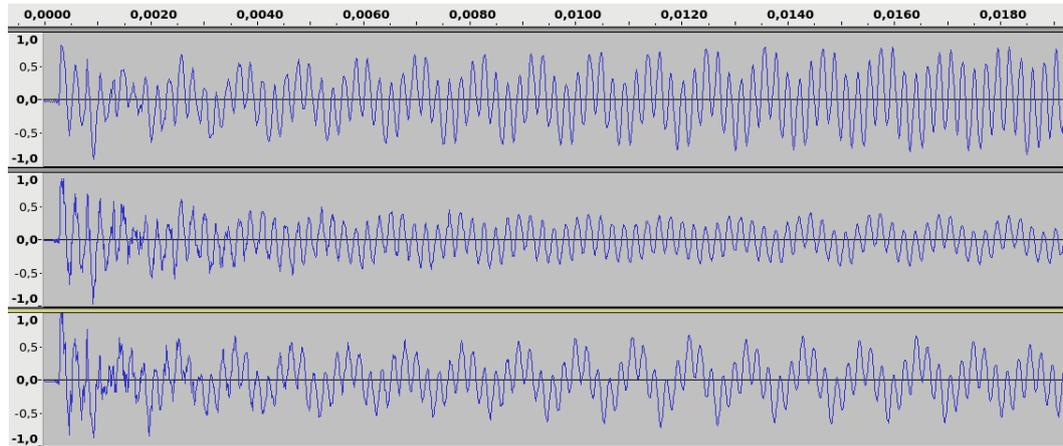


Abbildung 3.31.: Vergleich der erzeugten Schwingungen in den 3 verschiedenen Versionen der gedackten Pfeife, Screenshot aus Audacity. Oben die unveränderte, gedackte Pfeife, darunter die mit geringem Winddruck angeblasene Pfeife, ganz unten die schmale gedackte Pfeife.

Die Schwingung der Pfeife ist in der Visualisierung deutlich zu sehen. In den Abbildungen 3.32 und 3.34 ist dabei die Schwingung bei ihrer maximalen Amplitude zu sehen. In Abbildung 3.33 ist der Zeitpunkt des Nulldurchgangs visualisiert. Deutlich sichtbar sind zwei Schwingungsknoten bei  $\frac{1}{4}$  und  $\frac{3}{4}$  der Pfeifenlänge.

Vergleichend dazu können die Visualisierungen der Strömung zu diesen Zeitpunkten betrachtet werden. Deutlich ist in den Abbildungen 3.35, 3.36 und 3.37 der die Schwingungen verursachende Wirbel zu sehen. Interessanterweise gibt es genau zu dem Zeitpunkt, an dem der Druck innerhalb der Pfeife gleich ist, eine Strömung die schwach in Abbildung 3.36 sichtbar ist.

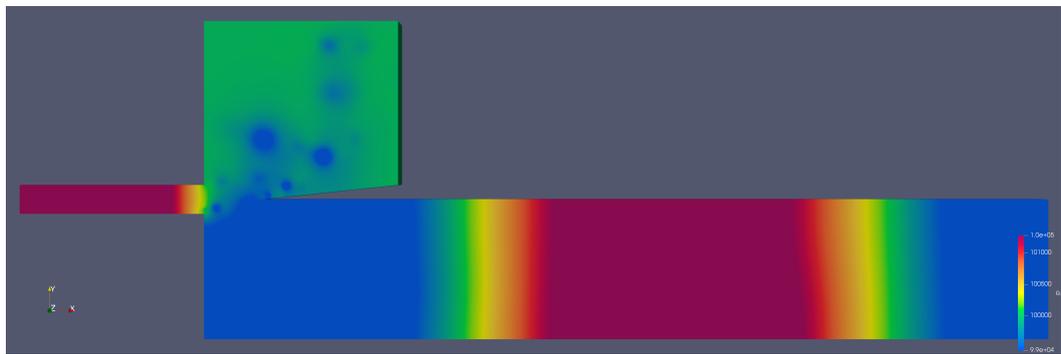


Abbildung 3.32.: Visualisierung des Drucks bei 0,01696 s

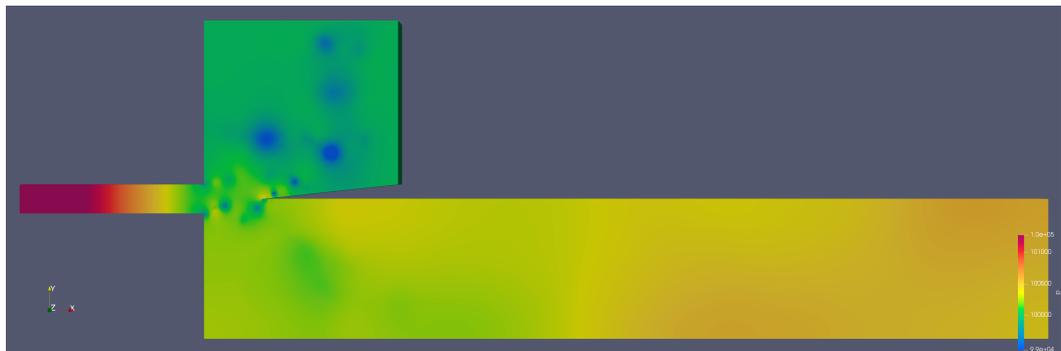


Abbildung 3.33.: Visualisierung des Drucks bei 0,01702 s

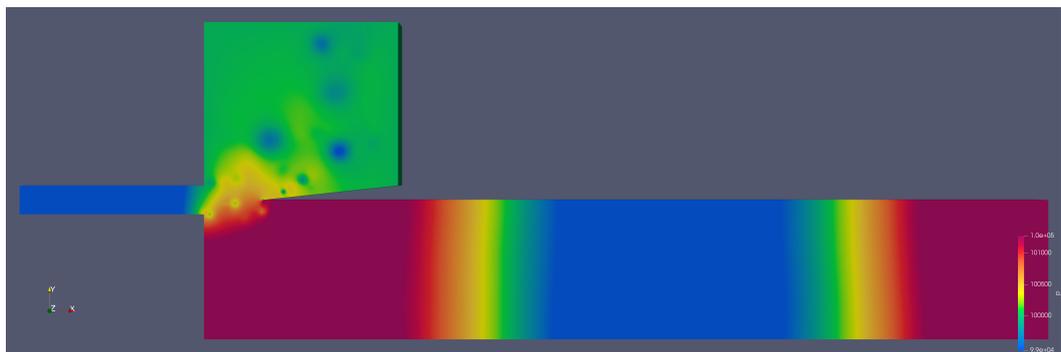


Abbildung 3.34.: Visualisierung des Drucks bei 0,01708 s

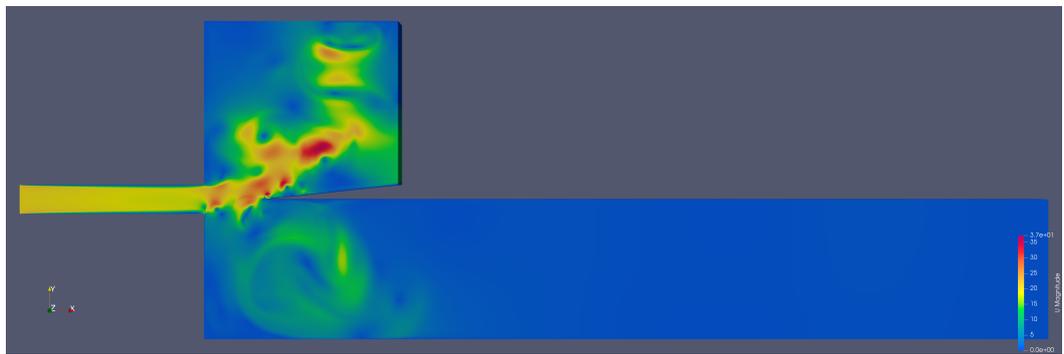


Abbildung 3.35.: Visualisierung der Strömung bei 0,01696 s

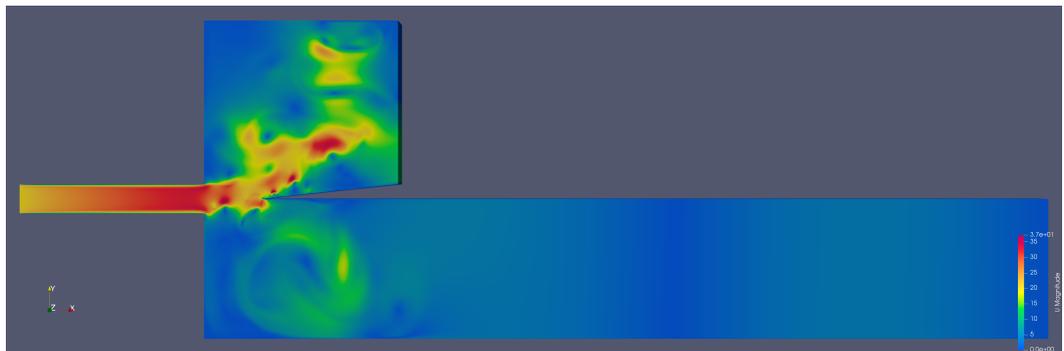


Abbildung 3.36.: Visualisierung der Strömung bei 0,01702 s

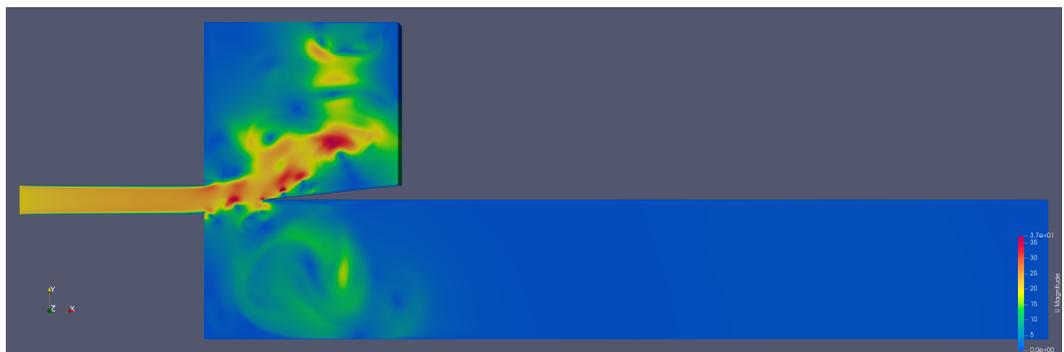


Abbildung 3.37.: Visualisierung der Strömung bei 0,01708 s

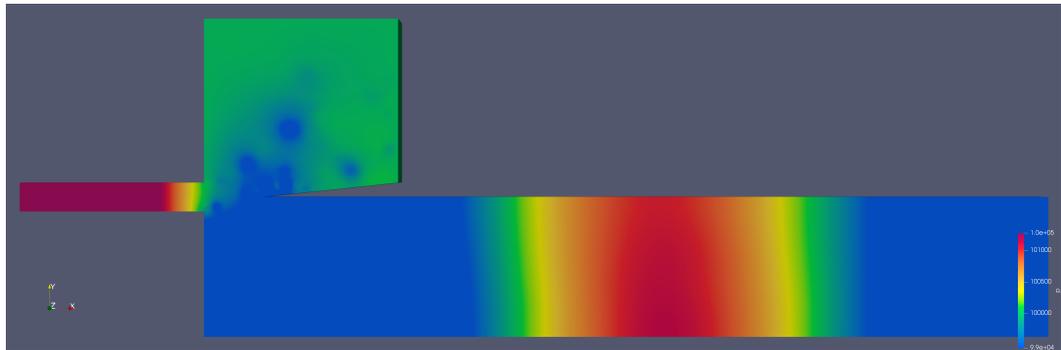


Abbildung 3.38.: Visualisierung des Drucks bei 0,01411 s

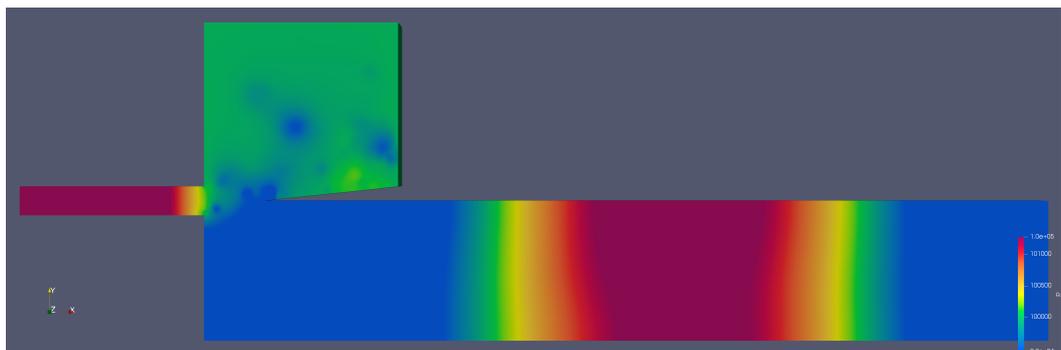


Abbildung 3.39.: Visualisierung des Drucks bei 0,01499 s

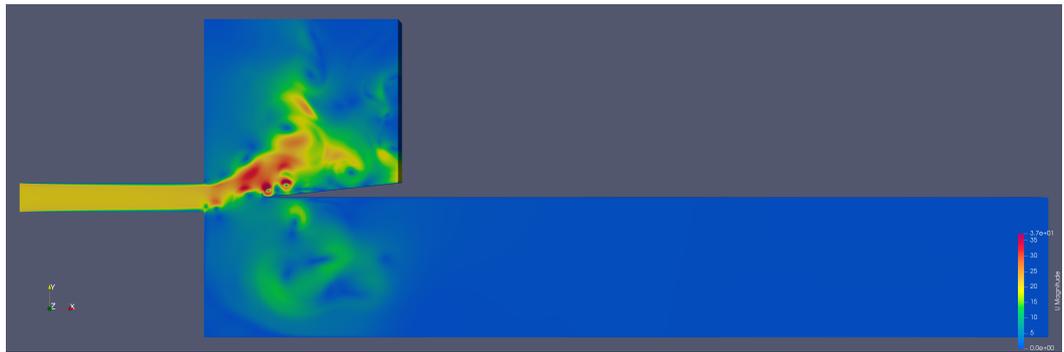


Abbildung 3.40.: Visualisierung der Strömung bei 0,01411 s

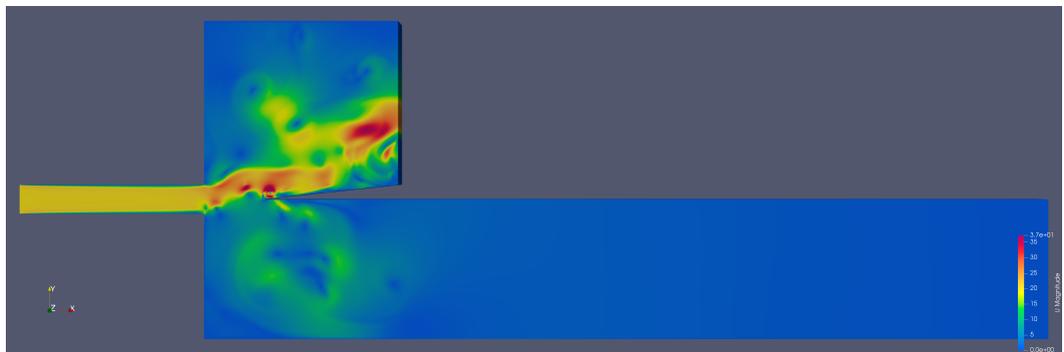


Abbildung 3.41.: Visualisierung der Strömung bei 0,01499 s

Die beiden deutlich sichtbaren Schwingungsknoten weisen darauf hin, dass es sich um eine vielfache der Grundschwingung handelt. Dies wird beim Vergleich mit dem Diagramm in Abbildung 3.31 deutlich. Betrachtet man die Visualisierung des Druckes an den Punkten der maximalen und minimalen Amplitude so ist zu erkennen, dass der Wert in der Pfeifenmitte in Abbildung 3.38 einen niedrigeren Wert annimmt als in Abbildung 3.39. Dies deckt sich mit den aus dem Diagramm ersichtlichen Werten. Auch die Betrachtung der Strömung zeigt dies auf. So ist deutlich zu sehen, dass die Strömung in Abbildung 3.40 vom Labium wegströmt, während sie in Abbildung 3.41 am Labium entlangströmt.

## Ausgangsversion

Die erste Version der gedackten Pfeife bildet den Ausgangspunkt für den Vergleich mit allen weiteren Varianten. Grundlage für die Betrachtung des Ergebnisses ist die Frequenzanalyse in Abbildung 3.42. Verglichen mit einem Spektrum wie in Abbildung 2.4 zeigt das Spektrum der Simulation keine klar abgegrenzten Frequenzen. Dennoch lassen sich einige Frequenzen erkennen. So ist eine erste Frequenz bei ca. 912 Hz, dem Grundton, zu erkennen, eine zweite Spitze bei ca. 2760 Hz, dem 2. Oberton, und eine dritte, große Spitze bei 4550 Hz, dem 4. Oberton. Zwei kleinere Spitzen sind bei 1500 Hz (keinem Oberton dieser Frequenz zuordenbar) und bei 1830 Hz (1. Oberton). Dies entspricht auch den Frequenzen, die mit Blick auf das Diagramm in Abbildung 3.43 zu erwarten sind.

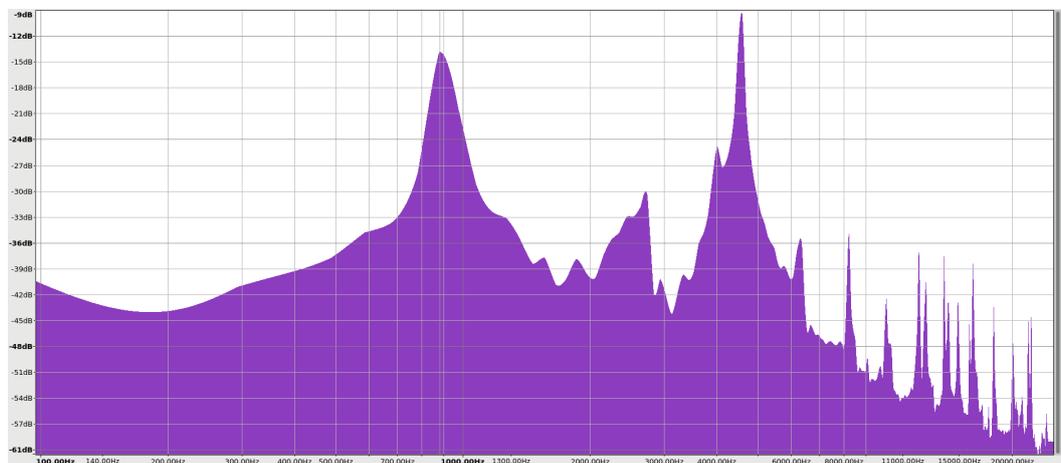


Abbildung 3.42.: Frequenzanalyse der gedackten Pfeife.

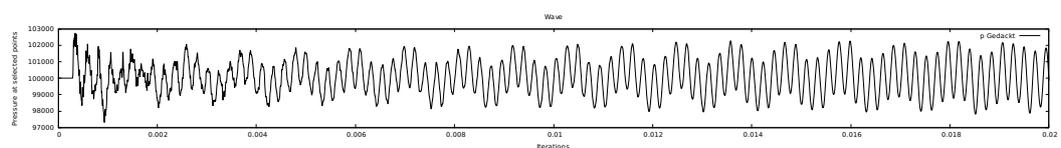


Abbildung 3.43.: Die erzeugte Schwingung der unveränderten, gedackten Pfeife.

Diese Graphik lässt auch den Einschwingvorgang sehr deutlich erkennen. Nach einer kurzen Phase gleichbleibenden Druckes beginnt der Einschwingvorgang mit harten, zackigen Druckschwankungen. Erst nach 0,005 s ist diese Phase des Einschwingvorgangs beendet und die Schwingung wird runder und gleichmäßiger. Bis zum Ende des simulierten Zeitabschnittes sind jedoch noch

leichte Schwankungen erkennbar, so wächst die maximale Amplitude minimal und die Amplitude der Hüllkurve verringert sich.

### Geringerer Winddruck

In der ersten Variation der gedackten Pfeife wurde der Winddruck in Form der Anströmgeschwindigkeit von  $20 \frac{m}{s}$  auf  $10 \frac{m}{s}$  reduziert. Die Frequenzanalyse in Abbildung 3.44 zeigt zwei Spitzen bei ca. 763 Hz (Grundton) und 4500 Hz (5. Oberton), dazwischen sind zwei kleinere Spitzen bei ca. 1500 Hz (1. Oberton) und ca. 2600 Hz (2. Oberton) zu sehen. Eine kleine Spitze gibt es noch bei 4029 Hz, diese ist jedoch keinem Oberton zuzuordnen. Interessant ist der Blick auf die Skala. Der verringerte Winddruck manifestiert sich in einem wesentlich leiseren Ton. Dies ist auch an dem Diagramm der erzeugten Schwingung in Abbildung 3.45 zu erkennen. Deutlich ist der Einschwingvorgang erkennbar, der bei dieser Pfeife bis etwa 0,01 s dauert.

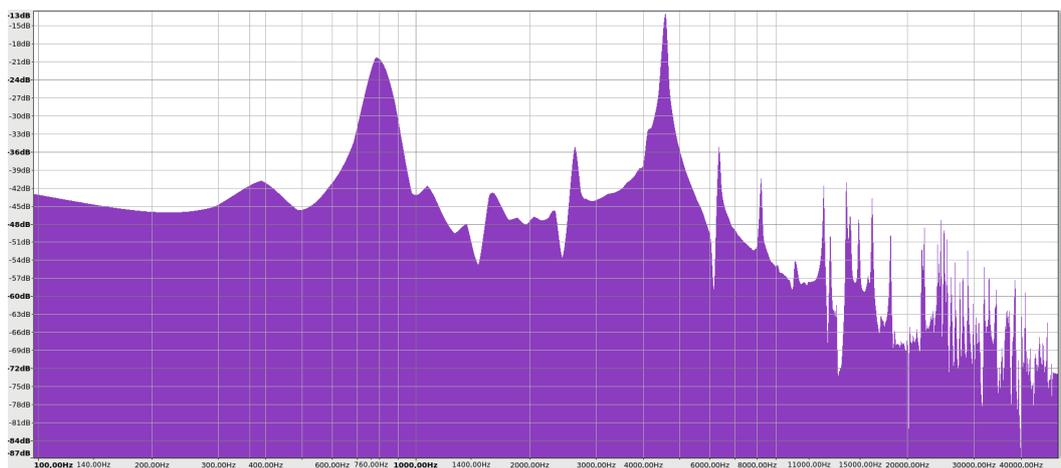


Abbildung 3.44.: Frequenzanalyse der mit geringerem Winddruck angeblasenen gedackten Pfeife.

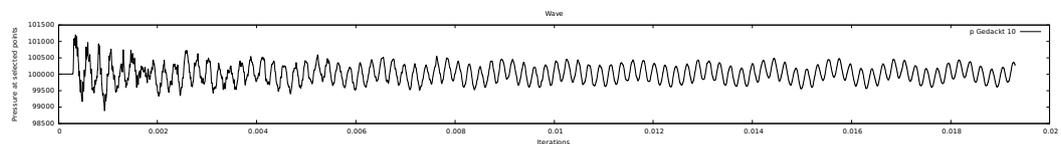


Abbildung 3.45.: Die erzeugte Schwingung der mit geringem Winddruck angeblasene gedackte Pfeife.

## Schmale Pfeife

Eine zweite Variation ist die Verringerung des Pfeifendurchmessers, der von 14,5 mm auf 10,5 mm. Die Frequenzanalyse in Abbildung 3.46 zeigt zwei große Spitzen bei ca. 920 Hz (Grundton) und 4800 Hz (4. Oberton) sowie zwei kleinere Spitzen bei ca. 1800 Hz (1. Oberton) und 2700 Hz (2. Oberton). Diese sind jedoch jetzt größer, was erkennen lässt, dass diese Obertöne eine größere Amplitude besitzen. In der Tat ist es so, dass Orgelpfeifen mit enger Mensur obertonreicher sind als Orgelpfeifen mit weiter Mensur<sup>6</sup>. Die Simulation zeigt also an dieser Stelle, dass sie durch die Eigenschaften einer Orgelpfeife genau so beeinflusst wird, wie es zu erwarten ist.

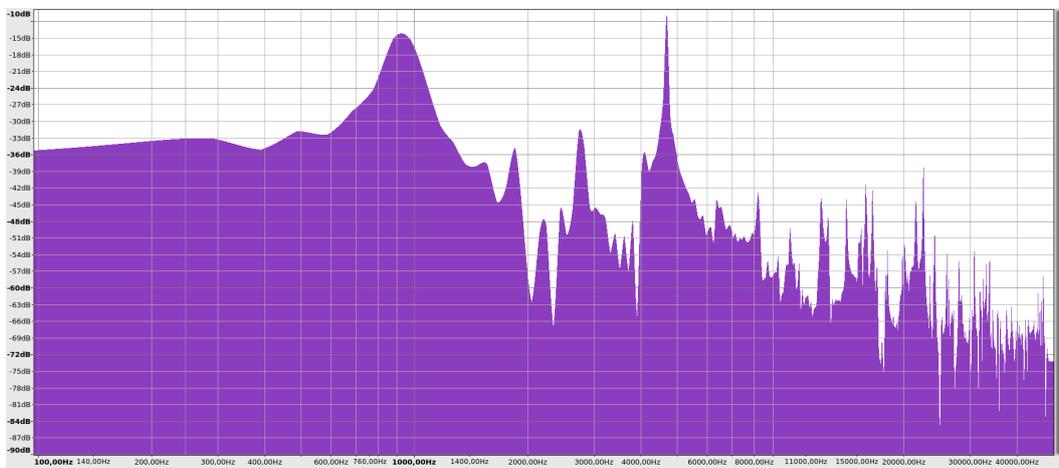


Abbildung 3.46.: Frequenzanalyse der schmalen, gedackten Pfeife.

Deutlich ist im Diagramm der erzeugten Schwingung in Abbildung 3.47 wieder der Einschwingvorgang und die Hüllkurve der Schwingung des Grundtöneidenees zu erkennen. Die stärkeren Obertöne führen hier zu einer Vergrößerung der Amplitude dieser Kurve.

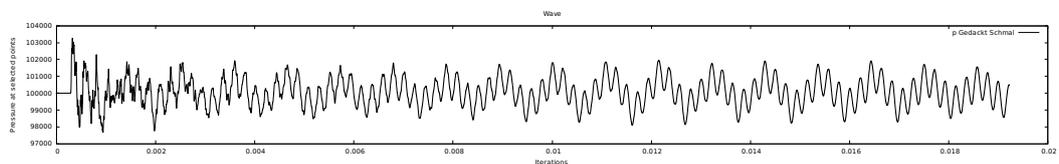


Abbildung 3.47.: Die erzeugte Schwingung der schmalen, gedackten Pfeife.

<sup>6</sup>Siehe Klotz (1998).

## Vergleich

Vergleicht man alle drei Versionen so finden sich sowohl Gemeinsamkeiten, als auch Unterschiede in den Schwingungen. Am Deutlichsten können diese Unterschiede anhand eines Kurvendiagramms der erzeugten Schwingungen sichtbar gemacht werden.

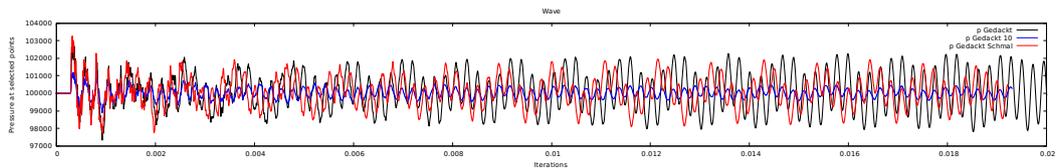


Abbildung 3.48.: Vergleich der erzeugten Schwingungen in den 3 verschiedenen Versionen der gedackten Pfeife. Schwarz die unveränderte, gedackte Pfeife, blau die mit geringen Winddruck angeblasene Pfeife, rot die schmale gedackte Pfeife.

Während in Abbildung 3.31 die Kurven zur besseren Sichtbarkeit der einzelnen Kurven untereinander dargestellt wurden, sind diese in Abbildung 3.48 in einem Diagramm zusammengefasst. Deutlich wird zunächst, dass die mit geringem Winddruck angeblasene Pfeife wesentlich leiser klingt als die anderen Pfeifen. Beim Vergleich zwischen weiter und enger Pfeife - im Diagramm schwarz respektive rot - ist der interessante Effekt zu sehen, dass die weite Pfeife mit minimal höherer Frequenz schwingt während die mit geringerem Winddruck angeblasene Pfeife mit deutlich niedrigerer Frequenz schwingt. Dies fällt in der Graphik besonders deshalb auf, weil die Schwingungen graphisch übereinander gelegt werden. Des Weiteren ist auch erkennbar, dass die weite Pfeife etwas lauter ist als die enge Pfeife.

### 3.3.2. Offene Pfeife

Die von den Varianten der offenen Pfeife erzeugten Schwingungen sind in Abbildung 3.49 dargestellt. Die Einschwingvorgänge sind deutlich zu sehen, die Schwingungen sind jedoch nicht deutlich ausgeprägt.

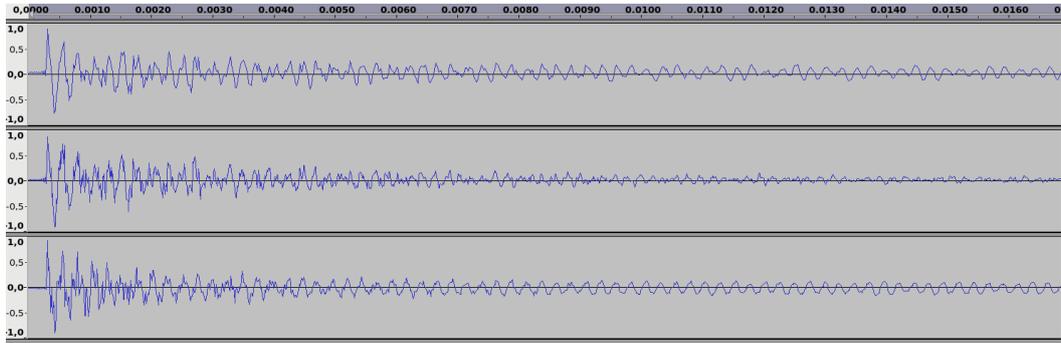


Abbildung 3.49.: Vergleich der erzeugten Schwingungen in den 3 verschiedenen Versionen der offenen Pfeife, Screenshot aus Audacity. Oben die unveränderte, offene Pfeife, darunter die mit geringem Winddruck angeblasene Pfeife, ganz unten die schmale offene Pfeife.

Die Visualisierung der Strömung in Abbildung 3.50 zeigt dennoch die typischen Wirbel am Pfeifenmund. Die Strömung verlässt das Labium nach außen während sich sowohl vor der Pfeife als auch in der Pfeife Wirbel bilden. Eine Besonderheit der offener Orgelpfeifen ist die über den Pfeifenkörper hinausgehende stehende Luftsäule. Der Bereich rechts stellt die Luft über der Pfeife dar, es ist deutlich eine leichte Verwirbelung am Ende des Pfeifenkörpers zu sehen.

Bei der weiteren Simulation kommt es zur Ausbildung eines weiteren Wirbels in der Pfeife. Dies ist in Visualisierung der Strömung in Abbildung 3.51 deutlich zu sehen. Auch die Darstellung der Druckverhältnisse in Abbildung 3.52 zeigt die Wirbel als runde Bereiche mit niedrigem Druck. In der gleichen Graphik ist auch ein Teil der Schwingung zu sehen, bei etwa  $\frac{2}{3}$  der Pfeifenlänge.

Verändert man den Wertebereich der Skala, so sind die Schwingungen in der Pfeife und die überstehende Luftsäule deutlich zu sehen. In Abbildung 3.54 ist bei etwa  $\frac{1}{4}$  der Pfeife ein Maximum der Schwingung zu sehen, zugleich sieht man dort die durch die Luftwirbel verursachten Bereiche mit minimalem Druck. Diese haben aber wenig Einfluss auf die dargestellte Schwingung. In Abbildung 3.55 ist zu sehen, dass auch die Luft außerhalb der Pfeife leicht zum Schwingen angeregt wird. Besonders deutlich ist dann in Abbildung 3.53 zu sehen, dass die stehende Luftsäule über das Pfeifenende hinausgeht.

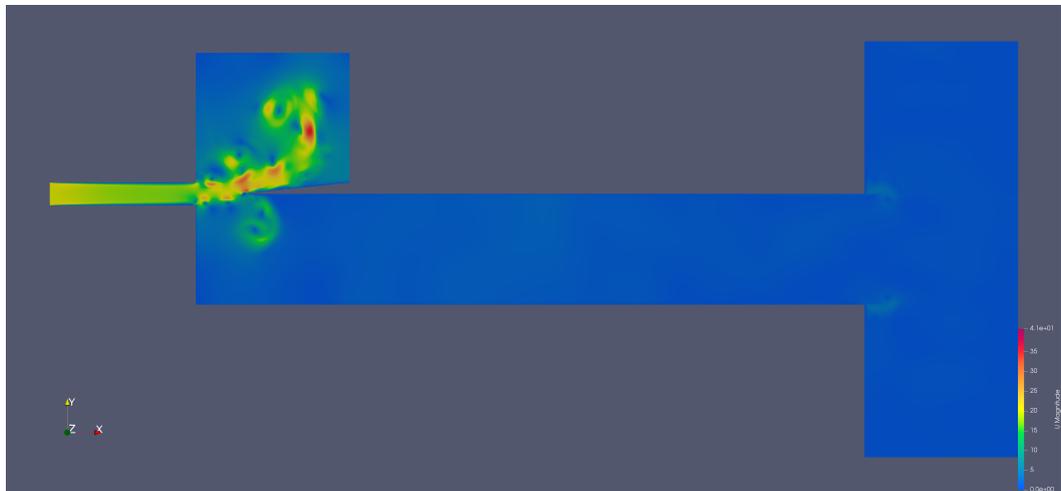


Abbildung 3.50.: Visualisierung der Strömung bei 0,002 s

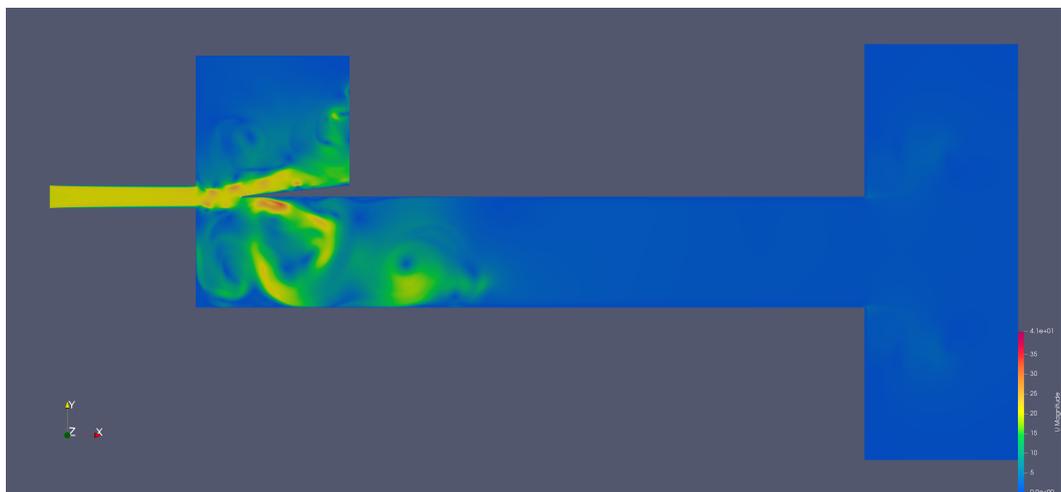


Abbildung 3.51.: Visualisierung der Strömung bei 0,01106 s

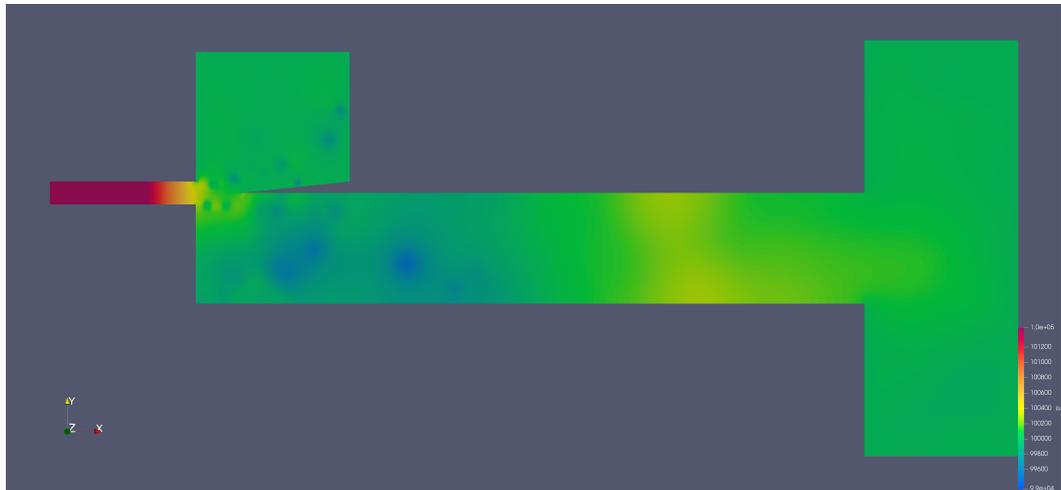


Abbildung 3.52.: Visualisierung des Drucks bei 0,01106 s

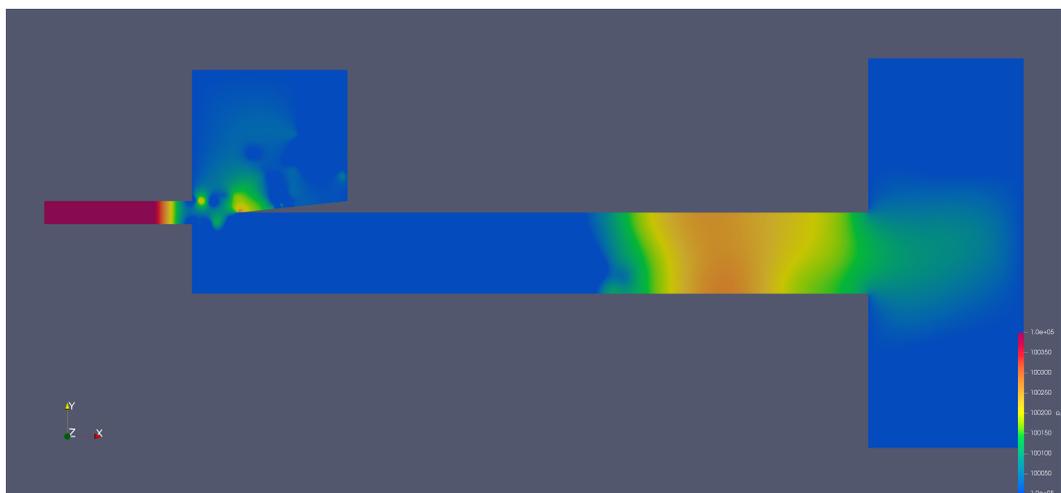


Abbildung 3.53.: Visualisierung des Drucks bei 0,01401s mit bewusst eingeschränktem Wertebereich der Skala, schmale Pfeife

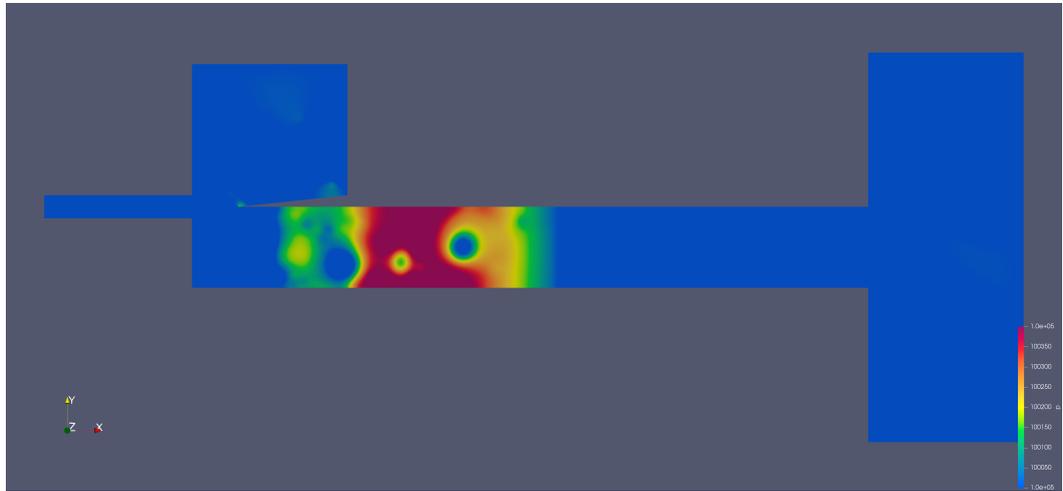


Abbildung 3.54.: Visualisierung des Drucks bei 0,01384s mit bewusst eingeschränktem Wertebereich der Skala, schmale Pfeife

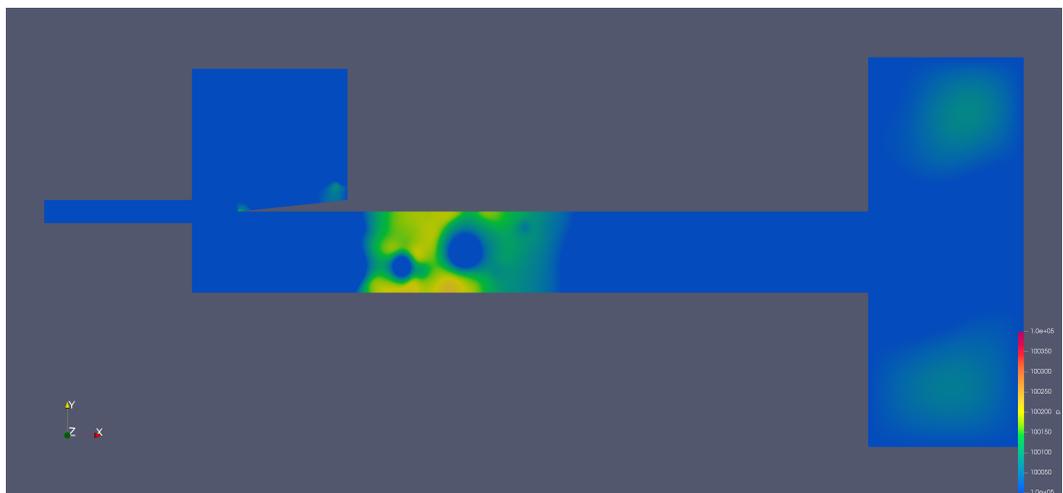


Abbildung 3.55.: Visualisierung des Drucks bei 0,01388s mit bewusst eingeschränktem Wertebereich der Skala, schmale Pfeife

### Ausgangsversion

Die Frequenzanalyse in Abbildung 3.56 zeigt einen erste Spitze bei 1473 Hz, dies entspricht dem Grundton der Pfeife. Weitere Spitzen bei 3229 Hz, 4093 Hz, 5080 Hz, 6689 Hz und 8109 Hz sowie einige Spitzen mit höherer Frequenz. Eine genaue Zuordnung der Spitzen zu einzelnen Obertönen ist hier nicht einfach möglich.

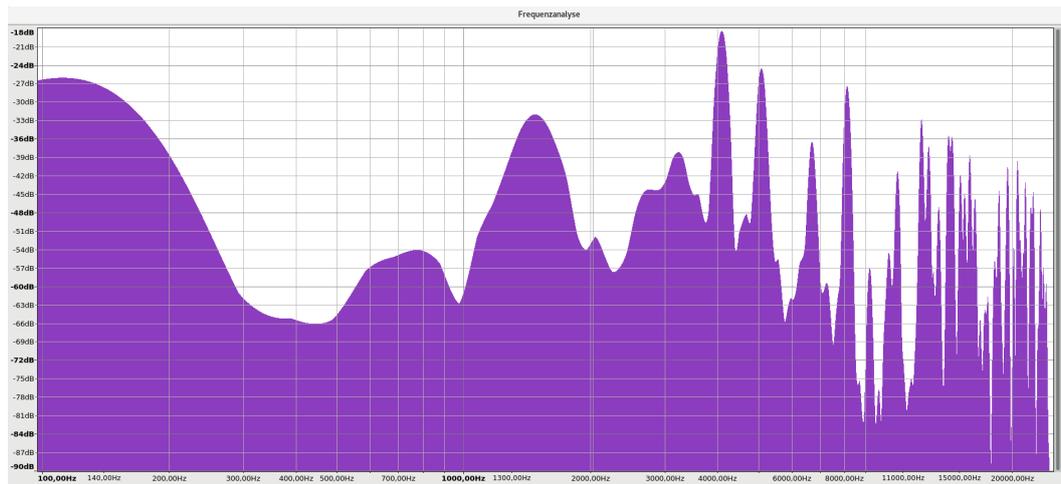


Abbildung 3.56.: Frequenzanalyse der offenen Pfeife.

Die Darstellung als Kurve in Abbildung 3.57 zeigt dennoch deutlich den Einschwingvorgang bis etwa 0,01 s mit harten Schwingungen der sich ab etwa 0,018 s stabilisiert.

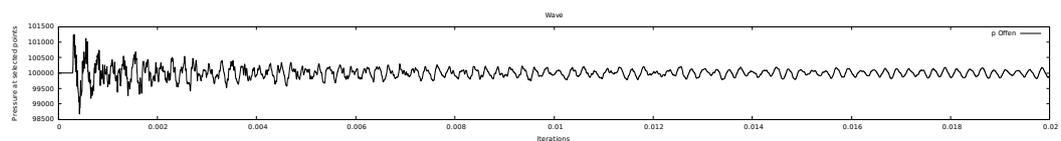


Abbildung 3.57.: Die erzeugte Schwingung der unveränderten, offenen Pfeife.

## Geringerer Winddruck

Die mit geringerem Winddruck angeblasene Pfeife zeigt in der Frequenzanalyse in Abbildung 3.58 deutlich abgegrenztere Frequenzen. Die erste Spitze ist hier bei 1619 Hz, dem Grundton. Es folgen 3297 Hz (1. Oberton), 4132 Hz (keinem Oberton zuordenbar) und 5088 Hz (2. Oberton), sowie zwei kleinere Spitzen 6712 Hz (3. Oberton) und 8102 Hz (4. Oberton). Anschließend noch weitere, hochfrequente Teiltöne.

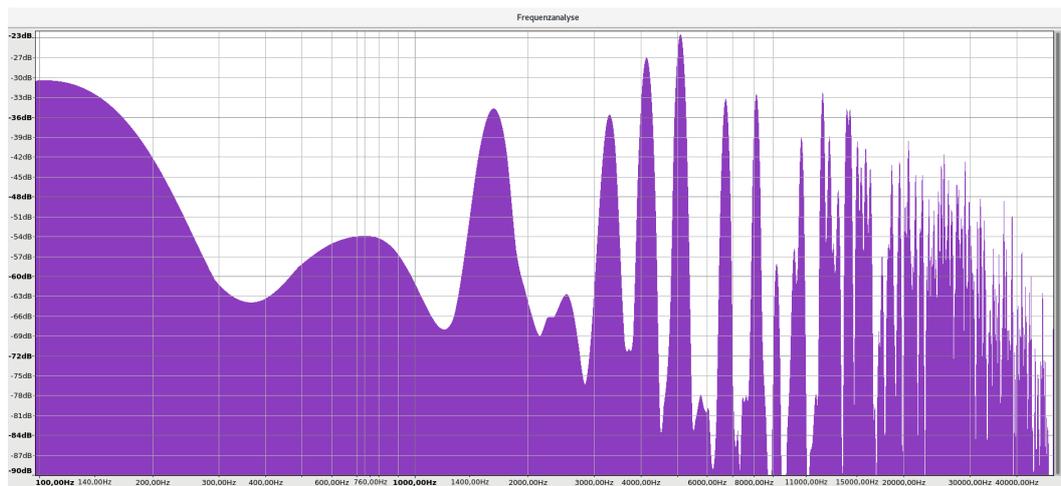


Abbildung 3.58.: Frequenzanalyse der mit geringerem Winddruck angeblasenen offenen Pfeife.

In der Darstellung als Kurve in Abbildung 3.59 ist wieder deutlich der Einschwingvorgang zu erkennen. Da die Pfeife mit weniger Winddruck angeblasen wurde ist zu sehen, dass die Amplitude wesentlich geringer ist als bei der unmodifizierten, offenen Pfeife.

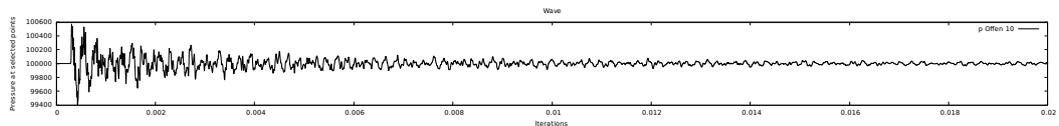


Abbildung 3.59.: Die erzeugte Schwingung der mit geringerem Winddruck angeblasenen offenen Pfeife.

Die Luftsäule in der Pfeife bildet keine Schwingungsknoten rechtwinklig zur Pfeifenwand aus. Dies wird deutlich in der Visualisierung in Abbildung 3.60. Zudem ist die Schwingung leicht chaotisch, was in Abbildung 3.61 deutlich wird. Diese Ausbildung der Schwingung geht vermutlich auf den geringen Winddruck zurück.

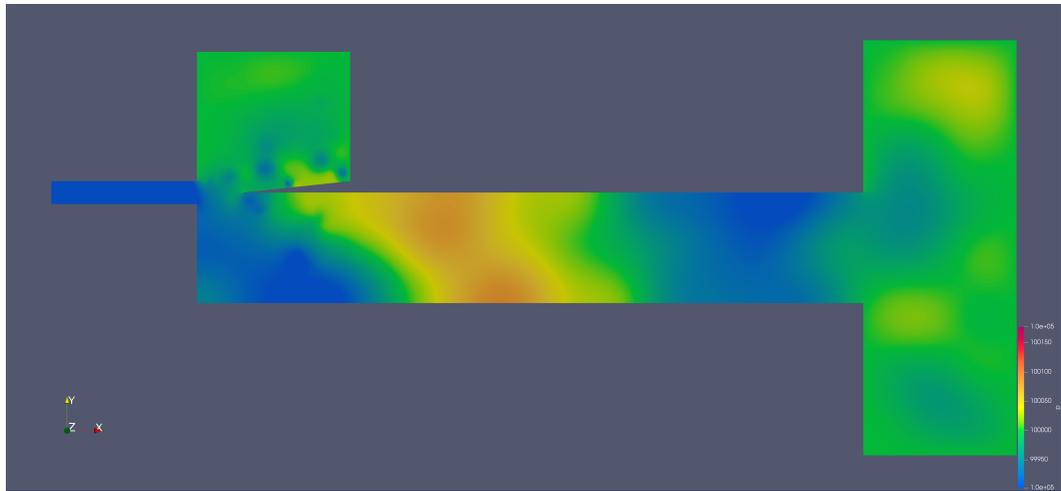


Abbildung 3.60.: Visualisierung des Drucks bei 0,00987s, offene Pfeife mit geringem Druck angeblasen

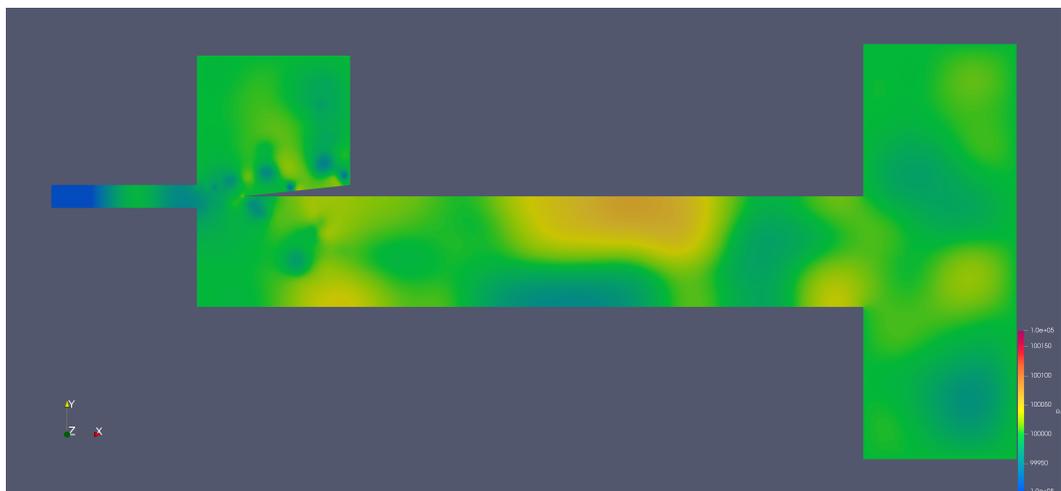


Abbildung 3.61.: Visualisierung des Drucks bei 0,00993s, offene Pfeife mit geringem Druck angeblasen

## Schmale Pfeife

Analog der Vorgehensweise bei der gedackten Pfeife wurde auch die offene Pfeife in einer schmalen Variante mit einem Durchmesser von 10,5 mm statt 14,5 mm simuliert.

Das Frequenzspektrum in Abbildung 3.62 zeigt eine interessante Verteilung der Frequenzen. So gibt es eine erste Spitze bei 1482 Hz, dem Grundton. Die nächste große Spitze folgt erst bei 4091 Hz, was dem 1,7ten Oberton entspricht.

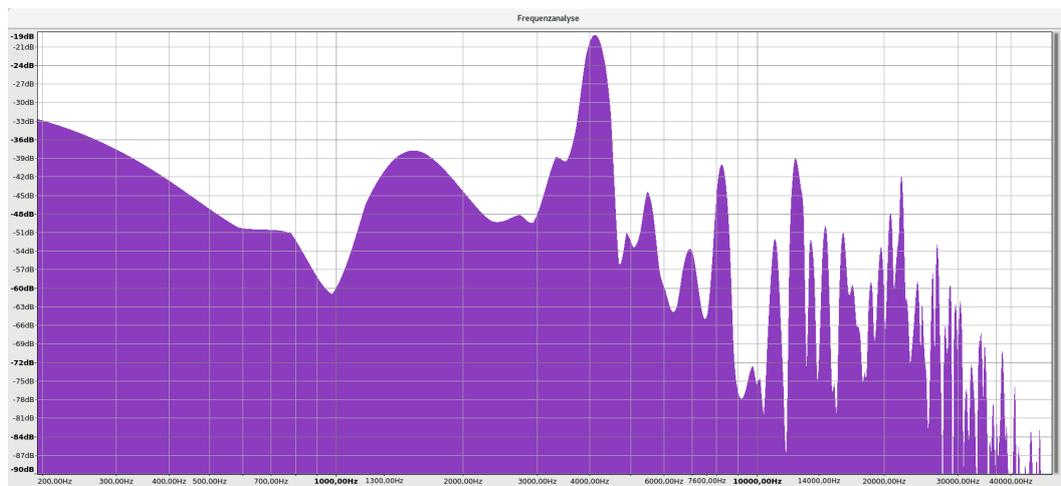


Abbildung 3.62.: Frequenzanalyse der schmalen, offenen Pfeife.

Die Darstellung als Kurve in Abbildung 3.63 zeigt nach dem Einschwingvorgang, der bis ca. 0,01 s andauert, eine Schwingung die annähernd sinusförmig ist.

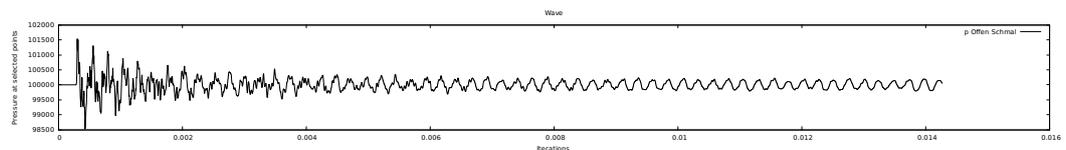


Abbildung 3.63.: Die erzeugte Schwingung der schmalen, offenen Pfeife.

## Vergleich

Auch hier finden sich im Vergleich sowohl Gemeinsamkeiten, als auch Unterschiede. Deutlich werden diese anhand der Kurvendiagramms, Abbildung 3.64, in dem die Kurven übereinander gelegt sind. Es fällt aus, dass die mit geringem Druck angeblasene Pfeife weniger stark schwingt als die beiden anderen Pfeifen. Die Einschwingvorgänge ähneln sich, die erzeugten Spitzen zum Anfang des Diagramms sind bei allen drei Kurven vorhanden.

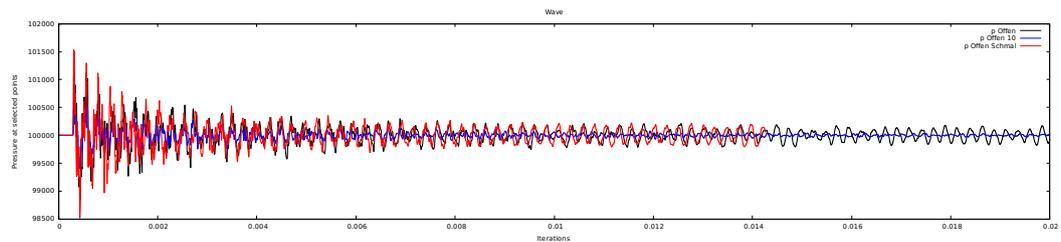


Abbildung 3.64.: Vergleich der erzeugten Schwingungen in den 3 verschiedenen Versionen der offenen Pfeife. Schwarz die unveränderte, gedackte Pfeife, blau die mit geringen Winddruck angeblasene Pfeife, rot die schmale offene Pfeife.

Alle drei Pfeifen zeigen doppelte Wirbel im Pfeifenkörper, wobei diese Wirbel unterschiedlich stark ausgeprägt sind. Unterschiede im Druck zeigen sich zum Einen in der Intensität, aber auch in der Form der Schwingungsknoten. Ausgangspunkt der Betrachtungen ist Abbildung 3.65. Die Wirbel sind deutlich als Druckminima im linken Bereich des Pfeifenkörpers zu sehen. Auf der rechten Seite ist ein Maximum ausgeprägt. Ein ähnliches Bild zeigt sich beim Vergleich mit der schmalen Pfeife in Abbildung 3.66. Auch dort sind die Wirbel zu sehen und ein Maximum auf der rechten Seite zum Ende der Pfeife hin. Etwas anders sieht das Maximum bei der mit geringerem Druck angeblasenen Pfeife in Abbildung 3.67 aus. Es fällt auf, dass dieses Maximum nicht rechtwinklig zur Pfeifenwand ausgeprägt ist.

### 3.3.3. Vergleich offen und gedackt

Die Unterschiede zwischen einer offenen und einer gedackten Pfeife sollten sich, so die Vermutung, auch in der Simulation manifestieren. Offensichtliches Merkmal ist dabei, dass die von offenen Pfeifen erzeugte Frequenz doppelt so groß ist wie die von gedackten Pfeifen erzeugte Frequenz.

Es ist aus Tabelle 3.1 ersichtlich, dass die offenen Pfeifen mit einer höheren Frequenz schwingen als die gedackten Pfeifen. Eine annähernde Frequenzverdopplung gibt es jedoch nur bei der mit geringerem Druck angeblasenen Pfeife.

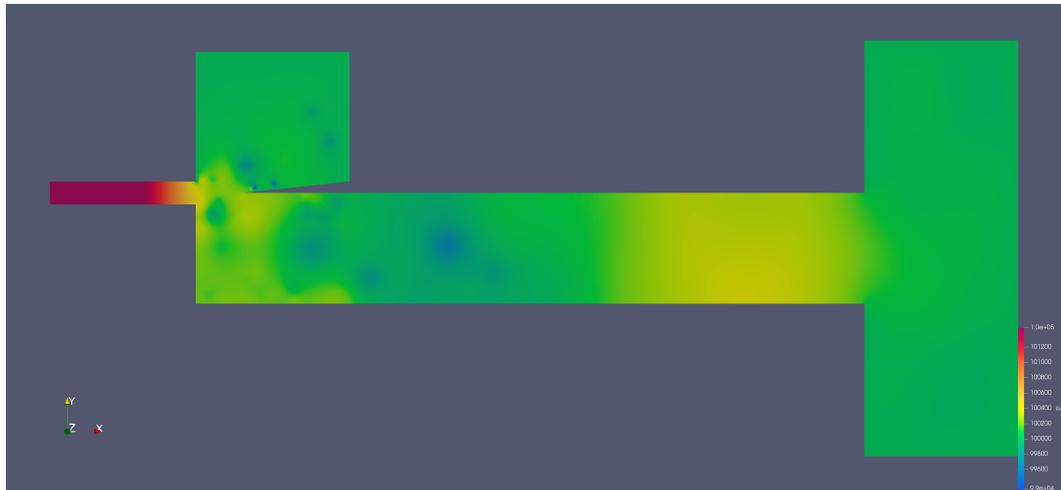


Abbildung 3.65.: Visualisierung des Drucks bei 0,01255 s

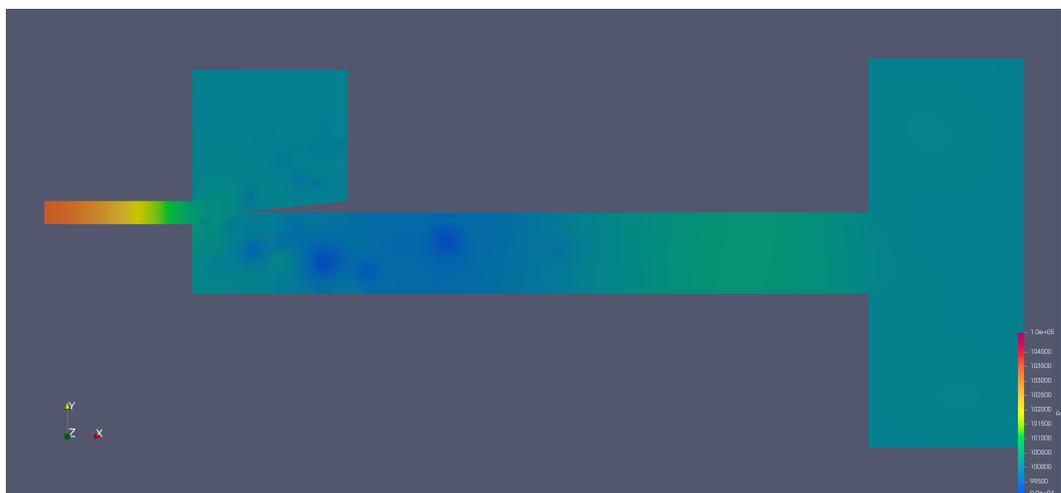


Abbildung 3.66.: Visualisierung des Drucks bei 0,01288 s, schmale, offene Pfeife

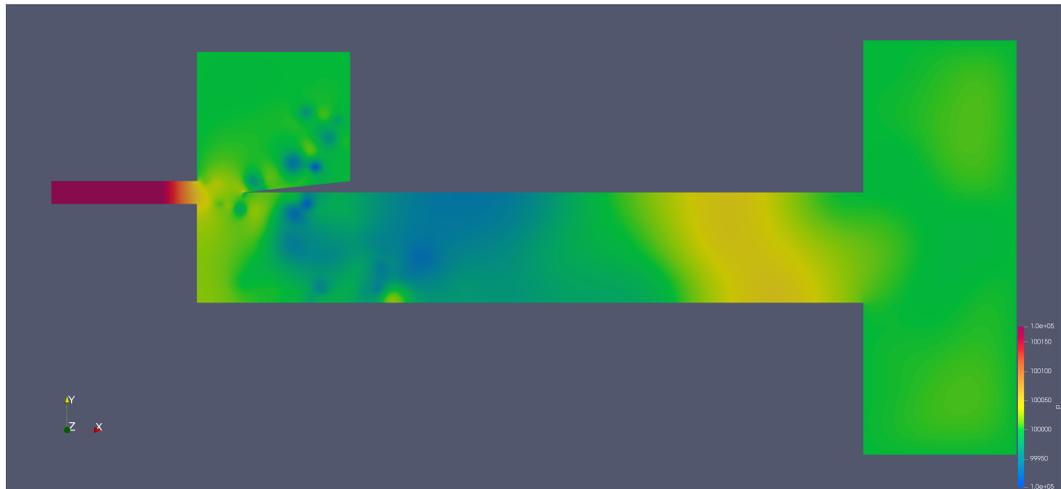


Abbildung 3.67.: Visualisierung des Drucks bei 0,01843 s, offene Pfeife mit geringem Druck angeblasen

Nichtsdestoweniger zeigen die Ergebnisse, dass die Simulation auch diese Eigenschaft von Orgelpfeifen nachbildet.

Tabelle 3.1.: Übersicht der Frequenzen der Grundtöne der simulierten Orgelpfeifen

Variante	Gedackt	Offen
unverändert	912 Hz	1473 Hz
geringer Winddruck	763 Hz	1619 Hz
schmale Pfeife	920 z	1382 Hz

### 3.3.4. Vergleich Simulation und Realität

Den Abschluss der Analyse bildet der Vergleich zwischen den simulierten Pfeifen und dem Vorbild. Zum Vergleich eignen sich dabei sowohl Messwerte aus anderen Arbeiten als auch der Vergleich der erzeugten Schwingung.

#### Darstellung

Die Visualisierung der Strömung erlaubt den Vergleich mit den Darstellungen aus den Forschungsergebnissen von Janke (1997). Ausgangspunkt für die Betrachtung sind die in den Abbildungen 2.8 und 2.11 dargestellten Strömungen.

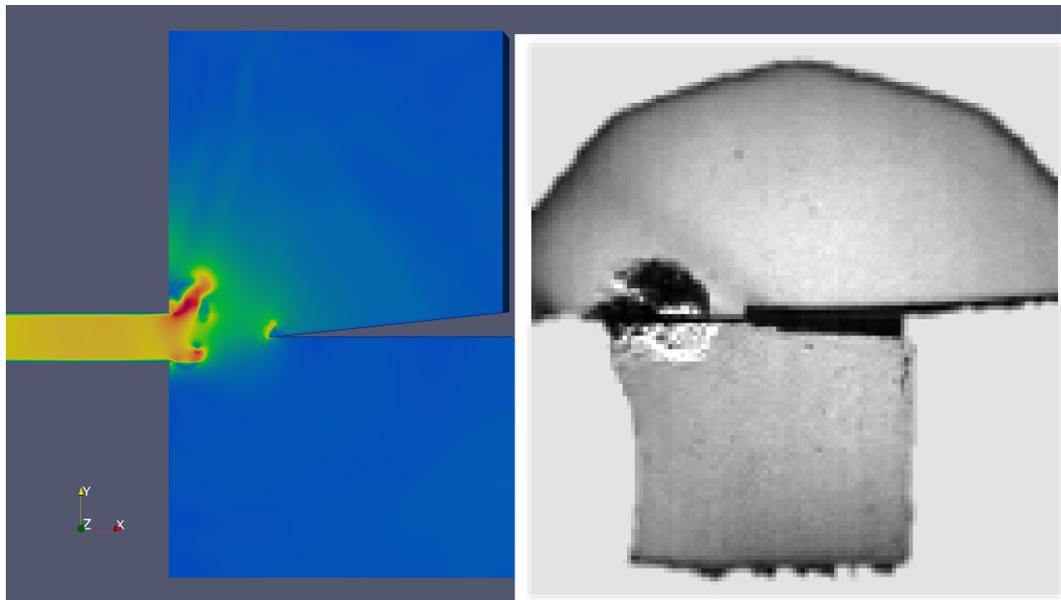


Abbildung 3.68.: Visualisierung und Vergleich der Strömung bei 0,00041 s

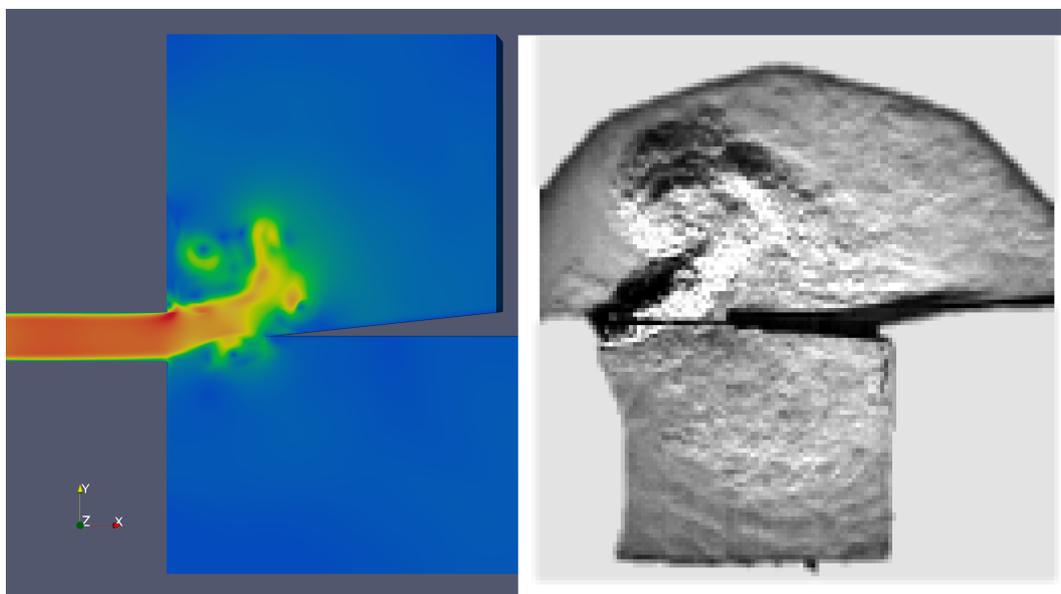


Abbildung 3.69.: Visualisierung und Vergleich der Strömung bei 0,00164 s

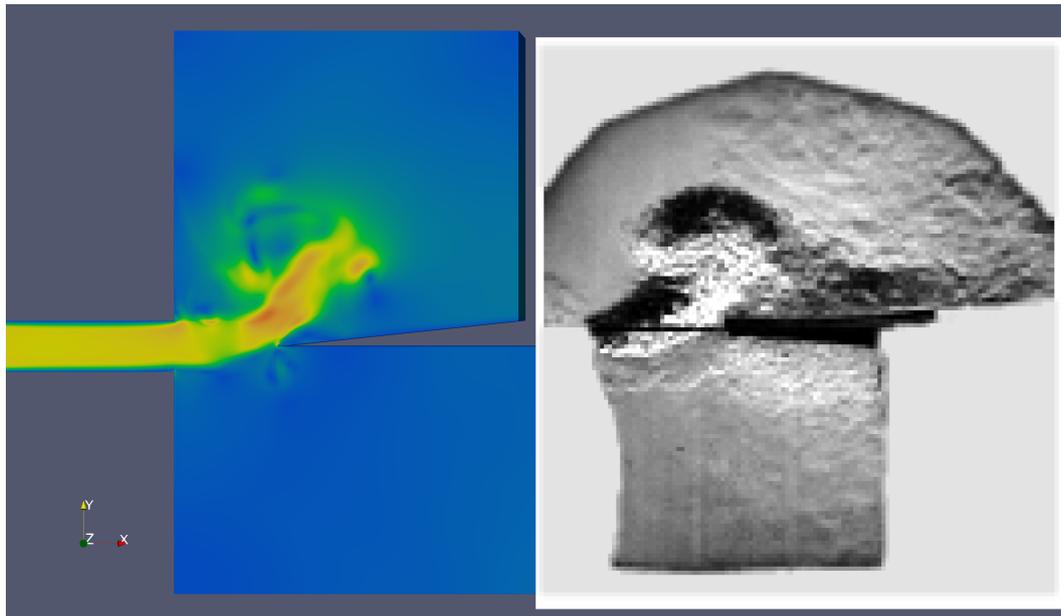


Abbildung 3.70.: Visualisierung und Vergleich der Strömung bei 0,00240 s

Obgleich die Bilder aufgrund der zum damaligen Zeitpunkt verwendeten Aufnahmetechnik nur sehr gering auflösen, können dennoch gut die Strömungen verglichen werden. Zur Verdeutlichung wurden Simulation und Messung nebeneinander kopiert. Abbildung 3.68 zeigt den Beginn des Einschwingvorganges. Die Strömung breitet sich aus der Kernspalte nach oben und unten aus. Kurze Zeit später zeigt sich in Abbildung 3.69 die Ausbildung des Wirbels vor dem Pfeifenmund. Es ist zu sehen, dass die Strömung eine Kurvenform beschreibt. Die in der Aufnahme zu sehende Strömung in der Pfeife ist aufgrund des geringeren Kontrastes der Visualisierung der Simulation an dieser Stelle nur schwer zu erkennen. Im dritten Vergleichsbild in Abbildung 3.70 ist zu sehen, dass sich der Wirbel vor der Pfeife weiter manifestiert. Dies zeigt auch der Vergleich der Simulation mit den Messwerten.

Zur besseren Vergleichbarkeit wurde in Abbildung 3.71 die Visualisierung aus Abbildung 2.9 gespiegelt und so ausgerichtet, dass sie der Ausrichtung der Simulation entspricht. Für die Visualisierung der Simulation in Abbildung 3.72 wurde die Farbskala analog gewählt sodass ein Vergleich beider Graphiken leichter fällt. Deutlich ist der Wirbel in der Pfeife zu sehen, der von der Luftströmung erzeugt wird, wenn sie in die Pfeife strömt. In der Simulation fällt der Wirbel größer aus als in der gemessenen Pfeife da der Durchmesser der simulierten Pfeife größer und die Kernspalte breiter ist.

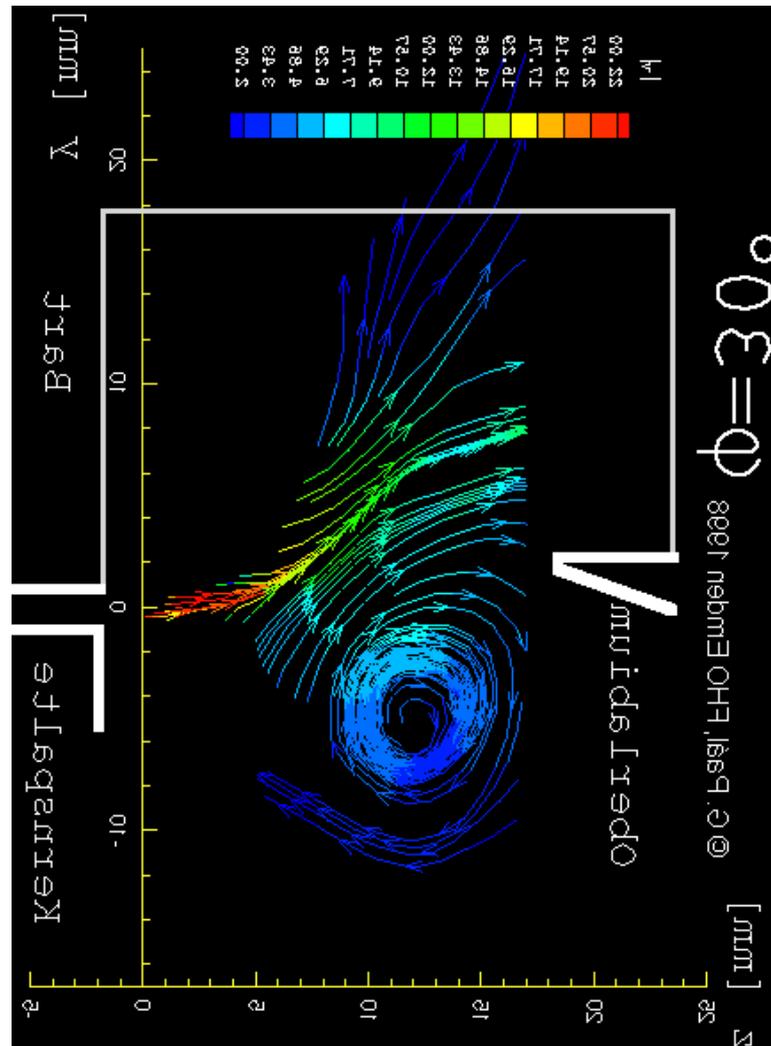


Abbildung 3.71.: Visualisierte Messwerte der Strömung am Pfeifenmund. Bild aus Janke (1997), zur besseren Vergleichbarkeit gespiegelt und wie Abbildung 3.72 ausgerichtet

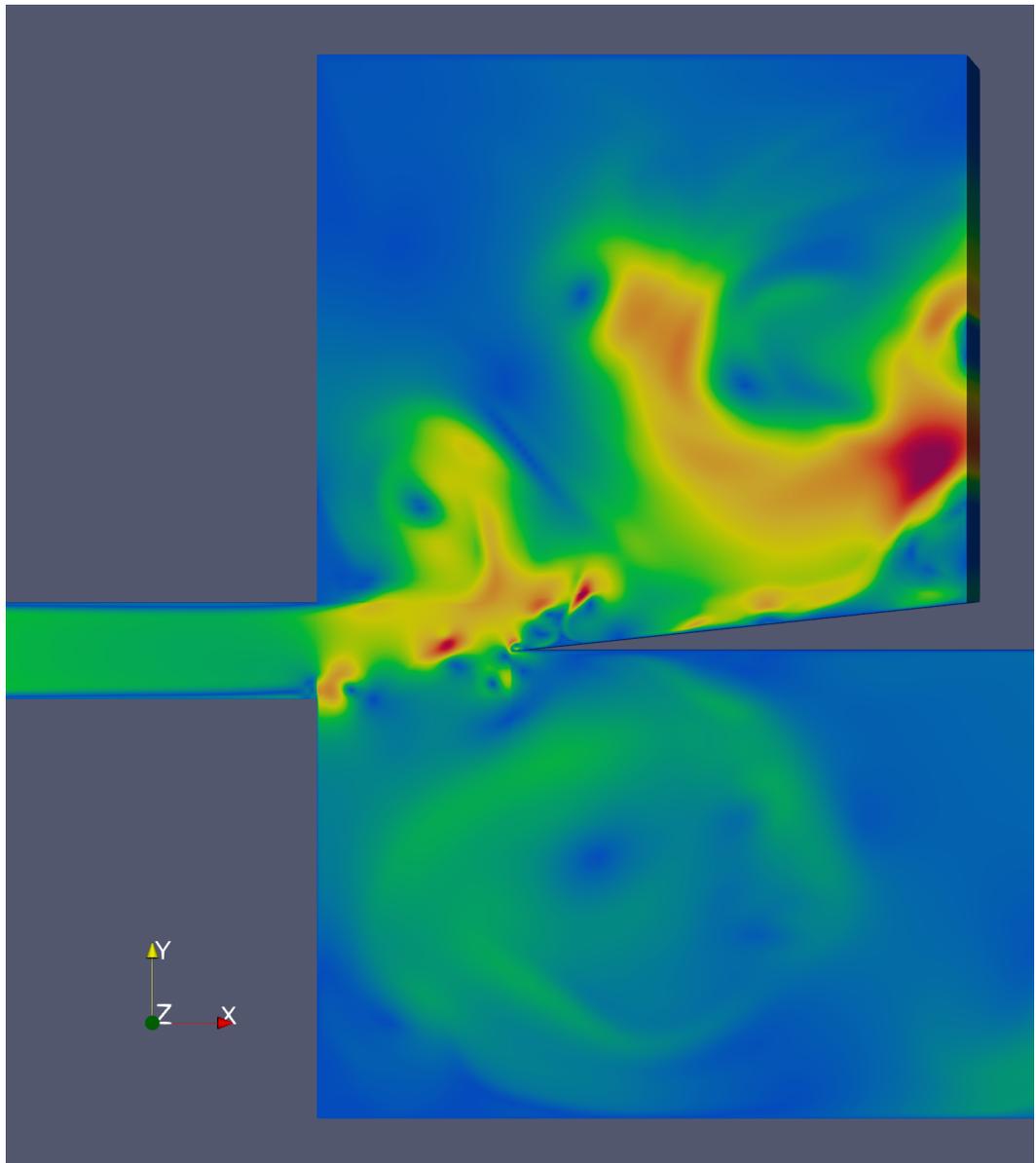


Abbildung 3.72.: Visualisierung der Strömung bei 0,02s

## Spektrum

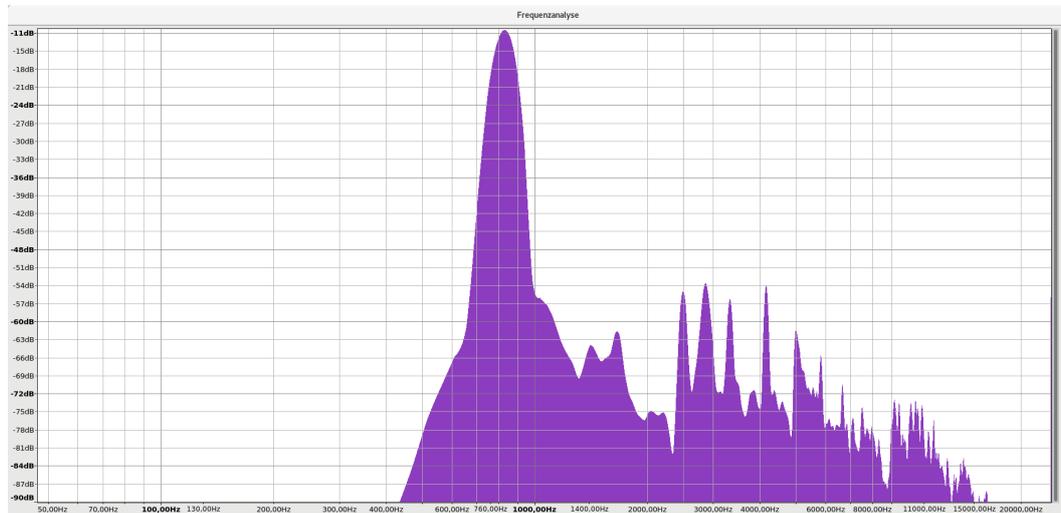


Abbildung 3.73.: Spektrum einer gedackten Orgelpfeife auf dem Ton gis2.

Die im Abschnitt 2.2.1 gezeigten Spektren, Abbildungen 2.4 und 2.5, zeigen ein sehr reines Spektrum ohne störende Frequenzen die das Spektrum verfälschen. Damit konnten der Aufbau des Spektrums und die Theorie der Obertöne gut gezeigt werden. Ein anderes Bild zeigt das Frequenzspektrum einer gedackten Pfeife in Abbildung 3.73. Der Ton gis2 wurde gewählt da die simulierte Pfeife ungefähr diesen Ton erzeugt, auf die Analyse hat das keinen hierfür relevanten Einfluss. Die Obertöne sind bei weitem nicht so klar erkennbar wie das bei den anderen Spektren der Fall war, es gibt viele störende Frequenzen. Dies ist darauf zurückzuführen dass die Pfeife leicht rauscht, eine Eigenschaft gedackter Pfeifen die nicht immer unerwünscht ist.

Legt man dieses Frequenzspektrum zum Vergleich skaliert über das Frequenzspektrum auf Abbildung 3.46 so entsteht Abbildung 3.74. Es ist zu sehen, dass die Spektren einander ähnlich sehen und somit ein für eine gedackte Pfeife typisches Frequenzspektrum zeigen. In der simulierten Pfeife sind lediglich der Grundton nicht so dominant gegenüber den restlichen Obertönen wie bei der realen Pfeife. Damit ist ersichtlich, dass in der Simulation tatsächlich eine gedackte Orgelpfeife zum Klingen gebracht wurde.

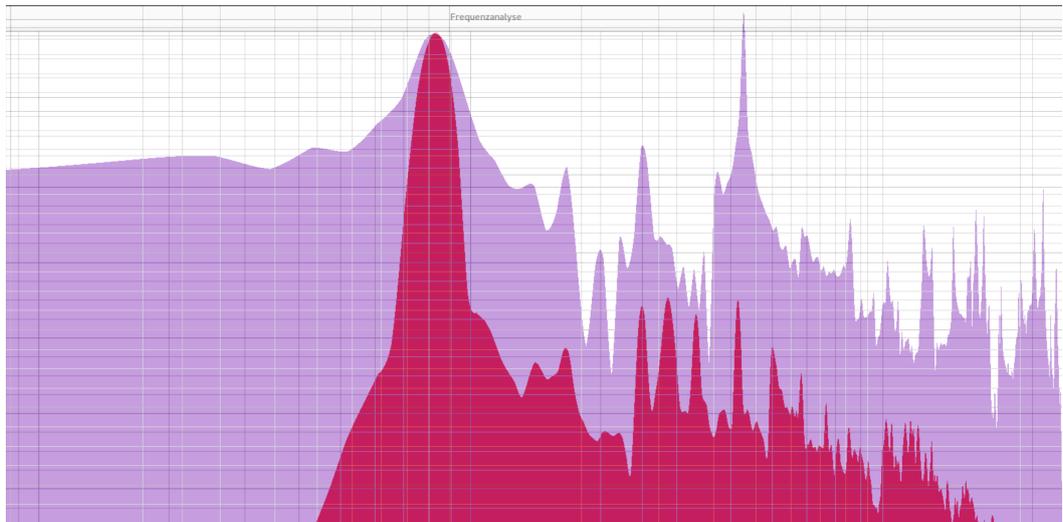


Abbildung 3.74.: Vergleich der Frequenzspektren der simulierten (hellblau) und einer realen Pfeife (sattrot).

### Berechnung der Frequenz

Nach der Bestimmung der Frequenz über die Frequenzanalyse kann diese auch aus der Länge der Pfeife berechnet werden. Die dafür verwendete Formel für gedackte Pfeifen lautet wie im Abschnitt 2.2.1 vorgestellt

$$f = \frac{c}{4l}$$

Die Länge der Pfeife beträgt  $l = 87\text{mm}$ , die Schallgeschwindigkeit entsprechend der Parameter der Simulation  $c = 343\frac{\text{m}}{\text{s}}$ . Diese Werte werden nun in die Gleichung eingesetzt:

$$\begin{aligned} f &= \frac{c}{4l} \\ f &= \frac{343\frac{\text{m}}{\text{s}}}{4 \cdot 0,087\text{m}} \\ f &= \frac{343\frac{\text{m}}{\text{s}}}{0,348\text{m}} \\ f &= 985\frac{1}{\text{s}} = 985\text{Hz} \end{aligned}$$

Dem gegenüber steht die in der Frequenzanalyse bestimmte Grundfrequenz von 920 Hz. Hier gibt es eine Diskrepanz deren Ursachen vielfältig sein können. So wird in OpenFOAM die Schallgeschwindigkeit aus den Parametern

der Simulation abgeleitet, während die Berechnung in der Formel die Schallgeschwindigkeit exakt berechnet. In OpenFOAM wird aber die Temperatur nachgebildet, sodass es sein kann, dass die Temperatur in der Simulation gesunken ist und dadurch die Pfeife langsamer schwingt als berechnet. Ein andere Möglichkeit ist, dass die Pfeife mit zu wenig Druck angeblasen wurde. Wie beim Vergleich der drei gedackten Pfeifen ersichtlich, schwingt die mit geringem Druck angeblasene Pfeife deutlich langsamer als die anderen beiden Pfeifen. Die Formel zur Berechnung der Frequenz berücksichtigt dies jedoch nicht.

### Einschwingvorgang

In Abbildung 3.75 ist der Einschwingvorgang der Simulation und einer realen Orgelpfeife zu sehen. Auch wenn die Grundfrequenz gleich ist fällt deutlich die unterschiedlich Wellenform der beiden Pfeifen auf obgleich beide Schwingungen die gleiche Grundfrequenz haben. Der Grund dafür sind wie der Vergleich der Spektren in Abbildung 3.75 bereits zeigte die deutlich stärkeren Obertöne.

Des Weiteren ist deutlich zu sehen, dass sich der Einschwingvorgang unterscheidet. So steigt bei der simulierten Pfeife der Druck schlagartig an, während die reale Pfeife leise anschwingt. Beiden Pfeifen gemein ist jedoch, dass es einige Zeit dauert bis die Schwingung stabil ist und nur noch die gewünschten Frequenzen erzeugt.

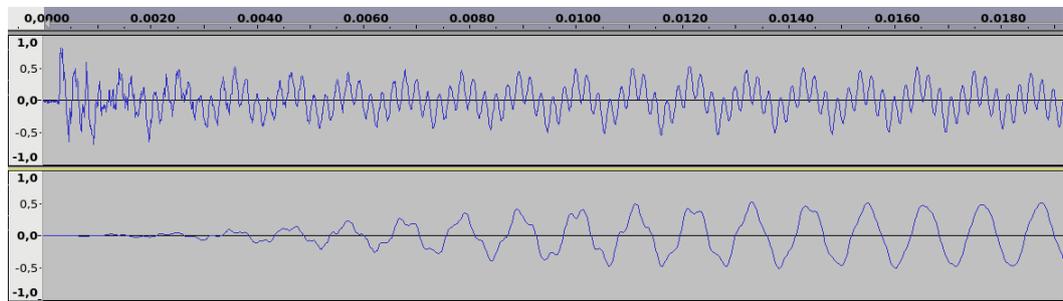


Abbildung 3.75.: Visualisierung der Wellen der Simulation (oben) und einer realen, gedackten Orgelpfeife (unten)

## 4. Nachbetrachtung

### 4.1. Einschränkungen der Simulation

Die hier vorgestellte Simulation unterliegt einigen Einschränkungen, die bei der Betrachten der Ergebnisse beachtet werden müssen.

#### 4.1.1. Zweidimensionale Darstellung

Die Entscheidung, ob eine Orgelpfeife einen runden oder einen eckigem Querschnitt besitzt, wird üblicherweise aufgrund des verwendeten Materiales getroffen. So hat es sich im Orgelbau durchgesetzt, dass Holzpfeifen mit eckigem Querschnitt, Metallpfeifen hingegen mit rundem Querschnitt gebaut werden. Ein eckiger Querschnitt erleichtert dabei das Zusammensetzen der Pfeife aus vier Holzleisten, ein runder Querschnitt das Löten der runden Pfeife. Auf den Klang der Pfeife hat die Form jedoch keinen Einfluss<sup>1</sup>. Es ist sogar nicht unüblich, innerhalb eines Registers beide Formen zu verbauen. Da die sogenannte Normmensur<sup>2</sup> auf der Querschnittsfläche aufbaut ist es möglich, den benötigten Durchmesser für einen über alle Töne gleichbleibenden Klang zu berechnen.

In der Simulation wurde die Pfeife jedoch zweidimensional modelliert<sup>3</sup>. Das bedeutet, dass die Querschnittsfläche unendlich groß ist. Des Weiteren wird aufgrund der fehlenden, dritten Dimension das Labium nur unvollständig modelliert. So betont beispielsweise ein im Verhältnis zum Pfeifendurchmesser schmales Labium vor allem den zweiten Oberton. Um das Anspracheverhalten zu verändern werden Intonationsmittel wie beispielsweise Kernstiche genutzt, auch seitlich des Labiums angebrachte Platten, so genannte Bärte, beeinflussen die Strömung am Pfeifenmund. Auch ist es möglich, Kernspalte und Labium nicht rechtwinklig zum Pfeifenkörper anzuordnen, sondern schräg<sup>4</sup>. Diese Maßnahmen und die daraus resultierenden Unterschiede in der Intonation lassen sich in einer zweidimensionalen Simulation jedoch keineswegs nachbilden.

---

<sup>1</sup>Siehe Brackhane (2018).

<sup>2</sup>Siehe Brüner (2002).

<sup>3</sup>OpenFOAM kennt keine Unterscheidung zwischen 2D und 3D. Wird jedoch in z-Richtung nur eine Zelle definiert wird keine Berechnung in z-Richtung durchgeführt.

<sup>4</sup>Siehe Wand (2018).

Somit kann eine zweidimensionale Simulation lediglich eine Annäherung darstellen.

Prinzipiell wäre eine dreidimensionale Simulation möglich, ist jedoch sehr zeitaufwändig. Die gedackte Orgelpfeife aus Ansatz 3 besteht aus 193800 Zellen, der Querschnitt des Pfeifenkörpers besteht aus 720 Zellen. Wird die Simulation in z-Richtung erweitert so ist zu erwarten, dass in diese Richtung auch mindestens 720 Zellen benötigt werden. Dabei nicht berücksichtigt ist, dass für die dreidimensionale Ausbreitung des Schalls vor der Pfeife dieser Bereich breiter als die Pfeife simuliert werden muss und damit dort auch mehr als 720 Zellen in dritter Dimension berechnet werden müssen. Die Berechnung von 0,2 Sekunden Pfeife benötigte auf einem Intel Core i7-8650U mit 4 Kernen á 1,90 GHz etwa 200 Stunden. Es ist leicht ersichtlich, dass die zusätzliche Berechnung der dritten Dimension die benötigte Zeit auf mehrere Jahre verlängert.

Im Vergleich dazu fällt der benötigte Speicherplatz kaum ins Gewicht. Während die in dieser Arbeit verwendeten, zweidimensionalen Beispiele, jeweils etwa 50 GB Speicherplatz für 0,2 Sekunden Simulationsdaten benötigen vervielfacht sich auch dieser Wert bei einer dreidimensionalen Berechnung. Eine Reduktion der gespeicherten Zeitschritte ist dabei im Sinne der Forschung nicht praktikabel da rein aufgrund der erzeugten Messwerte an ausgewählten Stellen keine sinnvolle Analyse und Darstellung möglich ist.

#### **4.1.2. Beschränkung auf die Simulation der Luft.**

Gegenstand der Simulation ist die Luft, die sich in der Pfeife befindet. Die Wände der Pfeife sind jedoch feststehend und unveränderbar simuliert. Es ist jedoch zunächst unbestreitbar, dass der besonders von tiefen Orgelpfeifen erzeugte Schall nicht nur hörbar, sondern auch spürbar ist. Dieser Effekt kann mit der hier vorgestellten Simulation jedoch nicht simuliert werden. Es wäre jedoch von Interesse, die in Bergweiler (2005) gezeigten Erkenntnisse auch in der Simulation nachvollziehen zu können. Dazu muss jedoch auch die Wand der Orgelpfeife simuliert werden.

Damit einher geht die Beschränkung der Simulation auf Labialpfeifen. Wie in Abschnitt 2.1.5 beschrieben gibt es noch eine zweite Gruppe von Orgelpfeifen, die Lingualpfeifen, bei denen der Ton durch eine schnarrende Zunge erzeugt wird. Um eine solche Pfeife vollständig simulieren zu können, muss die durch die Luft bewegte Zunge simuliert werden.

## 4.2. Zukünftige Entwicklung

Derzeit werden Orgeln am PC nur über die CPU berechnet. Auch große, spezialisierte Hersteller setzen inzwischen nicht mehr auf integrierte Schaltungen sondern nutzen PC-Komponenten zur Berechnung des Klanges<sup>5</sup>. Bei Anwendungen die viel Rechenleistung benötigen ist heutzutage jedoch eine Berechnung auf der GPU üblich. Standards für die Berechnung auf Graphikkarten wie z.B. CUDA oder OpenCL sind seit 2008 verfügbar. Dass diese vor allem für hochparallele Anwendungen geeignet sind ist dabei kein Nachteil. Eine mögliche Aufteilung der Parallelität ist dabei die Zuordnung, dass ein Register pro Graphikkarte simuliert wird. Die gegenseitige Beeinflussung der Pfeifen untereinander könnte dabei zunächst vernachlässigt werden um den Datenaustausch zwischen den Karten auf ein Minimum zu reduzieren.

Setzt man das Ziel, den Ton in Echtzeit zu berechnen, so wie z.B. bei der Software Aeolus, so muss genug Rechenleistung vorgehalten werden um auch ein schnell gespieltes Tutti ohne Verzögerung simulieren zu können. Damit unterscheidet sich die Pfeifenorgelsimulation deutlich von einer Simulation einer Hammondorgel wie der HX3<sup>6</sup>. Die Anzahl der Tonräder der simulierten Hammond B3 ist konstant 91 und alle Tonräder drehen sich unabhängig der gedrückten Tasten und eingestellten Zugriegeln. Bei einer Pfeifenorgel hingegen schwankt die Anzahl der aktiven Pfeifen je nach gezogenen Registern und Fähigkeiten des Organisten zwischen einigen wenigen bei leisen Stücken bis hin zu mehreren hundert wenn die Orgel im Tutti gespielt wird. Das bedingt, dass für eine solche Simulation Rechenleistung vorgehalten werden muss, die oft nur kurzzeitig benötigt wird.

Unabhängig der Verwendung für eine Tonwiedergabe in Echtzeit kann die Simulation wertvolle Arbeit im Orgelbau leisten. So sind bei Bau und Intonation von Orgeln sehr viel Wissen und Erfahrung notwendig, damit die Pfeife genau so klingt wie vorgesehen. Eine solche Simulation stellt hierbei die Möglichkeit zur Verfügung, die Pfeife vor dem Bau in der Simulation zu modellieren und zum Klingen zu bringen. Somit kann bereits im Vorfeld überprüft werden, ob der Klang der Pfeife den Erwartungen entspricht. Zusammen mit den in Wand (2018) gezeigten Methoden des 3D-Druckes für den Orgelbau kann dies Orgelbauern neue Technologien zur Verfügung stellen, die das alte, traditionelle Handwerk wirkungsvoll ergänzen können.

---

<sup>5</sup>Siehe Viscount (2019).

<sup>6</sup>Siehe Kraus (2012).

# Anhang A.

## Listings

Listing A.1: Die Quelltextdatei rhoPimpleFoam.C aus dem Quelltext von OpenFOAM v18.12 in welcher der Solver rhoPimpleFoam definiert ist.

```
1
2 /*-----*\
3 =====/
4 || / Field / OpenFOAM: The Open Source CFD
5 Toolbox
6 || / Operation /
7 || / And / Copyright (C) 2011–2017
8 || / OpenFOAM Foundation
9 || / Manipulation /
10
11
12 License
13 This file is part of OpenFOAM.
14
15 OpenFOAM is free software: you can redistribute it
16 and/or modify it
17 under the terms of the GNU General Public License as
18 published by
19 the Free Software Foundation, either version 3 of the
20 License, or
21 (at your option) any later version.
22
23 OpenFOAM is distributed in the hope that it will be
24 useful, but WITHOUT
25 ANY WARRANTY; without even the implied warranty of
26 MERCHANTABILITY or
27 FITNESS FOR A PARTICULAR PURPOSE. See the GNU General
28 Public License
29 for more details.
```

---

```

21
22     You should have received a copy of the GNU General
23     Public License
24     along with OpenFOAM. If not, see
25     <http://www.gnu.org/licenses/>.
26
27
28 Application
29     rhoPimpleFoam
30
31 Group
32     grpCompressibleSolvers
33
34 Description
35     Transient solver for turbulent flow of compressible
36     fluids for HVAC and
37     similar applications, with optional mesh motion and
38     mesh topology changes.
39
40     Uses the flexible PIMPLE (PISO–SIMPLE) solution for
41     time–resolved and
42     pseudo–transient simulations.
43
44 |*-----
45
46 #include "fvCFD.H"
47 #include "dynamicFvMesh.H"
48 #include "fluidThermo.H"
49 #include "turbulentFluidThermoModel.H"
50 #include "bound.H"
51 #include "pimpleControl.H"
52 #include "pressureControl.H"
53 #include "CorrectPhi.H"
54 #include "fvOptions.H"
55 #include "localEulerDdtScheme.H"
56 #include "fvcSmooth.H"
57
58 // * * * * *
59 // * * * * * //
60
61 int main(int argc, char *argv[])
62 {
63     argList::addNote
64     (

```

---

```

58         "Transient_solver_for_compressible_turbulent_
           flow.\n"
59         "With_optional_mesh_motion_and_mesh_topology_
           changes."
60     );
61
62     #include "postProcess.H"
63
64     #include "addCheckCaseOptions.H"
65     #include "setRootCaseLists.H"
66     #include "createTime.H"
67     #include "createDynamicFvMesh.H"
68     #include "createDyMControls.H"
69     #include "initContinuityErrs.H"
70     #include "createFields.H"
71     #include "createFieldRefs.H"
72     #include "createRhoUIfPresent.H"
73
74     turbulence->validate();
75
76     if (!LTS)
77     {
78         #include "compressibleCourantNo.H"
79         #include "setInitialDeltaT.H"
80     }
81
82     // * * * * *
83     * * * * * //
84
85     Info<< "\nStarting_time_loop\n" << endl;
86
87     while (runTime.run())
88     {
89         #include "readDyMControls.H"
90
91         // Store divrhoU from the previous mesh so that it
92         // can be mapped
93         // and used in correctPhi to ensure the corrected
94         // phi has the
95         // same divergence
96         autoPtr<volScalarField> divrhoU;
97         if (correctPhi)
98         {
99             divrhoU.reset

```

---

```

97         (
98             new volScalarField
99             (
100                 "divrhoU" ,
101                 fvc::div(fvc::absolute(phi, rho, U))
102             )
103         );
104     }
105
106     if (LTS)
107     {
108         #include "setRDeltaT.H"
109     }
110     else
111     {
112         #include "compressibleCourantNo.H"
113         #include "setDeltaT.H"
114     }
115
116     ++runTime;
117
118     Info<< "Time=_" << runTime.timeName() << nl <<
119         endl;
120
121     // — Pressure-velocity PIMPLE corrector loop
122     while (pimple.loop())
123     {
124         if (pimple.firstIter() ||
125             moveMeshOuterCorrectors)
126         {
127             // Store momentum to set rhoUf for
128             // introduced faces.
129             autoPtr<volVectorField> rhoU;
130             if (rhoUf.valid())
131             {
132                 rhoU.reset(new volVectorField("rhoU",
133                     rho*U));
134             }
135
136             // Do any mesh changes
137             mesh.update();
138
139             if (mesh.changing())
140             {

```

---

```

137         MRF.update();
138
139         if (correctPhi)
140         {
141             // Calculate absolute flux
142             // from the mapped surface velocity
143             phi = mesh.Sf() & rhoUf();
144
145             #include "correctPhi.H"
146
147             // Make the fluxes relative to the
148             // mesh-motion
149             fvc::makeRelative(phi, rho, U);
150         }
151
152         if (checkMeshCourantNo)
153         {
154             #include "meshCourantNo.H"
155         }
156     }
157
158     if (pimple.firstIter() && !pimple.SIMPLERho())
159     {
160         #include "rhoEqn.H"
161     }
162
163     #include "UEqn.H"
164     #include "EEqn.H"
165
166     // — Pressure corrector loop
167     while (pimple.correct())
168     {
169         if (pimple.consistent())
170         {
171             #include "pcEqn.H"
172         }
173         else
174         {
175             #include "pEqn.H"
176         }
177     }
178
179     if (pimple.turbCorr())

```

```

180         {
181             turbulence->correct ();
182         }
183     }
184
185     rho = thermo.rho ();
186
187     runTime.write ();
188
189     runTime.printExecutionTime (Info);
190 }
191
192 Info << "End\n" << endl;
193
194 return 0;
195 }
196
197 // *****
    ***** //

```

Listing A.2: Die Quelltextdatei `pimpleControl.C` aus dem Quelltext von OpenFOAM v18.12 in welcher der PIMPLE-Algorithmus implementiert ist.

```

1  /*-----*|
2  =====|
3  || / Field | OpenFOAM: The Open Source CFD
4  Toolbox
5  || / Operation |
6  || / And | Copyright (C) 2011–2017
7  OpenFOAM Foundation
8  ||/ Manipulation | Copyright (C) 2017 OpenCFD
9  Ltd.
10
11 License
12 This file is part of OpenFOAM.
13
14 OpenFOAM is free software: you can redistribute it
15 and/or modify it
16 under the terms of the GNU General Public License as
17 published by
18 the Free Software Foundation, either version 3 of the

```



---

```

46     nCorrPISO_ =
47         pimpleDict.lookupOrDefault<label>("nCorrectors", 1);
48     SIMPLErho_ = pimpleDict.lookupOrDefault("SIMPLErho",
49         false);
50     turbOnFinalIterOnly_ =
51         pimpleDict.lookupOrDefault("turbOnFinalIterOnly",
52         true);
53 }
54 bool Foam::pimpleControl::criteriaSatisfied()
55 {
56     // no checks on first iteration - nothing has been
57     // calculated yet
58     if ((corr_ == 1) || residualControl_.empty() ||
59         finalIter())
60     {
61         return false;
62     }
63     const bool storeIni = this->storeInitialResiduals();
64     bool achieved = true;
65     bool checked = false; // safety that some checks
66     // were indeed performed
67     const dictionary& solverDict =
68         mesh_.solverPerformanceDict();
69     forAllConstIters(solverDict, iter)
70     {
71         const entry& solverPerfDictEntry = *iter;
72         const word& fieldName =
73             solverPerfDictEntry.keyword();
74         const label fieldi = applyToField(fieldName);
75         if (fieldi != -1)
76         {
77             Pair<scalar> residuals =
78                 maxResidual(solverPerfDictEntry);
79             checked = true;
80         }

```

---

```

81         scalar relative = 0.0;
82         bool relCheck = false;
83
84         const bool absCheck =
85             (residuals.last() <
86              residualControl_[fieldi].absTol);
87
88         if (storeIni)
89         {
90             residualControl_[fieldi].initialResidual =
91                 residuals.first();
92         }
93         else
94         {
95             const scalar iniRes =
96                 (residualControl_[fieldi].initialResidual
97                  + ROOTVSMALL);
98
99             relative = residuals.last() / iniRes;
100            relCheck = (relative <
101                       residualControl_[fieldi].relTol);
102        }
103
104        achieved = achieved && (absCheck || relCheck);
105
106        if (debug)
107        {
108            Info << algorithmName_ << " loop:" << endl;
109
110            Info << "====" << fieldName
111                << " PIMPLE_iter_" << corr_
112                << ":_ini_res_="
113                <<
114                residualControl_[fieldi].initialResidual
115                << ",_abs_tol_=" << residualControl_[fieldi].absTol <<
116                << " (" <<
117                residualControl_[fieldi].absTol <<
118                ")"
119                << ",_rel_tol_=" << relative
120                << " (" <<
121                residualControl_[fieldi].relTol <<
122                ")"
123                << endl;
124        }

```

```

116     }
117 }
118
119     return checked && achieved;
120 }
121
122
123 void Foam::pimpleControl::setFirstIterFlag(const bool
      check, const bool force)
124 {
125     DebugInfo
126         << "corr:" << corr_
127         << "_corrPISO:" << corrPISO_
128         << "_corrNonOrtho:" << corrNonOrtho_
129         << endl;
130
131     solutionControl::setFirstIterFlag(check && corrPISO_ <=
      1, force);
132 }
133
134
135 // * * * * * Constructors * * * * *
      * * * * * //
136
137 Foam::pimpleControl::pimpleControl(fvMesh& mesh, const
      word& dictName)
138 :
139     solutionControl(mesh, dictName),
140     solveFlow_(true),
141     nCorrPIMPLE_(0),
142     nCorrPISO_(0),
143     corrPISO_(0),
144     SIMPLErho_(false),
145     turbOnFinalIterOnly_(true),
146     converged_(false)
147 {
148     read();
149
150     Info<< nl
151         << algorithmName_;
152
153     if (nCorrPIMPLE_ > 1)
154     {
155         if (residualControl_.empty())

```

```

156     {
157         Info<< " :_no_residual_control_data_found_"
158             << "Calculations_will_employ_" <<
                nCorrPIMPLE_
159             << "_corrector_loops" << nl;
160     }
161     else
162     {
163         Info<< " :_max_iterations_" << nCorrPIMPLE_ <<
                nl;
164
165         for (const fieldData& ctrl : residualControl_)
166         {
167             Info<< "    _field_" << ctrl.name <<
                token::TAB
168                 << " :_relTol_" << ctrl.relTol
169                 << " ,_tolerance_" << ctrl.absTol
170                 << nl;
171         }
172     }
173 }
174 else
175 {
176     Info<< " :_Operating_solver_in_PISO_mode" << nl;
177 }
178
179 Info<< endl;
180 }
181
182
183 // * * * * * * * * * * * * * * * * Member Functions * * * * *
184 // * * * * * * * * * * * * * * * * //
185
186 bool Foam::pimpleControl::loop()
187 {
188     read();
189
190     ++corr_;
191
192     if (debug)
193     {
194         Info<< algorithmName_ << "_loop:_corr_" << corr_
                << endl;
195     }

```

```
195
196     setFirstIterFlag();
197
198     if (corr_ == nCorrPIMPLE_ + 1)
199     {
200         if (!residualControl_.empty() && (nCorrPIMPLE_ !=
201             1))
202         {
203             Info<< algorithmName_ << ":_not_converged_
204                 within_"
205                 << nCorrPIMPLE_ << "_iterations" << endl;
206         }
207         corr_ = 0;
208         mesh_.data::remove("finalIteration");
209         return false;
210     }
211     bool completed = false;
212     if (converged_ || criteriaSatisfied())
213     {
214         if (converged_)
215         {
216             Info<< algorithmName_ << ":_converged_in_" <<
217                 corr_ - 1
218                 << "_iterations" << endl;
219             mesh_.data::remove("finalIteration");
220             corr_ = 0;
221             converged_ = false;
222
223             completed = true;
224         }
225         else
226         {
227             Info<< algorithmName_ << ":_iteration_" <<
228                 corr_ << endl;
229             storePrevIterFields();
230
231             mesh_.data::add("finalIteration", true);
232             converged_ = true;
233         }
234     }
235     else
```

```

235     {
236         if (finalIter())
237         {
238             mesh_.data::add("finalIteration", true);
239         }
240
241         if (corr_ <= nCorrPIMPLE_)
242         {
243             Info<< algorithmName_ << ":_iteration_" <<
                corr_ << endl;
244             storePrevIterFields();
245             completed = false;
246         }
247     }
248
249     return !completed;
250 }
251
252
253 // *****
    ***** //

```

Listing A.3: Die Datei thermophysicalProperties wie sie in den verschiedenen Ansätzen verwendet wurde.

```

1 FoamFile
2 {
3     version      2.0;
4     format       ascii;
5     class        dictionary;
6     location     "constant";
7     object       thermophysicalProperties;
8 }
9 // * * * * *
    * * * * * //
10
11 thermoType
12 {
13     type          hePsiThermo;
14     mixture       pureMixture;
15     transport     const;
16     thermo        eConst;
17     equationOfState perfectGas;
18     specie        specie;

```

```
19     energy          sensibleInternalEnergy;
20 }
21
22 mixture
23 {
24     specie
25     {
26         nMoles      1;
27         molWeight   28.9;
28     }
29     thermodynamics
30     {
31         Cv          712;
32         Hf          0;
33     }
34     transport
35     {
36         mu          1.8e-05;
37         Pr          0.7;
38     }
39 }
```

Listing A.4: Die Datei 0/T wie sie in den verschiedenen Ansätzen verwendet wurde.

```
1
2 FoamFile
3 {
4     version      2.0;
5     format       ascii;
6     class        volScalarField;
7     location     "0";
8     object       T;
9 }
10
11
12 dimensions [ 0 0 0 1 0 0 0 ];
13
14 internalField uniform 295; // 295 Kelvin = 22 Celsius
15
16 boundaryField
17 {
18     inlet
19     {
```

```
20     type fixedValue;
21     value uniform 295;
22 }
23
24 space
25 {
26     type inletOutlet;
27     inletValue uniform 295;
28     value uniform 295;
29 }
30
31 wall
32 {
33     type fixedValue;
34     value uniform 295;
35 }
36
37 frontAndBack
38 {
39     type empty;
40 }
41
42 default
43 {
44     type fixedValue;
45     value uniform 295;
46 }
47 }
```

Listing A.5: Die Datei blockMeshDict bei abfallendem Druck, Ansatz 2

```
1 FoamFile
2 {
3     version    2.0;
4     format     ascii;
5     class      dictionary;
6     object     blockMeshDict;
7 }
8
9 convertToMeters 0.001; //units are mm
10
11 vertices
12 (
13     (0 0 1) //0 A
```

---

```

14   (0 -0.5 1) //1 B
15   (0 -28 1) //2 C
16   (200 -28 1) //3 D
17   (200 0 1) //4 E
18
19   (7 1 1) //5 F
20   (34 6 1) //6 G
21   (500 6 1) //7 H
22   (500 106 1) //8 I
23   (34 106 1) //9 J
24
25   (7 106 1) //10 K
26   (0 106 1) //11 L
27   (-100 106 1) //12 M
28   (-100 0 1) //13 N
29   (0 6 1) //14 O
30
31   (7 -28 1) //15 P
32
33  //=====
34
35   (0 0 -1) //0 A
36   (0 -0.5 -1) //1 B
37   (0 -28 -1) //2 C
38   (200 -28 -1) //3 D
39   (200 0 -1) //4 E
40
41   (7 1 -1) //5 F
42   (34 6 -1) //6 G
43   (500 6 -1) //7 H
44   (500 106 -1) //8 I
45   (34 106 -1) //9 J
46
47   (7 106 -1) //10 K
48   (0 106 -1) //11 L
49   (-100 106 -1) //12 M
50   (-100 0 -1) //13 N
51   (0 6 -1) //14 O
52
53   (7 -28 -1) //15 P
54
55   (7 0 1) //32 Q
56   (7 0 -1) //33 Q
57 );

```

```
58
59 blocks
60 (
61   //A 0
62   hex (13 0 16 29 12 11 27 28) (50 1 51) simpleGrading (0.1
63     1 250)
64   //B 1
65   hex (0 5 21 16 11 10 26 27) (50 1 51) simpleGrading (1 1
66     250)
67   //C 2
68   hex (5 6 22 21 10 9 25 26) (60 1 51) simpleGrading (1 1
69     250)
70   //D 3
71   hex (6 7 23 22 9 8 24 25) (100 1 51) simpleGrading (10 1
72     250)
73   //E 4
74   hex (2 15 31 18 1 32 33 17) (50 1 80) simpleGrading (1 1
75     1)
76   //F 5
77   hex (15 3 19 31 32 4 20 33) (100 1 80) simpleGrading (2 1
78     1)
79   //G 6
80   hex (1 32 33 17 0 5 21 16) (30 1 50) simpleGrading (1 1 1)
81
82 );
83
84 patches
85 (
86   patch inlet
87   (
88     (0 1 17 16)
89   )
90
91
92   empty frontAndBack
93   (
94     (13 0 11 12) //A
95     (0 5 10 11) //B
```

```
96      (5 6 9 10) //C
97      (6 7 8 9) //D
98      (2 15 32 1) //E
99      (15 3 4 32) //F
100     (0 1 32 5) //G
101
102     (29 16 27 28) //A
103     (16 21 26 27) //B
104     (22 21 26 25)
105     (23 22 25 24)
106     (18 31 33 17) //E
107     (31 19 20 33)
108     (16 17 33 21)
109
110    )
111
112    wall wall
113    (
114      (15 3 19 31) //PD
115      (3 4 20 19) //DE
116      (4 32 33 20) //EQ
117      (32 5 21 33) //QF
118      (5 6 22 21) //FG
119      (6 7 23 22) //GH
120      (7 8 24 23) //HI
121      (8 9 25 24) //IJ
122      (9 10 26 25) //JK
123      (10 11 27 26) //KL
124      (11 12 28 27) //LM
125      (12 13 29 28) //MN
126      (1 2 18 17) //BC
127      (2 15 31 18) //CP
128      (13 0 16 29) //NA
129    )
130
131    patch space
132    (
133    )
134 );
```

Listing A.6: Die Datei 0/U bei abfallendem Druck, Ansatz 2

```
1 FoamFile
2 {
3     version      2.0;
4     format       ascii;
5     class        volVectorField;
6     location     "0";
7     object       U;
8 }
9
10 dimensions [0 1 -1 0 0 0 0];
11
12 internalField uniform ( 0 0 0 );
13
14 boundaryField
15 {
16     inlet
17     {
18         type turbulentInlet;
19         referenceField uniform (0.72 0 0);
20         fluctuationScale ( 0.02 0.01 0.01 );
21
22         value uniform (18 0 0);
23     }
24
25     space
26     {
27         type inletOutlet;
28         inletValue uniform (0 0 0);
29         value uniform (0 0 0);
30     }
31
32     wall
33     {
34         type slip;
35     }
36
37     frontAndBack
38     {
39         type empty;
40     }
41
```

```
42     defaultFaces
43     {
44         type slip;
45     }
46 };
```

Listing A.7: Die Datei 0/p bei abfallendem Druck, Ansatz 2

```
1 FoamFile
2 {
3     version      2.0;
4     format       ascii;
5     class        volScalarField;
6     location     "0";
7     object       p;
8 }
9
10 dimensions [1 -1 -2 0 0 0 0];
11
12 internalField uniform 1e5; // 10000Pa = 1 Bar
13
14 boundaryField
15 {
16     inlet
17     {
18         type zeroGradient;
19     }
20
21     wall
22     {
23         type slip;
24     }
25
26     frontAndBack
27     {
28         type empty;
29     }
30
31     space
32     {
33         type waveTransmissive;
34         field p;
35         phi phi;
36         rho rho;
```

```
37     psi psi;
38     gamma 1.3;
39     fieldInf 1e5;
40     lInf 0.3;
41     value uniform 1e5;
42 }
43
44     defaultFaces
45     {
46         type slip;
47     }
48 }
```

Listing A.8: Die Datei blockMeshDict bei nicht abfallendem Druck, Ansatz 2

```
1 FoamFile
2 {
3     version    2.0;
4     format     ascii;
5     class      dictionary;
6     object     blockMeshDict;
7 }
8
9 convertToMeters 0.001; //units are mm
10
11 vertices
12 (
13     (0 0 1) //0 A
14     (0 -0.5 1) //1 B
15     (0 -28 1) //2 C
16     (200 -28 1) //3 D
17     (200 0 1) //4 E
18
19     (7 1 1) //5 F
20     (34 6 1) //6 G
21     (500 6 1) //7 H
22     (500 106 1) //8 I
23     (34 106 1) //9 J
24
25     (7 106 1) //10 K
26     (0 106 1) //11 L
27     (-100 106 1) //12 M
28     (-100 0 1) //13 N
29     (0 6 1) //14 O
```

---

```

30
31   (7 -28 1) //15 P
32
33 //=====
34
35   (0 0 -1) //0 A
36   (0 -0.5 -1) //1 B
37   (0 -28 -1) //2 C
38   (200 -28 -1) //3 D
39   (200 0 -1) //4 E
40
41   (7 1 -1) //5 F
42   (34 6 -1) //6 G
43   (500 6 -1) //7 H
44   (500 106 -1) //8 I
45   (34 106 -1) //9 J
46
47   (7 106 -1) //10 K
48   (0 106 -1) //11 L
49   (-100 106 -1) //12 M
50   (-100 0 -1) //13 N
51   (0 6 -1) //14 O
52
53   (7 -28 -1) //15 P
54
55   (7 0 1) //32 Q
56   (7 0 -1) //33 Q
57 );
58
59 blocks
60 (
61   //A 0
62   hex (13 0 16 29 12 11 27 28) (50 1 51) simpleGrading (0.1
63     1 250)
64
65   //B 1
66   hex (0 5 21 16 11 10 26 27) (50 1 51) simpleGrading (1 1
67     250)
68
69   //C 2
70   hex (5 6 22 21 10 9 25 26) (60 1 51) simpleGrading (1 1
71     250)
72
73   //D 3

```

---

```

71   hex (6 7 23 22 9 8 24 25) (100 1 51) simpleGrading (10 1
      250)
72
73   //E 4
74   hex (2 15 31 18 1 32 33 17) (50 1 80) simpleGrading (1 1
      1)
75
76   //F 5
77   hex (15 3 19 31 32 4 20 33) (100 1 80) simpleGrading (2 1
      1)
78
79   //G 6
80   hex (1 32 33 17 0 5 21 16) (50 1 50) simpleGrading (1 1 1)
81 );
82
83 patches
84 (
85   patch inlet
86   (
87     (0 1 17 16)
88   )
89
90   empty frontAndBack
91   (
92     (13 0 11 12) //A
93     (0 5 10 11) //B
94     (5 6 9 10) //C
95     (6 7 8 9) //D
96     (2 15 32 1) //E
97     (15 3 4 32) //F
98     (0 1 32 5) //G
99
100    (29 16 27 28) //A
101    (16 21 26 27) //B
102    (22 21 26 25)
103    (23 22 25 24)
104    (18 31 33 17) //E
105    (31 19 20 33)
106    (16 17 33 21)
107
108   )
109
110   wall wall
111   (

```

```

112 (15 3 19 31) //PD
113 (3 4 20 19) //DE
114 (4 32 33 20) //EQ
115 (32 5 21 33) //QF
116 (5 6 22 21) //FG
117 (6 7 23 22) //GH
118
119 (1 2 18 17) //BC
120 (2 15 31 18) //CP
121 (13 0 16 29) //NA
122 )
123
124 patch space
125 (
126 (7 8 24 23) //HI
127 (8 9 25 24) //IJ
128 (9 10 26 25) //JK
129 (10 11 27 26) //KL
130 (11 12 28 27) //LM
131 (12 13 29 28) //MN
132 )
133 );

```

Listing A.9: Die Datei 0/U bei nicht abfallendem Druck, Ansatz 2

```

1 FoamFile
2 {
3     version      2.0;
4     format       ascii;
5     class        volVectorField;
6     location     "0";
7     object       U;
8 }
9
10 dimensions [0 1 -1 0 0 0 0];
11
12 internalField uniform ( 0 0 0 );
13
14 boundaryField
15 {
16     inlet
17     {
18         type    fixedValue;
19         value   uniform (10 0 0);

```

```
20
21     }
22
23     space
24     {
25         type inletOutlet;
26         inletValue uniform (0 0 0);
27         value uniform (0 0 0);
28     }
29
30     wall
31     {
32         type slip;
33     }
34
35     frontAndBack
36     {
37         type empty;
38     }
39
40     defaultFaces
41     {
42         type slip;
43     }
44 };
```

Listing A.10: Die Datei 0/p bei nicht abfallendem Druck, Ansatz 2

```
1 FoamFile
2 {
3     version      2.0;
4     format       ascii;
5     class        volScalarField;
6     location     "0";
7     object       p;
8 }
9
10 dimensions [1 -1 -2 0 0 0 0];
11
12 internalField uniform 1e5; // 10000Pa = 1 Bar
13
14 boundaryField
15 {
16     inlet
```

```
17     {
18         type zeroGradient;
19     }
20
21     wall
22     {
23         type slip;
24     }
25
26     frontAndBack
27     {
28         type empty;
29     }
30
31     space
32     {
33         type waveTransmissive;
34         field p;
35         phi phi;
36         rho rho;
37         psi psi;
38         gamma 1.3;
39         fieldInf 1e5;
40         lInf 0.3;
41         value uniform 1e5;
42     }
43
44     defaultFaces
45     {
46         type slip;
47     }
48 }
```

Listing A.11: Ausgabe von rhoPimpleFoam bei auf  $1e-06$  verringertes DeltaT;

```
1 Time = 7.8e-05
2
3 PIMPLE: iteration 1
4 diagonal: Solving for rho, Initial residual = 0, Final
   residual = 0, No Iterations 0
5 smoothSolver: Solving for e, Initial residual = 0.008471,
   Final residual = 6.58415e-05, No Iterations 3
6 DILUPBiCGStab: Solving for p, Initial residual =
   0.0122408, Final residual = 1.22273e-08, No Iterations 34
```

```

7 DILUPBiCGStab: Solving for p, Initial residual =
  0.00335776, Final residual = 8.80608e-16, No Iterations
  78
8 diagonal: Solving for rho, Initial residual = 0, Final
  residual = 0, No Iterations 0
9 time step continuity errors : sum local = 0.000127199,
  global = -3.15937e-06, cumulative = -3.03029e-06
10 PIMPLE: iteration 2
11 smoothSolver: Solving for e, Initial residual = 0.0378994,
  Final residual = 0.000174214, No Iterations 4
12
13
14 —> FOAM FATAL ERROR:
15 Negative initial temperature T0: -37.4407
16
17     From function Foam::scalar
      Foam::species::thermo<Thermo, Type>::T(Foam::scalar,
      Foam::scalar, Foam::scalar, Foam::scalar
      (Foam::species::thermo<Thermo,
      Type>::*)(Foam::scalar, Foam::scalar) const,
      Foam::scalar (Foam::species::thermo<Thermo,
      Type>::*)(Foam::scalar, Foam::scalar) const,
      Foam::scalar (Foam::species::thermo<Thermo,
      Type>::*)(Foam::scalar) const) const [with Thermo =
      Foam::eConstThermo<Foam::perfectGas<Foam::specie>>;
      Type = Foam::sensibleInternalEnergy; Foam::scalar =
      double; Foam::species::thermo<Thermo, Type> =
      Foam::species::thermo<Foam::eConstThermo<Foam::perfectGas
      <Foam::specie>>, Foam::sensibleInternalEnergy>]
18     in file
      /home/buzz2/pawan/OpenFOAM/1806-docker/OpenFOAM-v1806/src/
      thermophysicalModels/specie/lnInclude/thermoI.H at
      line 54.
19
20 FOAM aborting
21
22 #0 Foam::error::printStack(Foam::Ostream&) at ???
23 #1 Foam::error::abort() at ???
24 #2
      Foam::species::thermo<Foam::eConstThermo<Foam::perfectGas
      <Foam::specie>>,
      Foam::sensibleInternalEnergy>::TEs(double, double,
      double) const at ???
25 #3 Foam::hePsiThermo<Foam::psiThermo,

```

---

```

Foam::pureMixture<Foam::constTransport<Foam::species::thermo
<Foam::eConstThermo<Foam::perfectGas<Foam::specie>>,
Foam::sensibleInternalEnergy>>>
>::calculate(Foam::GeometricField<double,
Foam::fvPatchField, Foam::volMesh> const&,
Foam::GeometricField<double, Foam::fvPatchField,
Foam::volMesh>&, Foam::GeometricField<double,
Foam::fvPatchField, Foam::volMesh>&,
Foam::GeometricField<double, Foam::fvPatchField,
Foam::volMesh>&, Foam::GeometricField<double,
Foam::fvPatchField, Foam::volMesh>&,
Foam::GeometricField<double, Foam::fvPatchField,
Foam::volMesh>&, bool) at ????:?
26 #4 Foam::hePsiThermo<Foam::psiThermo,
Foam::pureMixture<Foam::constTransport<Foam::species::thermo
<Foam::eConstThermo<Foam::perfectGas<Foam::specie>>,
Foam::sensibleInternalEnergy>>> >::correct() at ????:?
27 #5 ? at ????:?
28 #6 __libc_start_main in /lib64/libc.so.6
29 #7 ? at ????:?
30 Aborted (core dumped)

```

# Abbildungsverzeichnis

1.1.	Ausschnitt aus einer der bei der Lichttonorgel verwendeten Scheiben. Bild aus Dangel (2006). . . . .	2
1.2.	Blick auf den Tongenerator einer Hammond B3. Bild aus Zbyszynski (2012). . . . .	3
1.3.	Blick auf den Tongenerator Vox Continental. Bild aus Starling (2009). . . . .	4
2.1.	Vereinfachte schematische Zeichnung des Prinzips der Windlade in der Draufsicht (blau=gedrückte Taste, rot=gezogenes Register). Jedes „o“ steht für eine Pfeife. Die grünen Striche stellen die Spieltraktur dar, die orangenen Striche die Registertraktur. . . . .	11
2.2.	Einige Pfeifen der Rohrflöte aus der St. Paul’s Catholic Church, Danville, Illinois, USA. Quelle: Stauff (2005) . . . . .	14
2.3.	Schematischer Aufbau von Labialpfeifen. Links eine offene, rechts eine gedackte Pfeife. Der blaue Pfeil zeigt die Strömungsrichtung an. Die stehende Welle des Grundtones ist rot gekennzeichnet. . . . .	15
2.4.	Frequenzspektrum einer Prinzipalpfeife auf dem Ton a1. . . . .	17
2.5.	Frequenzspektrum einer Rohrflöte auf dem Ton a1. . . . .	17
2.6.	Vergleich der Frequenzspektren auf dem Ton a1. Prinzipal: blau, Rohrflöte: gelb. Der Prinzipal ist obertonreicher, deutlich sind die blauen Spitzen zu sehen, die bei der Rohrflöte fehlen. . . . .	18
2.7.	Visualisierte Strömung am Pfeifenmund. Bild aus Janke (1997)	19
2.8.	Bildfolge der visualisierten Strömung am Pfeifenmund. Bild aus Janke (1997) . . . . .	19
2.9.	Visualisierte Messwerte der Strömung am Pfeifenmund. Bild aus Janke (1997) . . . . .	20
2.10.	Graphische Darstellung des Tons einer Rohrflöte auf dem Ton a1. . . . .	21
2.11.	Bildfolge des Einströmvorgangs am Pfeifenmund. Bild aus Janke (1997) . . . . .	21
2.12.	Detail vom Prospekt der Orgel der Christuskirche Stuttgart mit Holzpfeifen. Quelle: Praefcke (2014) . . . . .	22

2.13. Darstellung des Prinzips der Puls-Code-Modulation und der Quantisierung . . . . .	28
2.14. Darstellung zweier Signale mit Abtastung. Abbildung aus Hlawitschka (2015) . . . . .	29
2.15. Frequenzspektrum einer Prinzipalpfeife auf dem Ton a1 wenn sie mit 8 bit Auflösung verarbeitet wurde. . . . .	31
2.16. Ansicht der Waveform einer Prinzipalpfeife auf dem Ton a1. . . . .	31
2.17. Vergleich der Wellenform einer Prinzipalpfeife zwischen 8 bit und 16 bit Auflösung. . . . .	32
3.1. Screenshot aus ParaView - Originales Gitter, Ansatz 1. . . . .	38
3.2. Screenshot aus ParaView - Detailansicht der Pfeife, Ansatz 1. . . . .	38
3.3. Screenshot aus ParaView - Detailansicht des Bereichs des Labiums, Ansatz 1. . . . .	39
3.4. Screenshot aus ParaView - Druckverteilung vor und in der Pfeife, Ansatz 1. . . . .	40
3.5. Screenshot aus ParaView - Strömung vor und in der Pfeife, Ansatz 1. . . . .	40
3.6. Screenshot aus ParaView - Strömung vor und in der Pfeife, Ansatz 1. . . . .	41
3.7. Graphik des zweiten Versuches. . . . .	42
3.8. Screenshot aus ParaView - Gitter des zweiten Versuches. . . . .	43
3.9. Druckabfall in der Luft vor der Pfeife . . . . .	43
3.10. Kein Druckabfall in der Luft vor der Pfeife . . . . .	44
3.11. Druck in der Pfeife bei Zeitschritt 0.0, Version 2 . . . . .	45
3.12. Druck in der Pfeife bei Zeit 0,001 ms, Version 2 . . . . .	45
3.13. Druck in der Pfeife bei Zeit 0,002 ms, Version 2 . . . . .	46
3.14. Druck in der Pfeife bei Zeit 0,003 ms, Version 2 . . . . .	46
3.15. Druck in der Pfeife bei Zeit 0,004 ms, Version 2 . . . . .	47
3.16. Druck in der Pfeife bei Zeit 0,005 ms, Version 2 . . . . .	47
3.17. Druck in der Pfeife bei Zeit 0,01 ms, Version 2 . . . . .	48
3.18. Druck in der Pfeife bei Zeit 0,032 ms, Version 2 . . . . .	48
3.19. Druck in der Pfeife bei Zeit 0,041 ms, Version 2 . . . . .	49
3.20. Druck in der Pfeife bei Zeit 0,25 ms, Version 2 . . . . .	49
3.21. Luftströmung in der Pfeife bei Zeit 0,05 ms, Version 2 . . . . .	50
3.22. Luftströmung in der Pfeife bei Zeit 0,07 ms, Version 2 . . . . .	51
3.23. Graphik des dritten Versuches als gedackte Pfeife. . . . .	52
3.24. Graphik des dritten Versuches als offene Pfeife mit zusätzlichem Raum über der Pfeife. . . . .	53
3.25. Temperatur in der Pfeife bei Zeit 0,00007s, Version 2, Zeitschritt $1^{-07}$ s . . . . .	55

3.26. Temperatur in der Pfeife bei Zeit 0,00007s, Version 2, Zeitschritt $1^{-06}$ s . . . . .	55
3.27. Druck in der Pfeife bei Zeit 0,00007s, Version 2, Zeitschritt $1^{-07}$ s	56
3.28. Druck in der Pfeife bei Zeit 0,00007s, Version 2, Zeitschritt $1^{-06}$ s	56
3.29. Temperatur in der Pfeife bei Zeit 0,000077s, Version 2, Zeitschritt $1^{-06}$ s . . . . .	57
3.30. Temperatur in der Pfeife bei Zeit 0,000077s, Version 2, Zeitschritt $1^{-06}$ s, vergrößert und mit größerer Farbskala . . . . .	57
3.31. Vergleich der erzeugten Schwingungen in den 3 verschiedenen Versionen der gedackten Pfeife, Screenshot aus Audacity. Oben die unveränderte, gedackte Pfeife, darunter die mit geringem Winddruck angeblasene Pfeife, ganz unten die schmale gedackte Pfeife. . . . .	58
3.32. Visualisierung des Drucks bei 0,01696 s . . . . .	59
3.33. Visualisierung des Drucks bei 0,01702s . . . . .	59
3.34. Visualisierung des Drucks bei 0,01708 s . . . . .	59
3.35. Visualisierung der Strömung bei 0,01696 s . . . . .	60
3.36. Visualisierung der Strömung bei 0,01702s . . . . .	60
3.37. Visualisierung der Strömung bei 0,01708 s . . . . .	60
3.38. Visualisierung des Drucks bei 0,01411s . . . . .	61
3.39. Visualisierung des Drucks bei 0,01499 s . . . . .	61
3.40. Visualisierung der Strömung bei 0,01411s . . . . .	62
3.41. Visualisierung der Strömung bei 0,01499s . . . . .	62
3.42. Frequenzanalyse der gedackten Pfeife. . . . .	63
3.43. Die erzeugte Schwingung der unveränderten, gedackten Pfeife.	63
3.44. Frequenzanalyse der mit geringerem Winddruck angeblasenen gedackten Pfeife. . . . .	64
3.45. Die erzeugte Schwingung der mit geringem Winddruck angeblasene gedackten Pfeife. . . . .	64
3.46. Frequenzanalyse der schmalen, gedackten Pfeife. . . . .	65
3.47. Die erzeugte Schwingung der schmalen, gedackten Pfeife. . . . .	65
3.48. Vergleich der erzeugten Schwingungen in den 3 verschiedenen Versionen der gedackten Pfeife. Schwarz die unveränderte, gedackte Pfeife, blau die mit geringen Winddruck angeblasene Pfeife, rot die schmale gedackte Pfeife. . . . .	66
3.49. Vergleich der erzeugten Schwingungen in den 3 verschiedenen Versionen der offenen Pfeife, Screenshot aus Audacity. Oben die unveränderte, offene Pfeife, darunter die mit geringen Winddruck angeblasene Pfeife, ganz unten die schmale offene Pfeife.	67
3.50. Visualisierung der Strömung bei 0,002 s . . . . .	68
3.51. Visualisierung der Strömung bei 0,01106 s . . . . .	68
3.52. Visualisierung des Drucks bei 0,01106 s . . . . .	69

3.53. Visualisierung des Drucks bei 0,01401s mit bewusst eingeschränktem Wertebereich der Skala, schmale Pfeife . . . . .	69
3.54. Visualisierung des Drucks bei 0,01384s mit bewusst eingeschränktem Wertebereich der Skala, schmale Pfeife . . . . .	70
3.55. Visualisierung des Drucks bei 0,01388s mit bewusst eingeschränktem Wertebereich der Skala, schmale Pfeife . . . . .	70
3.56. Frequenzanalyse der offenen Pfeife. . . . .	71
3.57. Die erzeugte Schwingung der unveränderten, offenen Pfeife. . .	71
3.58. Frequenzanalyse der mit geringerem Winddruck angeblasenen offenen Pfeife. . . . .	72
3.59. Die erzeugte Schwingung der mit geringerem Winddruck angeblasenen offenen Pfeife. . . . .	72
3.60. Visualisierung des Drucks bei 0,00987 s, offene Pfeife mit geringem Druck angeblasen . . . . .	73
3.61. Visualisierung des Drucks bei 0,00993 s, offene Pfeife mit geringem Druck angeblasen . . . . .	73
3.62. Frequenzanalyse der schmaleren, offenen Pfeife. . . . .	74
3.63. Die erzeugte Schwingung der schmalen, offenen Pfeife. . . . .	74
3.64. Vergleich der erzeugten Schwingungen in den 3 verschiedenen Versionen der offenen Pfeife. Schwarz die unveränderte, gedackte Pfeife, blau die mit geringen Winddruck angeblasene Pfeife, rot die schmale offene Pfeife. . . . .	75
3.65. Visualisierung des Drucks bei 0,01255 s . . . . .	76
3.66. Visualisierung des Drucks bei 0,01288 s, schmale, offene Pfeife	76
3.67. Visualisierung des Drucks bei 0,01843 s, offene Pfeife mit geringem Druck angeblasen . . . . .	77
3.68. Visualisierung und Vergleich der Strömung bei 0,00041 s . . . .	78
3.69. Visualisierung und Vergleich der Strömung bei 0,00164 s . . . .	78
3.70. Visualisierung und Vergleich der Strömung bei 0,00240 s . . . .	79
3.71. Visualisierte Messwerte der Strömung am Pfeifenmund. Bild aus Janke (1997), zur besseren Vergleichbarkeit gespiegelt und wie Abbildung 3.72 ausgerichtet . . . . .	80
3.72. Visualisierung der Strömung bei 0,02 s . . . . .	81
3.73. Spektrum einer gedackten Orgelpfeife auf dem Ton gis <sup>2</sup> . . . .	82
3.74. Vergleich der Frequenzspektren der simulierten (hellblau) und einer realen Pfeife (sattrot). . . . .	83
3.75. Visualisierung der Wellen der Simulation (oben) und einer realen, gedackten Orgelpfeife (unten) . . . . .	84

# Verzeichnis der Quelltexte

A.1. Die Quelltextdatei rhoPimpleFoam.C aus dem Quelltext von OpenFOAM v18.12 in welcher der Solver rhoPimpleFoam definiert ist. . . . .	88
A.2. Die Quelltextdatei pimpleControl.C aus dem Quelltext von OpenFOAM v18.12 in welcher der PIMPLE-Algorithmus implementiert ist. . . . .	93
A.3. Die Datei thermophysicalProperties wie sie in den verschiedenen Ansätzen verwendet wurde. . . . .	100
A.4. Die Datei 0/T wie sie in den verschiedenen Ansätzen verwendet wurde. . . . .	101
A.5. Die Datei blockMeshDict bei abfallendem Druck, Ansatz 2 . . .	102
A.6. Die Datei 0/U bei abfallendem Druck, Ansatz 2 . . . . .	106
A.7. Die Datei 0/p bei abfallendem Druck, Ansatz 2 . . . . .	107
A.8. Die Datei blockMeshDict bei nicht abfallendem Druck, Ansatz 2	108
A.9. Die Datei 0/U bei nicht abfallendem Druck, Ansatz 2 . . . . .	111
A.10. Die Datei 0/p bei nicht abfallendem Druck, Ansatz 2 . . . . .	112
A.11. Ausgabe von rhoPimpleFoam bei auf 1e-06 verringertes DeltaT;	113

# Literaturverzeichnis

- [Adriaensen 2004] ADRIAENSEN, Fons: *Aeolus - A church organ in your PC*. April 2004. – URL <https://kokkinizita.linuxaudio.org/papers/aeolus-pres.pdf>
- [Beijer 2010] BEIJER, Hans: *Handleiding 20 toets draaiorgel (Tragbare 20er Drehorgel)*. 2010. – URL [www.draaiorgelweb.nl/software/beijer20compleet.pdf](http://www.draaiorgelweb.nl/software/beijer20compleet.pdf). – Zugriffsdatum: 02.01.2019
- [Bergweiler 2005] BERGWEILER, Steffen: *Körperszillation und Schallabstrahlung akustischer Wellenleiter unter Berücksichtigung von Wandungseinflüssen und Kopplungseffekten*, Universität Potsdam, Dissertation, 2005
- [Böhm 1986] BÖHM, Rainer: *Elektronische Orgeln und ihr Selbstbau*. Franzis Verlag GmbH, Januar 1986
- [Brackhane 2018] BRACKHANE, Fabian: *Orgel - was ist das? Das Pfeifenwerk*. 2018. – URL [http://www.orgelauskunft.de/orgel\\_erklaert\\_3.html](http://www.orgelauskunft.de/orgel_erklaert_3.html). – Zugriffsdatum: 25.01.2019
- [Brünner 2002] BRÜNNER, Arndt: *Mensurenrechner für Orgelpfeifen*. 2002. – URL <http://www.arndt-bruenner.de/hausorgel/CalcMens.htm>. – Zugriffsdatum: 16.04.2018
- [Combo-Organ 2019] COMBO-ORGAN: *Vox: Single Manual Continentals*. 2019. – URL <http://www.combo-organ.com/Vox/>. – Zugriffsdatum: 25.01.2019
- [Dangel 2006] DANGEL, Gerhard: *Lichttonorgelversuchsscheibe.jpg*. März 2006. – URL <https://commons.wikimedia.org/wiki/File:Lichttonorgelversuchsscheibe.jpg>. – Zugriffsdatum: 25.01.2019
- [Dangel 2007] DANGEL, Gerhard: *Lichttonorgel*. Juli 2007. – URL <https://de.wikipedia.org/wiki/Lichttonorgel>. – Zugriffsdatum: 24.01.2019
- [Duden 2017] DUDEN: *Duden - Deutsche Rechtschreibung*. Bibliographisches Institut GmbH, 2017

- [Feist 2008] FEIST, Rico: *Besondere Lernleistung: Analyse synthetischer und samplingbasierter Simulationen von Pfeifenorgeln*, Gymnasium Engelsdorf, Besondere Lernleistung, 2008
- [Hagenow 2001] HAGENOW, Henri: *Digitale Synthese komplexer Wellenformen zur Simulation akustischer, elektrischer und optischer Eigenzustände mehrdimensionaler Systeme*, Technische Universität Berlin, Diplomarbeit, 2001
- [Hammond 1955] HAMMOND: *Organ Manual B3/C3*. Hammond Company, 1955
- [Hau 2003] HAU, Rita: *Schülerwörterbuch Latein-Deutsch/Deutsch-Latein*. Ernst Klett Sprachen GmbH, Stuttgart, 2003. – ISBN 3-12-517523-2
- [Hlawitschka 2015] HLAWITSCHKA, Prof. M.: *Digitale Signalverarbeitung - Folien*. 2015
- [Janke 1997] JANKE, Reiner: *Strömungsforschung an klingenden Pfeifen*. 1997. – URL <http://www.orgel-info.de/emden.htm>. – Zugriffsdatum: 26.01.2019
- [Johannus 2018] JOHANNUS: *Johannus Live*. Johannus Orgelbau, 2018
- [Kleine 2019] KLEINE, Matthias P.: *Was ist GrandOrgue?* 2019. – URL <https://www.musikundtheologie.de/grandorgue.html>. – Zugriffsdatum: 20.01.2019
- [Klotz 1998] KLOTZ, Hans: *Das Buch von der Orgel: Über Wesen und Aufbau des Orgelwerkes, Orgelpflege und Orgelspiel*. Bärenreiter Verlag, 1998
- [Kraus 2012] KRAUS, Clemens: *Bau einer HX3 Hammond Orgel*. Juni 2012. – URL <http://www.clemens-kraus.de/technik/hx3-hammond-orgel.html>. – Zugriffsdatum: 27.01.2019
- [Lecheler 2008] LECHELER, Stefan: *Numerische Strömungsberechnung*. Vieweg+Teubner, Oktober 2008
- [Lesch 2008] LESCH, Harald: *Die kürzeste Geschichte allen Lebens*. Piper-Verlag, München, 2008
- [OpenCFD 2018] OPENCDFD: *OpenFOAM user guide - standard solvers*. 2018. – URL <https://www.openfoam.com/documentation/user-guide/>. – Zugriffsdatum: 24.01.2019

- [Praefcke 2014] PRAEFCKE, Andreas: *Stuttgart Christuskirche 05*. 2014. – URL [https://commons.wikimedia.org/wiki/File:Stuttgart\\_Christuskirche\\_05.jpg](https://commons.wikimedia.org/wiki/File:Stuttgart_Christuskirche_05.jpg). – Zugriffsdatum: 16.04.2018
- [Soundfont 1996] SOUNDFONT: *SoundFont 2.01 Technical Specification*. E-mu Systems, 1996
- [Starling 2009] STARLING, Jon: *Voxcontinentalinternals.jpg*. July 2009. – URL <https://en.wikipedia.org/w/index.php?title=File:Voxcontinentalinternals.jpg>. – Zugriffsdatum: 25.01.2019
- [Stauff 2005] STAUFF, Edward L.: *Encyclopedia of Organ Stops*. 2005. – URL <http://organstops.org>. – Zugriffsdatum: 5.1.2008
- [Versteeg 2007] VERSTEEG, H K.: *An Introduction to COMPUTATIONAL FLUID DYNAMICS The Finite Volume Method*. Pearson, 2007
- [Viscount 2019] VISCOUNT: *Technology*. Januar 2019. – URL <https://www.viscountinstruments.com/technology/>
- [Wand 2018] WAND, Benjamin und Jannik B.: *Open Source Orgelbau*. 2018. – URL [https://media.ccc.de/v/35c3-9768-open\\_source\\_orgelbau](https://media.ccc.de/v/35c3-9768-open_source_orgelbau). – Zugriffsdatum: 02.01.2019
- [Wild 2017] WILD, Michael: *FreeFOAM Home Page*. 2017. – URL <http://freefoam.sourceforge.net/>. – Zugriffsdatum: 23.01.2019
- [Zbyszynski 2012] ZBYSZYNSKI, Henry: *Color Coded Wires, Hammond B3*. August 2012. – URL <https://www.flickr.com/photos/7891030@N08/7852824542>. – Zugriffsdatum: 25.01.2019
- [Zscherpel 2001] ZSCHERPEL, Detlef: *Abschlussbericht zum Projekt „Verfahren im rekonstruierenden und restaurativen Orgelbau“*. Mai 2001
- [Zurek 2013] ZUREK, Jiri: *Do not load in 16-bit!* 2013. – URL <http://www.sonusparadisi.cz/en/blog/do-not-load-in-16-bit/>. – Zugriffsdatum: 20.09.2018

# Selbstständigkeitserklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann

Leipzig, den 30. Januar 2019

Rico Feist