

**Echtzeitfähige digitale Zwillinge auf Basis Physik-informierter
neuronaler Netze für die Strömungsvorhersage in und um Stallungen**

Von der Fakultät für Ingenieurwissenschaften, Abteilung
Maschinenbau und Verfahrenstechnik

der

Universität Duisburg-Essen

zur Erlangung des akademischen Grades

eines

Doktors der Ingenieurwissenschaften

Dr.-Ing.

genehmigte Dissertation

von

Fabian Pioch

aus

Wesel

Gutachter: Prof. Dr.-Ing. Dr. h.c. Dieter Schramm
Prof. Dr.-Ing. Peter Kerstiens
Prof. Dr.-Ing. Bettar Ould el Moctar

Tag der mündlichen Prüfung: 07.12.2023

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit am Mechatronik-Institut Bocholt der Westfälischen Hochschule. Mein besonderer Dank gilt Herrn Prof. Dr.-Ing. Dr. h.c. Dieter Schramm, meinem Erstgutachter, der mir bereits in unserem ersten Gespräch Vertrauen schenkte und meine Forschungsideen fortan unterstützte. Herrn Prof. Dr.-Ing. Peter Kerstiens, meinem Zweitgutachter, danke ich besonders für seine wertvolle und strategische Unterstützung und die einzigartige Gelegenheit, eine Vorlesungsreihe zu gestalten und zu halten. Mein Dank gilt auch Herrn Prof. Dr.-Ing. Bettar Ould el Moctar, meinem Drittgutachter, für seine Ratschläge zu Veröffentlichungen und die Erstellung des Gutachtens.

Ein herzlicher Dank gebührt meinen Kollegen am Institut, insbesondere Herrn Jan Hauke Harmening, M. Eng., der mir fachlich und persönlich stets zur Seite stand. Die gemeinsamen Erfahrungen in der Institutsarbeit und auf Konferenzen werden mir in Erinnerung bleiben. Besonderer Dank geht an Herrn Dipl.-Ing. (FH) André Schwack, M. Eng. für die herzliche Aufnahme in das gemeinsame Büro und die anregenden Diskussionen zu vielfältigen Themen. Außerdem danke ich den Studierenden, deren Ergebnisse der Bachelor- und Masterarbeiten zu dieser Arbeit beigetragen haben, allen voran Herrn Lucas Schmeing, M. Eng. für seinen herausragenden Einsatz und Beitrag im Rahmen seiner Masterarbeit.

Mein tiefster Dank gilt meiner Familie, insbesondere meinen liebevollen Eltern, die mich stets unterstützt und ermutigt haben. Ihre bedingungslose Liebe und Hingabe waren mein Rückhalt. Mein größter Dank geht an meine Freundin Esther für ihre unglaubliche Geduld und Ermutigung während aller Phasen der Promotion. Ohne meine Familie und meine Freundin wäre mein Weg nicht so bunt und erfüllt.

Kurzfassung

In dieser Arbeit wird eine Methode entwickelt, die echtzeitfähige digitale Zwillinge zur Strömungsvorhersage in und um frei belüftete Kuhställe ermöglicht. Diese digitalen Zwillinge erfassen den realen Systemzustand und ermöglichen es, digitale Analysen und Visualisierungen in Echtzeit durchzuführen. Dadurch können Anomalien frühzeitig identifiziert und der Systemzustand optimal eingestellt werden. Grundlage der Methode sind Daten aus drei saisonalen Messkampagnen in einem Versuchsstall und ergänzenden externen Wetterdaten, die Windgeschwindigkeit, -richtung, Lufttemperatur, -feuchte und Methankonzentration umfassen. Die umweltbezogenen Parameter können am realen Versuchsstall nicht systematisch variiert werden, weshalb ein numerisches Simulationsmodell entwickelt wird. Dieses Modell dient als Grundlage, um darauf statistische Versuchsplanungen anwenden zu können. Ein Plackett-Burman-Versuchsplan ermittelt signifikante Faktoren bezüglich der Zielgrößen Tierwohl, Emissionen und Durchlüftung. Ein Box-Behnken-Versuchsplan analysiert diese hinsichtlich Wechselwirkungen und quadratischen Effekten auf die Zielgrößen. Für eine Visualisierung des Strömungsfeldes in Echtzeit und bei variabler Einstellung ausgewählter Parameter werden Physik-informierte neuronale Netze eingesetzt. Physik-informierte neuronale Netze stellen einen dateneffizienten Ansatz im maschinellen Lernen dar. Der klassische Verlustterm wird um die Residuen der zugrundeliegenden physikalischen Gleichungen ergänzt, indem die approximierten Gradienten aus der automatischen Differentiation während des Trainings entnommen und in die Gleichung eingesetzt werden. Diese Arbeit verwendet die Reynolds-gemittelten Navier-Stokes-Gleichungen als physikalischen Verlustterm und stellt einen neuartigen Turbulenzmodellierungsansatz für Physik-informierte neuronale Netze vor. Für das Training werden 3 % der verfügbaren Daten verwendet. Bei der Validierung einer nicht trainierten Parameterkombination zeigt die Methode eine relative Abweichung von unter 15 % in den Bereichen hoher Gradienten. Die Vorhersage der Geschwindigkeitskomponenten und des Drucks für 10.000 Koordinatenpunkte erfolgt innerhalb von 6,5 Millisekunden und erfüllt somit die Anforderungen an die harte Echtzeit. Die Entwicklung einer geeigneten Kommunikationsarchitektur ermöglicht die Integration von Physik-informierten neuronalen Netzen und Sensordaten in den digitalen Zwilling frei belüfteter Kuhställe. Durch einen kryptografisch abgesicherten Zugang können Landwirtschaftsbetriebe und Dritte die Daten, Modelle und Funktionen aktualisieren, um die Luftführung im Stall in Echtzeit zu analysieren und somit das Tierwohl zu optimieren sowie die Emissionen zu reduzieren.

Abstract

Within this work, a method is developed to create real-time capable digital twins for predicting airflow in and around naturally ventilated barns. These digital twins capture the real system state and allow digital analysis and visualization in real-time. This enables early detection of anomalies and optimal adjustment of the system state. The method is based on data from three seasonal measurement campaigns in a real experimental barn and additional external weather data, including wind speed, wind direction, air temperature, humidity, and methane concentration. Since the environmental parameters cannot be systematically varied at the naturally ventilated barn, a numerical simulation model is developed. This model serves as the basis for the application of design of experiments. A Plackett-Burman design identifies significant factors with respect to the target variables animal welfare, emissions, and ventilation. A Box-Behnken design analyzes these factors for interactions and quadratic effects on the target variables. For a real-time visualization of the airflow field with variable settings of selected parameters, physics-informed neural networks are employed. Physics-informed neural networks represent a data-efficient approach in machine learning. The classical loss term is supplemented with the residuals of the governing equations by inserting the approximated gradients from the automatic differentiation during training into the governing equation. This work uses the Reynolds-averaged Navier-Stokes equations as the physical loss term and presents a novel turbulence modeling approach for physics-informed neural networks. Training uses 3% of the available data. When validating an untrained parameter combination, the method shows a relative deviation of less than 15% in areas of high gradients. Prediction of velocity components and pressure for 10,000 coordinate points is completed within 6.5 milliseconds, meeting hard real-time requirements. The development of a suitable communication architecture enables the integration of physics-informed neural networks and sensor data into the digital twin of naturally ventilated cow barns. Through cryptographically secured access, agricultural businesses and third parties can update the data, models, and functions to analyze barn airflow in real time to optimize animal welfare and reduce emissions.

Inhaltsverzeichnis

Kurzfassung	v
Abstract	vii
Notation	xiii
1 Einführung	1
1.1 Motivation	1
1.2 Problembeschreibung	3
1.3 Aufbau der Arbeit	5
2 Grundlagen und Stand der Forschung	7
2.1 Grundlagen des digitalen Zwillings	7
2.2 Grundlagen der numerischen Strömungssimulation	10
2.2.1 Grundgleichungen der Strömungsmechanik	11
2.2.2 Finite-Elemente-Methode und Finite-Volumen-Methode	14
2.2.3 Reynolds-gemittelte Gleichungen und Turbulenzmodellierung	16
2.2.4 Verifizierung und Validierung numerischer Modelle	19
2.3 Grundlagen der statistischen Versuchsplanung	23
2.4 Grundlagen (Physik-informierter) künstlicher neuronaler Netze	26
2.4.1 Vom Neuron zum neuronalen Netz	26
2.4.2 Berechnung, Fehlerbestimmung und Optimierung	27
2.4.3 Physik-informierte neuronale Netze	30
2.5 Stand der Forschung	32
2.5.1 Strömungssimulation landwirtschaftlicher Gebäude	32
2.5.2 Physik-informierte neuronale Netze und Strömungsmechanik	35
2.5.3 Digitale Zwillinge in der Landwirtschaft	37
2.6 Wissenschaftlicher Beitrag der Arbeit	38

3	Experimentelle Ergebnisse	41
3.1	Versuchsstall	41
3.2	Messinstrumente	43
3.2.1	Strömungssensoren	43
3.2.2	Methansensoren	45
3.2.3	LUBRATEC SMARTBOX	46
3.2.4	Wetterstation – Deutscher Wetterdienst	47
3.2.5	Wetterstation am Versuchsstall	48
3.3	Messaufbau	48
3.4	Messkampagnen und Auswertung	50
3.4.1	Lüftungselemente	50
3.4.2	Strömungsgeschwindigkeiten	51
3.4.3	Lufttemperatur	53
3.4.4	Luftfeuchte	54
3.4.5	Methangehalt	54
4	Numerische Ergebnisse	57
4.1	Parametrisierbares CAD-Modell	58
4.1.1	Geometrische Parameter	58
4.1.2	Modellierungsstrategie	59
4.1.3	CAD-Geometrien	60
4.2	Diskretisierung	66
4.3	Randbedingungen	68
4.4	Einstellungen des Lösers	71
4.5	Verifizierung und Validierung	72
4.5.1	Verifizierung	72
4.5.2	Validierung	75
4.6	Statistische Versuchsplanung	82
4.6.1	Zielgrößen der statistischen Versuchsplanung	83
4.6.2	<i>Screening</i> -Versuchsplan und Auswertung	84
4.6.3	Antwortflächen-Versuchsplan und Auswertung	88
5	Methodenentwicklung für echtzeitfähige Metamodelle von Strömungen	95
5.1	Zusammengesetzte Verlustfunktion	96
5.2	Trainingsalgorithmus	97
5.3	Turbulenzmodellierung für Physik-informierte neuronale Netze	98
5.3.1	Physik-informierte neuronale Netze und das Mischlängenmodell	100
5.3.2	Implementierung des k - ω -Modells für Physik-informierte neuronale Netze	100
5.3.3	Turbulenzmodellierung mit dem ν_t -Modell	101
5.3.4	Turbulenzmodellierung via pseudo-Reynolds-Spannungen	102
5.3.5	Auswertung der Turbulenzmodellierung	104
5.4	Variable Eingangsgrößen	104
5.5	Skalierung der Navier-Stokes-Gleichungen	105
5.6	Sampling-Methode	106
5.7	Definition der Echtzeit für die Strömungsvorhersage in und um Stallungen	108
5.8	Anwendung, Verifizierung und Validierung der Methode	109

6	Entwicklung des digitalen Zwillings für Stallungen	119
6.1	Kommunikationsteilnehmer und -protokolle	119
6.1.1	Kommunikationsteilnehmer	120
6.1.2	Analyse zum Kommunikationsprotokoll	121
6.2	Datenbanksystem	124
6.3	IT-Sicherheitskonzept und Kontenverwaltung	124
6.4	Kommunikationsarchitektur und Aufbau der Webapplikation	126
6.4.1	Kommunikationsschema	126
6.4.2	Webapplikation	127
6.4.3	Programmierschnittstelle	131
7	Diskussion	135
7.1	Analyse zum wissenschaftlichen Beitrag der Arbeit	135
7.2	Limitationen und Lösungsansätze	139
7.2.1	Experimentelle Ergebnisse	139
7.2.2	Numerische Ergebnisse	140
7.2.3	Echtzeitfähige Strömungsvorhersage	141
7.2.4	Digitaler Zwilling	142
8	Fazit	143
8.1	Zusammenfassung	143
8.2	Zukünftige Arbeiten	147
A	Ergänzende Auswertung der Ultraschallanemometer	149
B	Sitzungsprotokoll	151
C	Bilder des Versuchstalls	153
D	Konturdiagramme der numerischen Simulationen	155
D.1	Konturdiagramme der Durchläufe des <i>Screening</i> -Versuchsplans	156
D.2	Konturdiagramme des Box-Behnken-Versuchsplans	175
E	Programmcode	189
E.1	IPCC Berechnung des Methans	189
E.2	Programmcode PINN	191
E.3	Programmcode Webapplikation	201
E.4	Beispiele zur Programmierschnittstelle	211
E.5	Schnittstellenübersetzer von OPC UA zu REST	213
	Abbildungsverzeichnis	215
	Tabellenverzeichnis	221
	Literatur	223

Notation

Verwendete Symbole

Formelzeichen	Bedeutung
A	Vektorfeld
<i>a</i>	Aktivitätslevel eines Neurons
a	Aktivitäts-Vektor eines Neurons
<i>b</i>	Bias eines Neurons
b	Bias-Vektor
b_G	Gebäudebreite
<i>C</i>	Konstanten
<i>c</i>	Konzentration
\bar{c}	Zeitlich gemittelte Konzentration
c_r	Rauheitsbeiwert
D	Deformationsrate
<i>D</i>	Differentialanteil
<i>d</i>	Wandabstand
D_{CH_4}	Diffusionskoeffizient für Methan
<i>e</i>	Innere Energie
e_d	Dampfdruck
\hat{e}	Einheitsvektor
E_p	Effektstärke
e_s	Sättigungsdampfdruck
f	Kraftvektor
F_S	Sicherheitsfaktor
<i>G</i>	Wachstumsrate der Elemente

Formelzeichen	Bedeutung
g	Gravitationskonstante
h	Spezifische Enthalpie
h_D	Montagehöhe Deckenventilator
h_E	Elementkantenlänge
h_G	Gebäudehöhe
h_T	Traufhöhe
\mathbf{I}	Einheitstensor
I	Integralanteil
I_v	Turbulenzintensität
k	Turbulente kinetische Energie
k_s	Sandrauheit
\mathcal{L}	(Zusammengesetzte) Verlustfunktion
l	Allgemeine Längenangabe
l_G	Gebäudelänge
l_m	Mischungsweg
l_{NG}	Distanz zum Nachbargebäude
l_S	Schlauchlänge
M	Molmasse
$m_i^{(j)}$	Momentum der Ordnung j und Iteration i
\dot{m}	Massestrom
$\hat{m}^{(j)}$	Korrekturterm zum Momentum der Ordnung j
\mathbf{n}	Orthogonal auf der Oberfläche stehender Vektor
N	Anzahl
o	Systemausgang
\mathbf{o}_i	Optimierungsvektor der Iteration i
P	Proportionalanteil
p	Druck
p_{num}	Konvergenzrate der numerischen Lösungen
p_{ref}	Referenzdruck
Q	Quartil
\mathbf{q}	Vektor mit Quelltermen
q	Quellterm
q_E	Elementqualität
\dot{Q}	Wärmeleistung
R	Elektrischer Widerstand
r	Netzverfeinerungsrate
Re	Reynolds-Zahl
R_G	Gaskonstante
R_V	Luftwechselrate

Formelzeichen	Bedeutung
S	Oberfläche
s	Vorzeichenfunktion
Sc	Schmidt-Zahl
\mathbf{T}	Spannungstensor
T	Temperatur
t	Zeit
T_{virt}	Virtuelle Temperatur
t_W	Wanddicke
\bar{T}	Zeitlich gemittelte Temperatur
U	Geschwindigkeitsbetrag
\mathbf{u}	Geschwindigkeitsvektor
u	Geschwindigkeitskomponente in x -Achsenrichtung
U_{num}	Unsicherheit gemäß einer Gauß-Verteilung
u_{num}	Unsicherheit ohne Annahme einer Gauß-Verteilung
u_{ref}	Referenzwert
u'	Fluktuierende Geschwindigkeitskomponente in x
\bar{U}	Zeitlich gemittelter Geschwindigkeitsbetrag
\bar{u}	Zeitlich gemittelte Geschwindigkeitskomponente
\tilde{u}	Approximierter Ausgabewert
V	Volumen
v	Geschwindigkeitskomponente in y -Achsenrichtung
V_A	Axialventilatoren
v_b	Basisgeschwindigkeit
V_D	Deckenventilator
v'	Fluktuierende Geschwindigkeitskomponente in y
V_S	Schlauchbelüftung
$v_m(z)$	Mittlere Geschwindigkeit in Abhängigkeit der Höhe z
W	Wickellüftung
w	Geschwindigkeitskomponente in z -Achsenrichtung
W_o	Obere Wickellüftung
\mathbf{W}^T	Transponierte Gewichtungsmatrix
W_u	Untere Wickellüftung
$w_j^{(i)}$	Gewicht des Neurons j der Schicht i
x	Kartesische Koordinate
X_{CH_4}	Methankonzentration in ppm
x_i	Systemeingang
y	Kartesische Koordinate
y^+	Dimensionsloser Wandabstand
z	Kartesische Koordinate

Formelzeichen	Bedeutung
z_a	Aktivierungsfunktion
z_0	Rauheitslänge
α	Windrichtung
$\bar{\alpha}$	Zeitlich gemittelte Windrichtung
α_W	Profilexponent
β	Dachneigungswinkel
Γ	Diffusivität
γ	p-Wert
δ	Kronecker-Delta
ε	Dissipationsrate
ϵ	Fehler
η	Lernrate
Θ	Parameter einer Funktion
κ	Von Kármán Konstante
Λ	Flussterm
λ	Gewichtungskoeffizient
λ_Q	Wärmeleitkoeffizient
μ	Dynamische Viskosität
ν	Kinematische Viskosität
ν_t	Turbulente Viskosität
ρ	Dichte
σ	Standardabweichung
τ_{ij}	Reynolds-Spannungen
Φ	Beliebige Zustandsgröße
ϕ	Intensive Zustandsgröße
φ	Luftfeuchte
φ_t	Testfunktion
Φ'	Fluktuierende beliebige Zustandsgröße
$\bar{\Phi}$	Zeitlich gemittelte beliebige Zustandsgröße
Ψ	Basisfunktionen
Ω	Berechnungsgebiet
ω	Spezifische Dissipationsrate
$\partial\Omega$	Berandung des Berechnungsgebiets

Verwendete Akronyme

Akronym	Bedeutung
API	<i>Application Programming Interface</i>
CAD	<i>Computer-aided design</i>
CFD	<i>Computational Fluid Dynamics</i>
CFL	Courant-Friedrichs-Lewy
DoE	<i>Design of experiments</i>
DNS	Direkte numerische Simulation
DWD	Deutscher Wetterdienst
FAC2	<i>Factor of two</i>
FB	<i>Fractional bias</i>
FDM	Finite-Differenzen-Methode
FEM	Finite-Elemente-Methode
FVM	Finite-Volumen-Methode
GCI	<i>Grid Convergence Index</i>
GMRES	<i>Generalized minimal residual method</i>
HTML	<i>Hypertext Markup Language</i>
IoT	<i>Internet of Things</i>
IPCC	<i>Intergovernmental Panel on Climate Change</i>
IQR	<i>Interquartile range</i>
JSON	<i>JavaScript Object Notation</i>
JWT	<i>JavaScript Object Notation Webtoken</i>
KV	Kontrollvolumen
LoRaWAN	<i>Long Range Wide Area Network</i>
L-BFGS	<i>Limited-memory Broyden-Fletcher-Goldfarb-Shanno</i>
MG	<i>Geometric mean bias</i>
MOX	Metalloxid-Halbleiter
MSE	<i>Mean squared error</i>
NASA	<i>National Aeronautics and Space Administration</i>
NMSE	<i>Normalized mean square error</i>
OPC UA	<i>Open Platform Communication Unified Architecture</i>
PDE	<i>Partial differential equation</i>
PINN	Physik-informiertes neuronales Netz
ppm	<i>Parts per million</i>
RANS	<i>Reynolds-Averaged Navier-Stokes</i>
RB	Randbedingung
REST	<i>Representational State Transfer</i>
SQL	<i>Structured Query Language</i>
SSL	<i>Secure Sockets Layer</i>

Akronym	Bedeutung
tanh	<i>Tangens hyperbolicus</i>
THI	<i>Temperature Humidity Index</i>
TLS	<i>Transport Layer Security</i>
USA	Ultraschallanemometer
VG	<i>Geometric variance</i>
XML	<i>Extensible Markup Language</i>

Einführung

1.1 Motivation

„That’s one small step for <a> man... one... giant leap for mankind!“
Neil Armstrong, 21. Juli 1969, 02:56:20 Uhr (UTC)

Die Verknüpfung von Elementen aus der realen und virtuellen Welt sowie ihre gemeinsame Erprobung erwies sich als entscheidendes Werkzeug, das die Mondlandung ermöglichte und letztendlich zu dem berühmten Zitat führte. Jahrzehnte später wurde dieses Konzept der bidirektionalen Kommunikation in der Roadmap der amerikanischen *National Aeronautics and Space Administration* (NASA) erstmals als „digitaler Zwilling“ bezeichnet (SHAFTO et al., 2010; SHAFTO et al., 2012).

In den 1960ern begann der Wettlauf ins All und damit die Geschichte des digitalen Zwillings: Als Lehre aus der Explosion des Sauerstofftanks und anschließender Beschädigung des Triebwerks während der *Apollo 13* Mission setzte die NASA Simulatoren ein, um das Versagen zu bewerten und ergänzte dafür das physische Modell des Raumschiffs um digitale Komponenten (GLAESSGEN und STARGEL, 2012; ALLEN, 2021). Dieser erste digitale Zwilling erlaubte forensische Analysen der Vorgänge, die zu dem Ereignis und damit dem Abbruch der Mission geführt hatten. Darüber hinaus konnten Erkenntnisse gewonnen werden, um Programme, Prozesse und Komponenten zu optimieren. Die erste erfolgreiche Umsetzung eines digitalen Zwillings ist nach heutigen Standards rudimentär. Es erforderte einen hohen Aufwand und proprietäre Schnittstellen, um das physische Objekt mit der virtuellen Welt

zu koppeln. Seitdem haben sich die Konzepte und Umsetzungen zunehmend verbessert, sie werden immer effizienter und effektiver gestaltet und werden auch in Zukunft Gegenstand weiterer Entwicklungen sein.

Mit der Standardisierung und Ausweitung der Datenverfügbarkeit und Kommunikation im industriellen Umfeld (s. Industrie 4.0, Industrial Internet of Things), können zunehmend unterschiedlichste Sensoren, Aktoren, Kontrolleinheiten oder digitale Objekte über standardisierte Schnittstellen miteinander verknüpft und damit ein digitaler Zwilling ermöglicht werden. Die Industrie 4.0 hat in Bereichen wie Fertigung, Medizin und Logistik bedeutende Fortschritte in der Digitalisierung vorangetrieben, wohingegen diese Entwicklungen in der Landwirtschaft bisher nicht vollständig umgesetzt werden konnten (SREEDEVI und KUMAR, 2020; PURCELL und NEUBAUER, 2023). Vor dem Hintergrund der prognostizierten Weltbevölkerung von ca. 9,66 Mrd. Menschen bis zum Jahr 2050 (UNITED NATIONS, 2019) sind dringend Konzepte für eine effizientere Landwirtschaft erforderlich, um eine nachhaltige Ernährung sicherzustellen. Die umfassende Digitalisierung, einschließlich der Implementierung von digitalen Zwillingen, könnte eine entscheidende Rolle bei der Bewältigung dieser Herausforderung für die Landwirtschaft spielen (NASIRAHMADI und HENSEL, 2022). Dabei sind zunehmend das Tierwohl und die Reduktion klimaschädlicher Emissionen von Bedeutung, welche im Rahmen des Konzepts der Landwirtschaft 5.0 zusätzlich berücksichtigt werden (ABBASI, MARTINEZ und AHMAD, 2022). Ein bedeutender Wirtschaftszweig in der deutschen Landwirtschaft ist dabei die Milchviehhaltung, da nahezu jeder vierte Betrieb hierzulande Milchvieh hält (TERGAST und HANSEN, 2021). Die häufigste Bauweise zur Haltung von Milchvieh in Deutschland ist der Laufstall, der neun von zehn Ställen ausmacht (STATISTISCHES BUNDESAMT DEUTSCHLAND, 2020). Das Mikroklima in diesen Ställen wird maßgeblich durch die von außen aufgeprägten Umweltbedingungen und die Lüftungselemente beeinflusst (s. Abbildung 1.1).

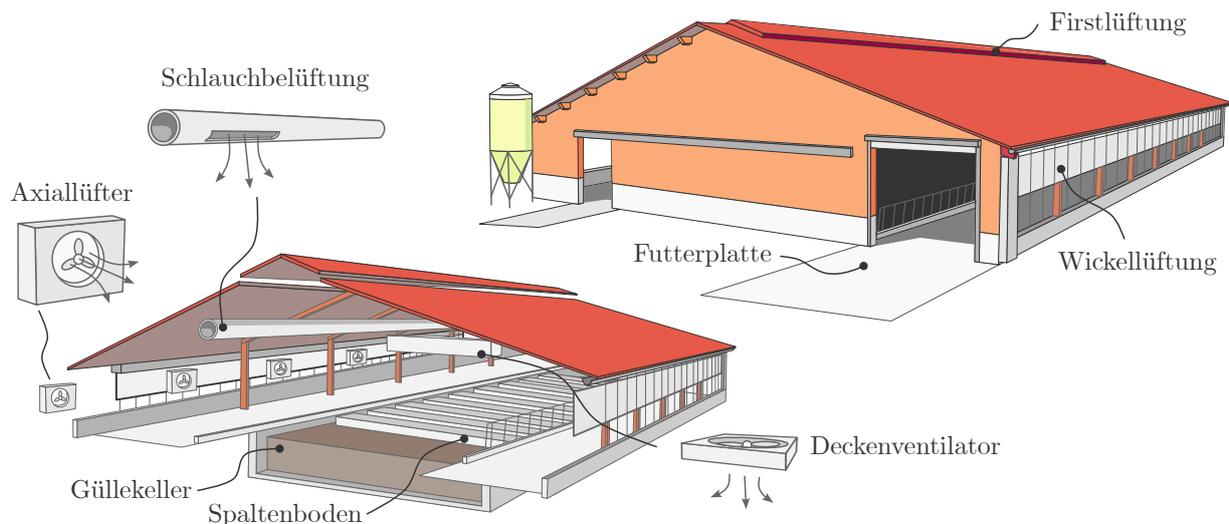


Abbildung 1.1: Ein typischer frei belüfteter Stall mit außen liegender Futterplatte in der Vollansicht (oben) und Schnittdarstellung (unten) sowie gekennzeichneten und detaillierten Lüftungselementen (eigene Darstellung)

Laut PYLIANIDIS, OSINGA und ATHANASIADIS (2021) könnte sich durch intelligente Regelungen mittels digitalem Zwilling ein großes Potenzial für Effizienzsteigerungen ergeben, indem bspw. der Energiebedarf der Gebäude gesenkt würde. In der Milchviehhaltung betrifft dies insbesondere die Lüftungssysteme der Ställe, die ca. 20 % des elektrischen Energiebedarfs eines landwirtschaftlichen Betriebs ausmachen (MONDACA, CHOI und COOK, 2019). VERDOUW et al. (2021) weisen darauf hin, dass digitale Zwillinge landwirtschaftlich Beschäftigte aus der Ferne und in Echtzeit mit Informationen versorgen könnten und so die Steuerung der Betriebsabläufe auf Basis digitaler Informationen stattfinden könnte, ohne dass auf die manuelle Beobachtung vor Ort zurückgegriffen werden müsste. Eine umfassendere Systemüberwachung durch digitale Zwillinge würde außerdem dazu beitragen, dass die Unsicherheiten der verschiedenen Prozesseinflüsse besser quantifiziert werden könnten (PYLIANIDIS, OSINGA und ATHANASIADIS, 2021).

JO, D.-H. PARK, H. PARK, KWAK et al. (2019) stellen heraus, dass der digitale Zwilling in der Lage sei, die für Tiere und Umwelt optimalen Bedingungen zu ermitteln und bei entsprechender Konnektivität mit den Akteuren auch einzustellen.

Der digitale Zwilling könnte als zentrales Werkzeug eingesetzt werden, um die Landwirtschaft der Zukunft nachhaltiger und effizienter zu gestalten – für Menschen, Tiere und Umwelt.

1.2 Problembeschreibung

Trotz der drängenden Thematik fassen die *Literature Reviews* von SREEDEVI und KUMAR (2020), PYLIANIDIS, OSINGA und ATHANASIADIS (2021), SLOB und HURST (2022) und ABBASI, MARTINEZ und AHMAD (2022) zusammen, dass sich ein Großteil der Arbeiten im prototypischen Stadium befinden und es weiteren Forschungsbedarf auf unterschiedlichen Gebieten gibt. Eine aktuellere Arbeit von PURCELL und NEUBAUER (2023) zeigt auf, dass ein zunehmender Anteil der Konzepte vom prototypischen Stadium in die Realisierung übergeht. Die Umsetzungen sind jedoch häufig im Themenfeld der Getreideernte oder urbanen Landwirtschaft zu finden (s. Abbildung 1.2).

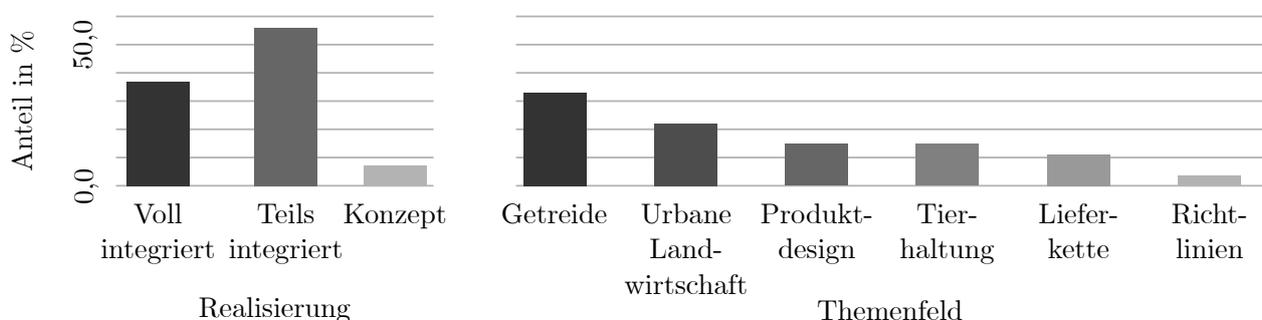


Abbildung 1.2: Einordnung der Literatur zum digitalen Zwilling in der Landwirtschaft gemäß den von PURCELL und NEUBAUER (2023) publizierten Daten (eigene Darstellung)

Die von PURCELL und NEUBAUER (2023) benannten Arbeiten mit Bezug zur Tierhaltung behandeln die hierzulande wichtige Thematik der Mikroklimatik nicht. Es werden entweder die Modellierung des Verhaltens und Rückschlüsse auf den Gesundheitszustand fokussiert (NISWAR et al., 2018; ERDÉLYI und JÁNOSI, 2019) oder Sensordaten für eine Optimierung des Energiebedarfs in Schweineställen genutzt (JO, D.-H. PARK, H. PARK und S.-H. KIM, 2018; JO, D.-H. PARK, H. PARK, KWAK et al., 2019).

Die Analyse von PURCELL und NEUBAUER (2023) kondensiert die Herausforderungen und Potenziale für die Umsetzung digitaler Zwillinge in der Landwirtschaft 4.0 u. a. auf folgende Punkte:

- (i) Simulationen landwirtschaftlicher Anwendungsfälle sind äußerst begrenzt verfügbar.
- (ii) Mithilfe von maschinellem Lernen könnten digitale Zwillinge komplexer landwirtschaftlicher Systeme entwickelt werden.
- (iii) Es besteht weiterer Bedarf der Konzeptentwicklung für die Datensicherheit, Datenlage, Datenqualität und Datenverarbeitung.

(i) Die begrenzte Verfügbarkeit von Simulationen ist vor allem relevant, da eine Simulation lediglich ein spezifisches Problem mit den a priori bekannten Randbedingungen abdeckt. Zudem müssen Simulationen verifiziert und validiert werden, um aussagekräftige Ergebnisse zu liefern (ASME, 2009). Die Verifizierung kann vollständig am digitalen Objekt durchgeführt werden, wohingegen es für die Validierung Messungen der relevanten Quantitäten benötigt. Damit geht die Auswahl bzw. Entwicklung von Messmitteln einher, die für den Einsatz in Stallungen geeignet sind.

Für die Milchviehhaltung bedeuten dies, dass verifizierte und validierte Strömungssimulationen benötigt werden, bei denen unterschiedliche Parameter variiert werden können, um eine Abschätzung des Parametereinflusses vornehmen zu können. Wobei die Parameter vielfältig sind und sowohl die Geometrie als auch die Randbedingungen in der Simulation betreffen, wie z. B.: Windgeschwindigkeit, Windrichtung, Temperatur, Geometrien am Stall oder Luftfeuchte.

(ii) Die größte Herausforderung ist die Realisierung des digitalen Abbilds eines komplexen landwirtschaftlichen Systems, mit der Anforderung dieses zur Regelung einzusetzen. Basierend auf einem digitalen Abbild, welches das Verhalten anhand der Sensordaten prognostiziert, soll eine Optimierung des laufenden Systems vorgenommen werden. Damit die Simulationsergebnisse rechtzeitig an das physische Objekt gelangen und die Regelung eingreifen kann, müssen die klassischen Simulationen in Echtzeit-Simulationen überführt und die Datenübertragung echtzeitfähig gestaltet werden.

Eine Möglichkeit, Echtzeit-Modelle zu erzeugen, ist die Approximation des Zusammenhangs zwischen Eingangsparametern und Systemantwort mittels Metamodellen. Klassische Ansätze zur Metamodellierung sind allerdings fehleranfällig (SIEBERTZ, BEBBER und HOCHKIRCHEN, 2017): Da bei komplexen Simulationsmodellen der Zusammenhang zwischen Eingängen und

Systemantworten typischerweise nicht bekannt ist, führt eine falsche Wahl der Grundfunktion schnell zu ungenauen Metamodellen. Um dieser Problematik aus dem Wege zu gehen, werden Verfahren wie beispielsweise künstliche neuronale Netze oder Kriging eingesetzt, die sich eigenständig an gegebene Datenpunkte und Zusammenhänge anpassen. Damit ein künstliches neuronales Netz diese Aufgaben beherrschen kann, wird i. d. R. ein großer Datensatz von mehreren zehntausenden Eingangsdaten mit der zugehörigen Lösung für das Training benötigt (REY und WENDER, 2018). Effektiv ist dieser Ansatz bisher also nur für repetitive Aufgaben oder bei bereits ausreichend vorhandener Datenbasis. Außerhalb der Massenproduktion ist diese Methode für die oft komplexen technischen Produkte oder Prozesse in der Landwirtschaft demnach nicht sinnvoll.

Einen Lösungsansatz könnten Physik-informierte neuronale Netze bieten, welche durch das Inkorporieren der zu Grunde liegenden Gleichungen und Randbedingungen in die Verlustfunktion, gegenüber klassischen künstlichen neuronalen Netzen, einen datensparsamen Ansatz bieten (RAISSI, YAZDANI und KARNIADAKIS, 2020; CHAUDHARI, KULKARNI und DAMODARAN, 2021). Aufgrund des geringen Forschungsstandes ist hierbei die Eignung für komplexe Strömungsgebiete und hohe Reynolds-Zahlen zu prüfen und das Konzept ggf. dahingehend zu erweitern.

(iii) Die Lücke der Datensicherheit, Datenlage und Datenqualität kann mittels geeigneter Kommunikationskonzepte geschlossen werden, die es für die Landwirtschaft 4.0 in großen Teilen noch zu entwickeln gilt. Ein relevanter Aspekt ist hierbei der Einbezug flexibler Kommunikationsschnittstellen, um die Aktoren und Sensoren unterschiedlicher Hersteller einbinden zu können. Ebenso sollte die Kommunikation nach Bedarf on- und offline realisierbar sein, da nicht jeder Hof über eine vollständige Netzabdeckung verfügt. Für das Online-Konzept ist die informationstechnische Sicherheit ein maßgeblicher Entwicklungsabschnitt. Um den missbräuchlichen Zugriff zu behindern, sind die Kontenverwaltung und das Datenbankmanagement nach aktuellen Standards sicher zu gestalten. Dennoch ist der Zugriff externer Verarbeitungsprogramme erforderlich, um bspw. Werte für eine Regelung abzurufen.

1.3 Aufbau der Arbeit

Der Aufbau der Arbeit gliedert sich in den Grundlagenteil, die Erzeugung einer validen Datenbasis, anschließende Nutzung der Daten für eine Metamodellierung sowie die Integration des Metamodells in einen digitalen Zwilling. Es folgen Diskussion und Fazit. Abbildung 1.3 visualisiert die Struktur.

Das Kapitel 2 vermittelt die erforderlichen theoretischen Grundlagen der Strömungsdynamik, statistischen Versuchsplanung und (Physik-informierten) künstlichen neuronalen Netze. Über die Grundlagen hinaus wird der spezifische Stand der Forschung zu den Themen der Arbeit dargelegt, woraus sich der wissenschaftliche Beitrag der Arbeit ableitet. Im Ergebnisteil der Arbeit wird das Kapitel 3 die experimentellen Ergebnisse zur Erzeugung einer

validen Datenbasis dokumentieren. Diese Datenbasis wird anschließend zur Validierung des in Kapitel 4 erstellten numerischen Simulationsmodells genutzt. Die experimentellen und numerischen Ergebnisse werden in Kapitel 5 aufgegriffen, um eine Methode für ein echtzeitfähiges Metamodell der Strömungsvorhersage zu generieren. Das Kapitel 6 beschreibt die Implementierung dieses echtzeitfähigen Metamodells in eine Kommunikationsarchitektur für den digitalen Zwilling. Abschließend diskutiert und reflektiert das Kapitel 7 die Ergebnisse der Arbeit. Kapitel 8 resümiert die vorliegende Arbeit und gibt einen Ausblick für zukünftige Arbeiten.

Ergänzende Daten, Abbildungen und der Programmcode der entwickelten Programme sind im Anhang hinterlegt.

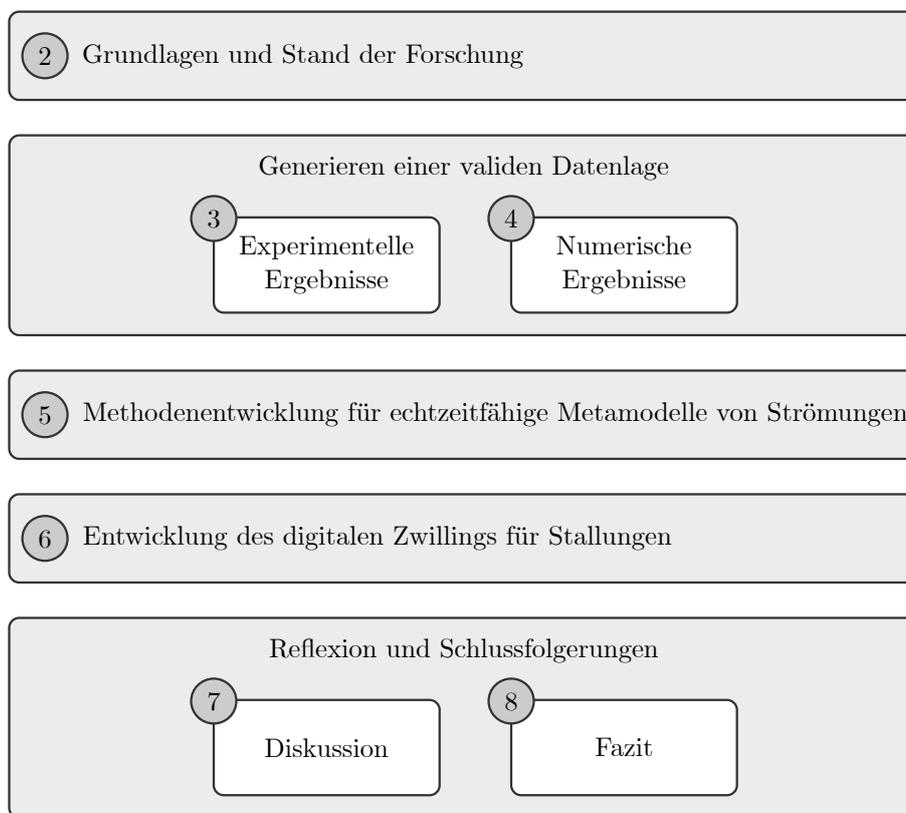


Abbildung 1.3: Aufbau der Arbeit (eigene Darstellung)

Grundlagen und Stand der Forschung

Dieses Kapitel wird die für die Arbeit relevanten Grundlagen und den Stand der Forschung vermitteln. Jedes Unterkapitel behandelt einen essentiellen Entwicklungsschritt zum digitalen Zwilling für Stallungen, auf dem die Arbeit aufbaut. Im Kapitel 2.1 wird zunächst das Konzept des digitalen Zwillings beschrieben, mit dem die durchgängige Kommunikation aller für diese Arbeit relevanten Komponenten umgesetzt werden soll. Darauffolgend erläutert das Kapitel 2.2 die mathematischen Grundlagen der numerischen Strömungssimulation, mit der Fluid-, Wärme- und Stofftransport in und um den virtuellen Stall berechnet werden können. Im Kapitel 2.3 wird die Methode der statistischen Versuchsplanung vorgestellt, die es ermöglicht, effizientere Experimente zu entwerfen und die Anzahl der Durchläufe zu reduzieren. Kapitel 2.4 dokumentiert die Grundlagen (Physik-informierter) neuronaler Netze, welche für die echtzeitfähige Vorhersage der Strömungen eingesetzt werden sollen. Das Kapitel 2.5 dokumentiert den Stand der Forschung zur Um- und Durchströmung von (landwirtschaftlichen) Gebäuden, der Anwendung von Physik-informierten neuronalen Netzen zur Approximation der Navier-Stokes-Gleichungen und die Entwicklungen zum digitalen Zwilling in der Landwirtschaft. Abschließend beantwortet die Analyse des Standes der Forschung, welchen wissenschaftlichen Beitrag die Arbeit leisten kann (Kapitel 2.6).

2.1 Grundlagen des digitalen Zwillings

Die in der Einleitung dargelegte Entstehungsgeschichte des digitalen Zwillings verdeutlicht bereits die Notwendigkeit einer umfangreichen Datenverfügbarkeit und Konnektivität der

Komponenten, um die Realisierung eines digitalen Zwillings zu ermöglichen. Erste Prinzipien der in Abbildung 2.1 dargestellten Kopplung zwischen realer und digitaler Welt gehen auf die von GRIEVES (2005) publizierten Konzepte zum Produktlebenszyklusmanagement zurück, ohne jedoch bereits den Begriff des digitalen Zwillings zu verwenden. Die Entwicklungsschritte für den digitalen Zwilling lassen sich demnach mit einer sukzessiven Erweiterung der Digitalisierung des gesamten Produktlebenszyklus erklären.

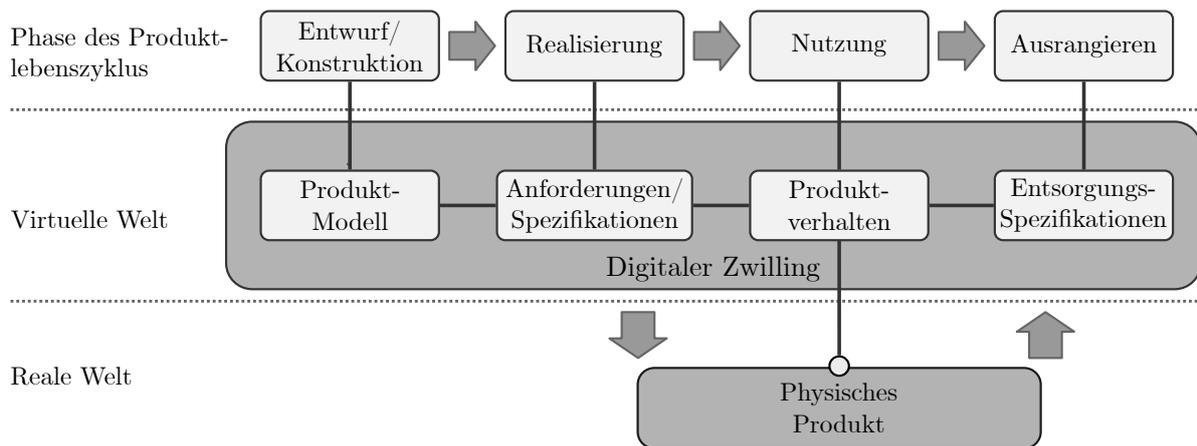


Abbildung 2.1: Die Rolle des digitalen Zwillings über den Produktlebenszyklus in Anlehnung an VERDOUW et al. (2021)

In der Entwurfs- oder Konstruktionsphase existiert noch kein physischer Zwilling, und die Analyse und Optimierung des Produkts oder Prozesses findet ausschließlich im virtuellen Raum statt. Ein Beispiel dafür ist die Untersuchung der Montage oder des Platzbedarfs anhand von computergestützt konstruierten Geometriemodellen (engl. *computer-aided design* (CAD)) oder die Analyse des Produkt- oder Prozessverhaltens mittels Simulationen. Gemäß VERDOUW et al. (2021) folgt erst im Anschluss die Entstehung des physischen Zwillings. Während der operativen Nutzung erfassen Sensoren den aktuellen Zustand des physischen Produkts, den es für historische Betrachtungen zu archivieren gilt. Überdies kann der zukünftige Zustand durch die Modellierung des Verhaltens prognostiziert und mithilfe von Aktoren beeinflusst werden. GRIEVES (2005) betont dabei die Visualisierung als eine Notwendigkeit für die bestmögliche Analyse des Systemzustands und weist auf die Einschränkungen durch die Rechenleistung hin. Schließlich erfolgt die Entsorgungsphase für das physische Objekt, während das digitale Objekt länger für eine Rückverfolgbarkeit bestehen bleiben kann.

Basierend auf der Kopplung von physischen und digitalen Objekten unterteilen KRITZINGER et al. (2018) den Grad der Realisierung des digitalen Zwillings in drei Ebenen. Beim digitalen Modell (Ebene 1) erfolgt ein manueller Datenfluss zwischen dem physischen und dem digitalen Objekt. Für den digitalen Schatten (Ebene 2) ist der Datenfluss unidirektional automatisiert, indem beispielsweise Sensordaten des physischen Objekts automatisch in eine Simulation integriert werden. Der digitale Zwilling (Ebene 3) stellt die konsequente und vollständige Automatisierung des Datenflusses zwischen dem physischen und dem digitalen

Objekt dar. VERDOUW et al. (2021) betrachten die Einteilung differenzierter und fokussieren die landwirtschaftliche Anwendung. In Abbildung 2.2 sind die daraus resultierenden sechs Typen digitaler Zwillinge dargestellt. In der oberen Zeile ist der imaginäre, überwachende und prädiktive digitale Zwilling abgebildet. Die Illustrationen zum präskriptiven, autonomen und historischen digitalen Zwilling sind in der unteren Zeile dargestellt.

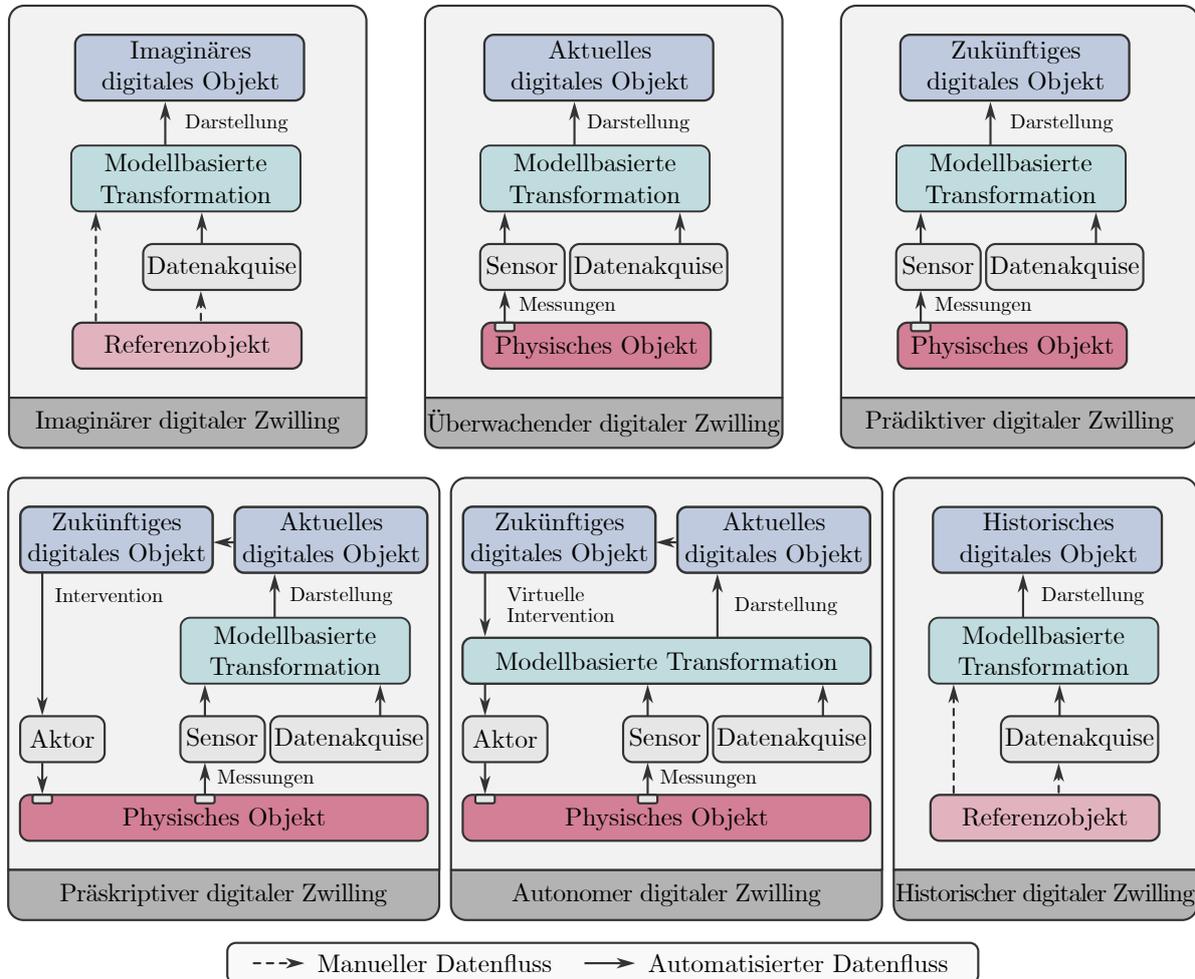


Abbildung 2.2: Darstellung der unterschiedlichen Konzepte digitaler Zwillinge nach VERDOUW et al. (2021) (aus dem engl.)

Der imaginäre digitale Zwilling ist eine Konzeptdarstellung eines bislang nicht real existierenden Objekts. Er enthält Informationen, wie funktionale Anforderungen, 3D-Modelle, Material- und Ressourcen-Spezifikationen, die benötigt werden, um den physischen Zwilling zu realisieren. Imaginäre digitale Zwillinge können auch das Verhalten von Objekten simulieren, die bisher nicht existierten. Ein überwachender digitaler Zwilling ist eine digitale Darstellung des aktuellen Zustands und Verhaltens des physischen Objekts. Er ist (nahezu) in Echtzeit mit seinem physischen Zwilling verbunden und wird verwendet, um dessen Zustand, Betrieb und Umgebung zu überwachen. Der vorausschauende digitale Zwilling nutzt prädiktive Analytik, wie statistische Prognosen, Simulationen und maschinelles Lernen, um

digitale Vorhersagen über zukünftige Zustände und Verhaltensweisen physischer Objekte zu treffen. Die Vorhersage erfolgt dynamisch, basierend auf (nahezu) Echtzeitdaten des physischen Zwillings. Mithilfe des präskriptiven digitalen Zwillings können korrigierende und präventive Maßnahmen entwickelt und am realen Objekt auch eingestellt werden. Die Maßnahmen für den optimalen Betriebszustand werden aus den Ergebnissen des überwachenden und vorausschauenden digitalen Zwillings abgeleitet. Die Entscheidungen über die empfohlenen Maßnahmen werden immer noch von Menschen getroffen, die auch die Durchführung von Interventionen vor Ort oder aus der Ferne auslösen. Ein autonomer digitaler Zwilling kontrolliert das Verhalten von realen Objekten vollständig, ohne dass Menschen vor Ort oder aus der Ferne eingreifen müssen. Autonome Zwillinge können auch zu selbstadaptiven Systemen werden, die in der Lage sind, ihre Umgebung zu erlernen, ihre eigenen Serviceanforderungen selbst zu diagnostizieren und sich an das Nutzungsverhalten anzupassen. VERDOUW et al. (2021)

Über die Konzepte hinaus existieren bereits Standards für die Umsetzung einer durchgängigen Kommunikation. HEIDEL et al. (2017) beschreiben das Referenzarchitekturmodell für die Industrie 4.0, welches geeignete Kommunikationsprotokolle festlegt. Die Schnittstelle *Open Platform Communication Unified Architecture* (OPC UA) ist dabei für die Fertigungsebene definiert, während es keinen Standard für die Kommunikation auf Unternehmensebene oder für die übergreifende Konnektivität gibt. Des Weiteren werden Aspekte der IT-Sicherheit und der virtuellen Modellbildung diskutiert, jedoch werden keine spezifischen Standards vorgegeben.

2.2 Grundlagen der numerischen Strömungssimulation

Der erste Entwicklungsschritt des digitalen Zwillings besteht darin, ein digitales Modell zu erstellen. Im Fall des frei belüfteten Kuhstalls sollte dieses Modell die wesentlichen Charakteristika der Durchströmung erfassen, da sie maßgeblich für den Stoff- und Energietransport zwischen dem Stall und seiner Umgebung verantwortlich sind. Frei belüftete Kuhställe nutzen die erzwungene Konvektion durch den Wind, um das Mikroklima im Stall zu regulieren, einschließlich der Luftfeuchtigkeit, Lufttemperatur und Konzentration lokal emittierter Gase. Da die Grundgleichungen der Strömungsmechanik nur für spezielle Fälle analytisch gelöst werden können, werden numerische Simulationen eingesetzt, um eine Näherungslösung mittels iterativer Verfahren zu berechnen. Die Grundlagen der Strömungsmechanik und die darauf aufbauenden numerischen Verfahren werden in den Kapiteln 2.2.1 bis 2.2.3 beschrieben. Da die numerischen Verfahren und deren Ergebnisse mit Fehlern und Unsicherheiten behaftet sein können (aufgrund der Näherungslösungen), ist es erforderlich, diese im Rahmen einer Verifizierung abzuschätzen und in einer Validierung mit der Realität abzugleichen. Hierbei werden verschiedene Vorgehensweisen und Metriken empfohlen, die im Kapitel 2.2.4 dokumentiert sind.

2.2.1 Grundgleichungen der Strömungsmechanik

Die mathematischen Beschreibungen dieses Kapitels basieren im Wesentlichen auf den Ausführungen von FERZIGER, PERIĆ und STREET (2020) und LECHLER (2018).

Der Erhaltungssatz besagt, dass eine Erhaltungsgröße innerhalb eines Kontrollvolumens (KV) konstant bleibt, also weder vernichtet noch erzeugt werden kann. Angewendet auf eine nicht isotherme Strömung müssen demnach die Größen Masse, Impuls und Energie erhalten bleiben. Aus dem Kontrollvolumen-Ansatz ergibt sich das Reynolds-Transporttheorem, mit ϕ als intensive Zustandsgröße:

$$\frac{d}{dt} \int_{V_{KM}} \rho \phi \, dV = \frac{d}{dt} \int_{V_{KV}} \rho \phi \, dV + \int_{S_{KV}} \rho \phi (\mathbf{u} - \mathbf{u}_S) \cdot \mathbf{n} \, dS. \quad (2.1)$$

Hierbei steht t für die Zeit, ρ für die Dichte und V_{KM} für das von der Masse eingenommene Volumen. V_{KV} ist das Kontrollvolumen und S_{KV} steht für die Oberfläche, die das Kontrollvolumen umschließt. \mathbf{n} ist der orthogonal auf der Oberfläche S stehende Einheitsvektor. \mathbf{u} ist der Geschwindigkeitsvektor des Fluids, wohingegen \mathbf{u}_S den Geschwindigkeitsvektor der Oberfläche beschreibt. Letzterer ist bei der verwendeten Betrachtungsweise eines ortsfesten KV mit $\mathbf{u}_S = 0$ anzunehmen.

Für die **Massenerhaltung** gilt $\phi = 1$, womit sich aus Gleichung 2.1 die Integralform der Kontinuitätsgleichung ergibt:

$$\frac{\partial}{\partial t} \int_V \rho \, dV + \int_S \rho \mathbf{u} \cdot \mathbf{n} \, dS = 0. \quad (2.2)$$

Äquivalent zur Massenerhaltung kann die Integralform der **Impulserhaltung** anhand des Reynolds-Transporttheorems hergeleitet werden, indem $\phi = \mathbf{u}$ gesetzt wird:

$$\frac{\partial}{\partial t} \int_V \rho \mathbf{u} \, dV + \int_S \rho \mathbf{u} \mathbf{u} \cdot \mathbf{n} \, dS = \sum \mathbf{f}. \quad (2.3)$$

Die rechte Seite der Gleichung summiert die auf das KV wirkenden Körperkräfte (z. B. Schwerkraft) und Oberflächenkräfte (z. B. Druck-, Normal- und Scherspannungen). Für newtonsche Fluide ergibt sich die Schreibweise in Integralform:

$$\frac{\partial}{\partial t} \int_V \rho \mathbf{u} \, dV + \int_S \rho \mathbf{u} \mathbf{u} \cdot \mathbf{n} \, dS = \int_S \mathbf{T} \cdot \mathbf{n} \, dS + \int_V \rho \mathbf{f} \, dV, \quad (2.4)$$

wobei \mathbf{T} für den Spannungstensor und \mathbf{f} für die Körperkräfte (pro Masseneinheit) steht.

Für die Herleitung der **Energieerhaltung** lässt sich die Erhaltung skalarer Größen allgemeiner formulieren:

$$\frac{\partial}{\partial t} \int_V \rho \phi \, dV + \int_S \rho \phi \mathbf{u} \cdot \mathbf{n} \, dS = \sum \mathbf{f}_\phi . \quad (2.5)$$

Die rechte Seite der Gleichung kommt durch Transportmechanismen zustande, die nicht durch Quellen, Senken oder Konvektion zu erklären sind. Dies wird durch eine Gradientenapproximation beschrieben:

$$\sum \mathbf{f}_\phi = \int_S \Gamma \nabla \phi \cdot \mathbf{n} \, dS , \quad (2.6)$$

wobei Γ für die Diffusivität der Quantität ϕ steht.

Aus Gleichung 2.5 und 2.6 lässt sich mittels Fourier-Gesetz (Wärmediffusion) und dem Fickschen Gesetz (Massendiffusion) eine für die meisten technischen Strömungen gültige Form der Energieerhaltung formulieren:

$$\frac{\partial}{\partial t} \int_V \rho h \, dV + \int_S \rho h \mathbf{u} \cdot \mathbf{n} \, dS = \int_S \lambda_Q \nabla T \cdot \mathbf{n} \, dS + \int_V (\mathbf{u} \cdot \nabla p + S : \nabla \mathbf{u}) \, dV + \frac{\partial}{\partial t} \int_V p \, dV , \quad (2.7)$$

wobei h die spezifische Enthalpie und e die innere Energie kennzeichnen. Die Temperatur ist mit T angegeben und der Wärmeleitkoeffizient mit λ_Q . Für inkompressible Fluide mit konstanter spezifischer Wärmekapazität kann diese Gleichung zu einer Konvektions-Diffusionsgleichung der Temperatur vereinfacht werden:

$$\frac{\partial}{\partial t} \int_V \rho T \, dV + \int_S \rho T \mathbf{u} \cdot \mathbf{n} \, dS = \int_S \frac{\mu}{\text{Pr}} \nabla T \cdot \mathbf{n} \, dS , \quad (2.8)$$

wobei Pr die Prandtl-Zahl ist (Verhältnis von Impuls- zu Wärmediffusivität).

Diese Vereinfachung kann auf die **Erhaltungsgleichung für Konzentrationen von Spezies** übertragen werden. Dabei wird die Temperatur T durch die Konzentration c ersetzt und die Prandtl-Zahl Pr durch die Schmidt-Zahl Sc , welche das Verhältnis von Impuls- zu Speziesdiffusivität beschreibt:

$$\frac{\partial}{\partial t} \int_V \rho c \, dV + \int_S \rho c \mathbf{u} \cdot \mathbf{n} \, dS = \int_S \frac{\mu}{\text{Sc}} \nabla c \cdot \mathbf{n} \, dS . \quad (2.9)$$

Die zuvor beschriebenen Integralformen der Erhaltungsgleichungen können mittels Umformung und der Annahme, sie wirken in einem infinitesimal kleinen Volumen, in eine Divergenzform überführt werden. Dazu wird zunächst der Gaußsche Divergenzatz angewandt,

anhand dessen das Volumenintegral durch ein Flächenintegral ersetzt wird. Der Divergenz-
satz besagt, dass die Ansammlung einer Quantität innerhalb eines Volumens nur über den
Fluss der Quantität über die Oberflächen des Volumens erfolgen kann (s. Abbildung 2.3).

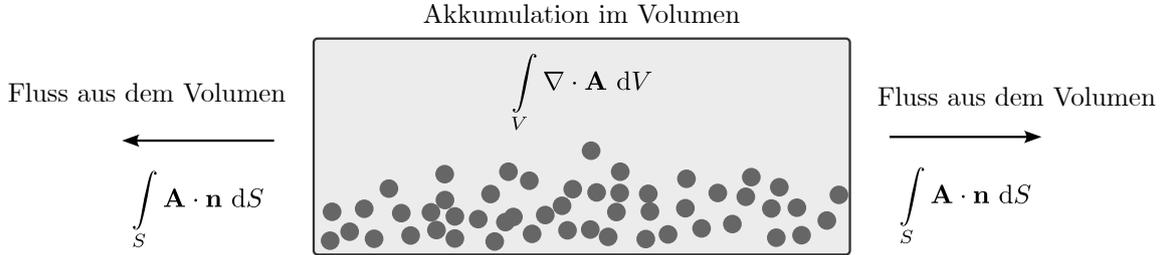


Abbildung 2.3: Visualisierung des Gaußschen Divergenztheorems in der Anwendung auf ein
Vektorfeld \mathbf{A} . Die Akkumulation im Volumen entspricht dem Fluss über die
Oberflächen des Volumens (eigene Darstellung)

Es ergibt sich die koordinatenfreie Divergenzform der **Massenerhaltungsgleichung** zu:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 . \quad (2.10)$$

Für die **Impulserhaltung** ergibt sich:

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u}) = \nabla \cdot \mathbf{T} + \rho \mathbf{f} , \quad (2.11)$$

mit:

$$\mathbf{T} = - \left[p + \frac{2}{3} \mu (\nabla \cdot \mathbf{u}) \right] \mathbf{I} + 2\mu \mathbf{D} , \quad (2.12)$$

wobei p der statische Druck, μ die dynamische Viskosität, \mathbf{I} der Einheitstensor und \mathbf{D} die
Deformationsrate ist. Die Deformationsrate steht für $\mathbf{D} = \frac{1}{2}[\nabla \mathbf{u} + (\nabla \mathbf{u})^T]$.

Der Term $\rho \mathbf{f}$ entfällt für diese Arbeit, da keine Gravitationskräfte betrachtet werden. Außer-
dem kann ρ bei inkompressiblen Fluiden als konstant angenommen werden, woraus für die
Kontinuitätsgleichung $\nabla \cdot \mathbf{u} = 0$ folgt. Wird dies auf Gleichung 2.11 angewendet, ergibt sich:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \mathbf{u}) = -\frac{1}{\rho} \nabla p + \nu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T) , \quad (2.13)$$

wobei $\nu = \mu/\rho$ die kinematische Viskosität ist.

Für die **Erhaltung der Energie und skalarer Größen** kann die Divergenzform allgemein
formuliert werden. Erneut wird das Fluid als inkompressibel vereinfacht:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u} \phi) = \frac{1}{\rho} (\nabla \cdot (\Gamma \nabla \phi)) + \mathbf{q}_\phi . \quad (2.14)$$

2.2.2 Finite-Elemente-Methode und Finite-Volumen-Methode

Die hergeleiteten Navier-Stokes-Gleichungen lassen sich lediglich für Spezialfälle analytisch lösen. Daher werden die (Integro-)Differentialgleichungen für komplexere Geometrien numerisch (näherungsweise) gelöst. Die Geometrie wird dazu in viele kleine, aber endlich viele (finite) Elemente unterteilt. Dieser Prozess der Diskretisierung approximiert die stetigen Gleichungen, indem ein Gleichungssystem algebraischer Gleichungen gebildet wird, welches anschließend numerisch gelöst werden kann. Beim Diskretisieren der stetigen Gleichungen gibt es unterschiedliche Methoden, wie die Finite-Differenzen-Methode (FDM), Finite-Elemente-Methode (FEM) und Finite-Volumen-Methode (FVM). In COMSOL MULTIPHYSICS wird die FEM eingesetzt, wohingegen viele andere Programme für *Computational Fluid Dynamics* (CFD) die FVM verwenden. Das Kapitel beschreibt daher vorwiegend die FEM gemäß der Implementierung für COMSOL MULTIPHYSICS (FONTES, 2018), geht aber auf die relevanten Unterschiede ein.

Die Anwendung der Methoden auf unterschiedliche Transportgleichungen unterscheidet sich grundsätzlich nicht, weshalb an dieser Stelle eine allgemeine Form der Transportgleichung verwendet wird. Die für die FEM erforderliche schwache Formulierung der Gleichung kann mittels Integration und Umformung aus Gleichung 2.14 entwickelt werden. Die Terme auf der rechten Seite der Gleichung sind durch einen generischen Quellterm q ersetzt. Die Flussterme sind als Λ zusammengefasst:

$$\int_{\Omega} \frac{\partial \phi}{\partial t} \varphi_t \, dV - \int_{\Omega} \Lambda \cdot \nabla \varphi_t \, dV + \int_{\partial \Omega} \Lambda \cdot \mathbf{n} \varphi_t \, dS = \int_{\Omega} q \varphi_t \, dV, \quad (2.15)$$

wobei Ω für das Berechnungsgebiet, $\partial \Omega$ für die Berandung und φ_t für die Ansatzfunktionen steht. In den Ansatzfunktionen besteht der wesentliche Ansatz der FEM: Die stetige Gleichung wird in n Gleichungen überführt, indem die Terme der Gleichung mit n Basisfunktionen gewichtet werden. Bei der im Folgenden vorgestellten Galerkin-Methode sind die Basisfunktionen mit den Ansatzfunktionen identisch ($\Psi = \varphi_t$). Beispiele für Ansatzfunktionen sind die aus dem Pascalschem Dreieck entwickelten Lagrange-Polynome. Ein Sonderfall ist die konstante Ansatzfunktion $\varphi_t = 1$. Eingesetzt ergibt sich die bekannte Form aus Gleichung 2.5 und 2.6:

$$\int_{\Omega} \frac{\partial \phi}{\partial t} \, dV + \int_{\partial \Omega} \Lambda \cdot \mathbf{n} \, dS = \int_{\Omega} q \, dV. \quad (2.16)$$

Hieraus wird deutlich, dass der für die FEM verwendete Ansatz lediglich einen Sonderfall der schwachen Formulierung der Transportgleichung darstellt.

Für die Methode der FEM wird vorausgesetzt, dass Gleichung 2.15 für die Auswahl der finiten Anzahl von Ansatzfunktionen $\varphi_{te} \subset \varphi_t$ im gesamten Rechengebiet Ω gilt (Abbildung 2.4, links). Die FVM hingegen unterteilt das Rechengebiet in Kontrollvolumen und fordert, dass

Gleichung 2.16 für jedes finite Volumen $\Omega_e \subset \Omega$ gilt (Abbildung 2.4, rechts). Der Unterschied der Methoden liegt demnach in der Diskretisierung.

Es zeigt sich, dass bei der FEM lediglich die direkte Umgebung des Knotens relevant ist (Abbildung 2.4, links, hellgrau). Der dritte Term auf der linken Seite in Gleichung 2.15 ist aufgrund des lokalen Ansatzes ausschließlich für Elemente zu berücksichtigen, deren Fläche oder Knoten an der äußeren Gebietsgrenze $\partial\Omega$ liegen (Abbildung 2.4, markierter Knoten unten links).

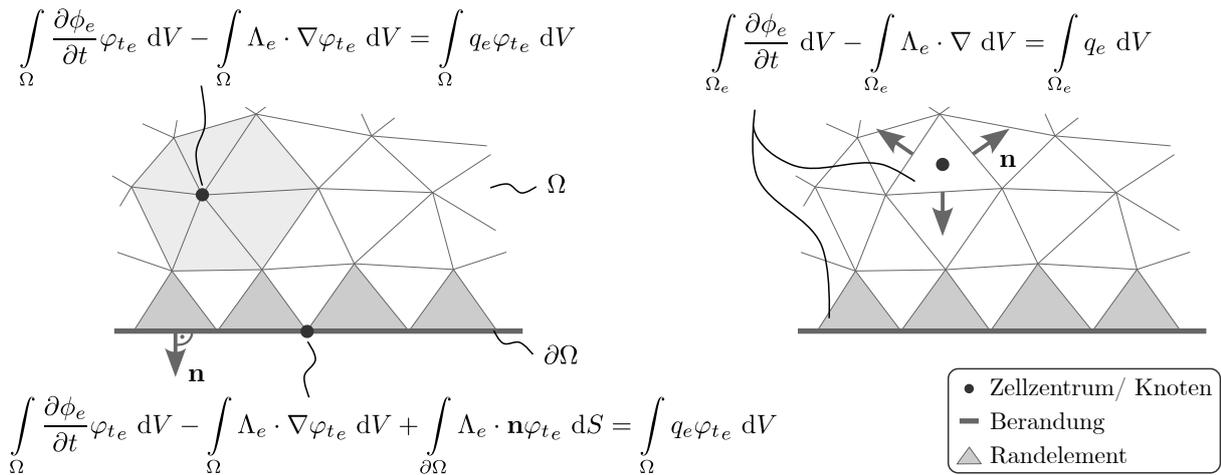


Abbildung 2.4: Darstellung eines diskretisierten Rechengebiets Ω , mit der Gebietsgrenze $\partial\Omega$ und der Darstellung einer Diskretisierung durch die FEM (links) und FVM (rechts), in Anlehnung an FONTES (2018) (eigene Darstellung)

Bei der FVM ist die Anzahl der einzubeziehenden Elemente von der Ordnung des Diskretisierungsschemas abhängig. Bei einem Diskretisierungsschema höherer Ordnung müssen mehr Elemente einbezogen werden. Das Diskretisierungsschema approximiert den Fluss einer Quantität über die Oberfläche zwischen den Elementen mittels Interpolation von Zellzentrum zu Zellzentrum. Abbildung 2.5 (S. 16) stellt die FEM mit linearer Ansatzfunktion (links) der FVM mit einem Diskretisierungsschema erster Ordnung gegenüber (rechts).

Beide Methoden führen zu einem algebraischen Gleichungssystem mit dünnbesetzten Matrizen, die gegebenenfalls vorkonditioniert werden müssen. Ein Beispiel hierfür ist die LU-Zerlegung, die im Kontext des algebraischen Mehrgitterverfahrens angewendet werden kann. Es gibt unterschiedliche iterative Lösungsverfahren, wobei häufig die *Generalized minimal residual method* (GMRES) eingesetzt wird, die zu der Gruppe der Krylow-Unterraum-Verfahren gehört. Dieses Verfahren eignet sich insbesondere für dünn besetzte Matrizen (SAAD und SCHULTZ, 1986). Weiterführende Erläuterungen zu den Lösungsverfahren für die FEM und FVM lassen sich FERZIGER, PERIĆ und STREET (2020) oder RIEG, HACKENSCHMIDT und ALBER-LAUKANT (2012) entnehmen.

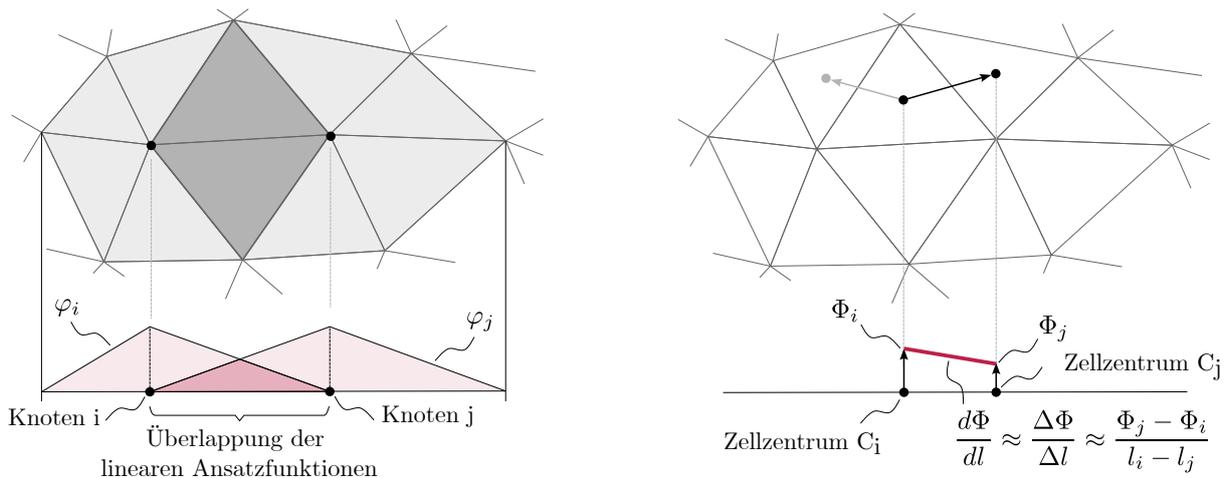


Abbildung 2.5: Illustration der Anwendung von (linearen) Ansatzfunktionen in der FEM (links) und eines Diskretisierungsschemas erster Ordnung für die FVM (rechts) (eigene Darstellung)

2.2.3 Reynolds-gemittelte Gleichungen und Turbulenzmodellierung

Die numerische Berechnung der zuvor hergeleiteten vollständigen räumlichen und zeitlichen Auflösung der Navier-Stokes-Gleichungen ist für heutige Computer rechenintensiv, da eine sehr feine Diskretisierung erforderlich ist, die wiederum zu einem sehr großen Gleichungssystem führt. Daher werden für technische Anwendungen oft die Reynolds-gemittelten Navier-Stokes-Gleichungen (RANS-Gleichungen) genutzt, die lediglich die mittleren und niederfrequenten Schwankungen in der Strömung auflösen und die hochfrequenten Schwankungen über Ersatzmodelle (Turbulenzmodelle) beschreiben.

Der Ansatz beginnt mit der Überlegung, dass jede Zielgröße in einen gemittelten Anteil und dessen Schwankungsgröße zerlegt werden kann (s. Abbildung 2.6):

$$\Phi = \bar{\Phi} + \Phi' . \quad (2.17)$$

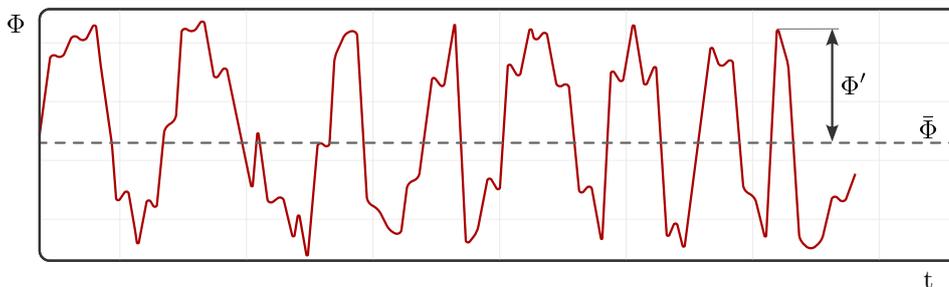


Abbildung 2.6: Zerlegung einer Zielgröße in zeitliche Mittelung plus turbulente Fluktuation (eigene Darstellung)

Wendet man dies auf die Impulsgleichungen an, erhält man nach Zerlegung der Größen in seinen Mittelwert und die Fluktuation folgende Gleichung in Indexnotation:

$$\frac{\partial \bar{u}_i + u'_j}{\partial t} + \frac{\partial (\bar{u}_i + u'_i)(\bar{u}_j + u'_j)}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p} + p'}{\partial x_i} + \nu \frac{\partial^2 \bar{u}_i + u'_i}{\partial x_j \partial x_j}. \quad (2.18)$$

Die Idee von Reynolds war es, die gesamte Gleichung 2.18 zeitlich zu mitteln. Dadurch entfallen die gemittelten turbulenten Fluktuationen. Nach Umstellung der gemittelten Gleichung fallen weitere Terme weg, und es bleibt lediglich ein zusätzlicher Term mit der Korrelation der turbulenten Fluktuationen erhalten:

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \bar{u}_j \bar{u}_i}{\partial x_j} + \frac{\partial \overline{u'_j u'_i}}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j}. \quad (2.19)$$

Nach Umstellung der Gleichung 2.19 ergibt sich die bekannte Form, mit den Reynolds-Spannungen auf der rechten Seite der Gleichung:

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \bar{u}_j \bar{u}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j} \underbrace{\left(\nu \frac{\partial \bar{u}_i}{\partial x_j} - \overline{u'_j u'_i} \right)}_{=\tau_{ij}/\rho}. \quad (2.20)$$

Da sich die Korrelation der turbulenten Fluktuationen nicht durch Mittelwerte ersetzen lässt (s. Gleichung 2.20), ist es erforderlich, diese zu kennen oder adäquat zu ersetzen. Diese Problematik wird als Schließungsproblem bezeichnet. Der heute gängige Ansatz zur Lösung dieses Schließungsproblems ist der Ersatz der Korrelation der Fluktuationen durch Turbulenzmodelle. Diesen Turbulenzmodellen liegt i. d. R. der Ansatz zu Grunde, dass der Transport von Masse, Energie und Impuls quer zu den Stromlinien durch die Viskosität gesteuert wird (FERZIGER, PERIĆ und STREET, 2020). Folglich kann angenommen werden, dass der Einfluss der Turbulenz auf den Transportprozess durch eine künstliche, erhöhte Viskosität beschrieben werden kann. Diese erhöhte Viskosität wird turbulente Viskosität oder Wirbelviskosität genannt (Anm.: $\nu_t = \mu_t/\rho$):

$$\tau_{ij} = \rho \nu \frac{\partial \bar{u}_i}{\partial x_j} - \overline{u'_j u'_i} := \rho \nu_t \frac{\partial \bar{u}_i}{\partial x_j}. \quad (2.21)$$

Ein Korrekturterm in Gleichung 2.21 garantiert, dass die Kontinuitätsgleichung und turbulente kinetische Energie ($k = \frac{1}{2} \overline{u'_i u'_i}$) weiterhin erfüllt werden. Da der Term lediglich auf die Hauptspannungskomponenten anzuwenden ist, wirkt dieser nur auf der Spur des Spannungstensors (s. Kronecker-Delta in Gleichung 2.22). In der Literatur wird die resultierende Gleichung 2.22 oft erweitertes Wirbelviskositätsprinzip genannt:

$$-\overline{u'_j u'_i} = \nu_t \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} k. \quad (2.22)$$

Die Wirbelviskositätsmodelle liefern ein Ersatzmodell für die turbulente Viskosität in der Gleichung und lösen so das Schließungsproblem. Dabei gibt es unterschiedliche, oft semiempirische, Ansätze.

Die **Prandtl'sche Mischungsweghypothese** ist ein Null-Gleichungsmodell, dem die Annahme zu Grunde liegt, dass die Wirbelviskosität als Produkt einer charakteristischen Länge l_m (Mischungsweg) und charakteristischen Geschwindigkeit ausgedrückt werden kann:

$$\nu_t = l_m^2 \left\| \frac{\partial \bar{u}_i}{\partial x_j} \right\| = \sqrt{k} l_m = C_\mu \frac{k^2}{\varepsilon} . \quad (2.23)$$

Die Bestimmung des Mischungsweges kann sich je nach Modell und Anwendung unterscheiden, da diese maximal die Dicke der Scherschicht annehmen kann. VAN DRIEST (1956) entwickelte die Methode weiter, um zu berücksichtigen, dass die Viskosität in der viskosen Unterschicht die Wirbel dämpft und ihre Größe verringert:

$$l_m = \kappa d \left[1 - \exp\left(-\frac{y^+}{C_{A^*}}\right) \right] \quad C_{A^*} = 26,0 , \quad (2.24)$$

wobei κ die von Kármán Konstante, d der Wandabstand, y^+ der dimensionslose Wandabstand und C_{A^*} eine Konstante ist.

Das **k- ε Modell** gehört zu den Zweigleichungsmodellen. Hierbei wird angenommen, dass sowohl die kinetische Energie k im Berechnungsgebiet transportiert wird:

$$\underbrace{\frac{\partial k}{\partial t}}_{\text{Zeitliche Veränderung}} + \underbrace{\rho(\mathbf{u} \cdot \nabla)k}_{\text{Konvektion}} = \underbrace{\nabla \cdot \left[\left(\mu + \frac{\mu_t}{C_{\sigma_k}} \right) \nabla k \right]}_{\text{Diffusion}} + \underbrace{q_k + \rho\varepsilon}_{\text{Quellen und Senken}} , \quad (2.25)$$

als auch die Dissipationsrate ε :

$$\frac{\partial \varepsilon}{\partial t} + \rho(\mathbf{u} \cdot \nabla)\varepsilon = \nabla \cdot \left[\left(\mu + \frac{\mu_t}{C_{\sigma_\varepsilon}} \right) \nabla \varepsilon \right] + C_{\varepsilon 1} \frac{\varepsilon}{k} q_k + C_{\varepsilon 2} \rho \frac{\varepsilon^2}{k} , \quad (2.26)$$

wobei $C_{\sigma_k} = 1.0$, $C_{\sigma_\varepsilon} = 1.3$, $C_{\varepsilon 1} = 1.44$ und $C_{\varepsilon 2} = 1.92$ modellspezifische Konstanten sind (LAUNDER und SPALDING, 1974). Aus der Lösung der zusätzlichen Transportgleichungen ergibt sich die Wirbelviskosität:

$$\nu_t = C_\mu \frac{k^2}{\varepsilon} . \quad (2.27)$$

Beim **k- ω Modell** handelt es sich ebenfalls um ein Zweigleichungsmodell. Unter anderem WILCOX und ALBER (1972) entwickelten das k - ω Modell, um die Ersatz-Modellierung der Turbulenz für Grenzschichten mit ungünstigen Druckgradienten zu optimieren. Erneut ist die Transportgleichung der kinetischen Energie zu lösen, allerdings wird $+\rho\varepsilon$, entgegen Gl. 2.25,

nun als Senke angenommen ($-\rho\varepsilon$). Anstelle der Dissipationsrate ε ist für das k - ω Modell eine Transportgleichung der spezifischen Dissipationsrate ω zu lösen:

$$\frac{\partial \omega}{\partial t} + \rho(\mathbf{u} \cdot \nabla)\omega = \nabla \cdot \left[\left(\mu + \frac{\mu_t}{C_{\sigma_k}} \right) \nabla \omega \right] + \frac{C_\gamma}{\nu_t} q_k - C_\beta \rho \omega^2 . \quad (2.28)$$

Mögliche Werte der Koeffizienten C_{σ_k} , C_γ und C_β können z. B. WILCOX (2004) entnommen werden.

Die Ähnlichkeit der vorgestellten Zweigleichungsmodelle lässt sich mathematisch ausdrücken, da diese ineinander überführt werden können:

$$\omega = \frac{\varepsilon}{C_\mu k} , \quad \text{mit } C_\mu = 0,09 . \quad (2.29)$$

2.2.4 Verifizierung und Validierung numerischer Modelle

Der Statistiker BOX (1976) hielt fest, dass alle Modelle falsch, aber manche nützlich seien. Die Verifizierung und Validierung helfen dabei, die Nützlichkeit eines Modells objektiv einzuschätzen. Im Kontext numerischer Strömungssimulationen adressieren die Validierung und Verifizierung dafür zwei unterschiedliche Fragestellungen, die darauf abzielen, die Genauigkeit des Simulationsmodells zu quantifizieren. Die Validierung umfasst den Abgleich des Simulationsmodells mit der Realität, während die Verifizierung die korrekte Lösung der zugrunde liegenden Gleichungen überprüft (s. Abbildung 2.7). Die Genauigkeit setzt sich

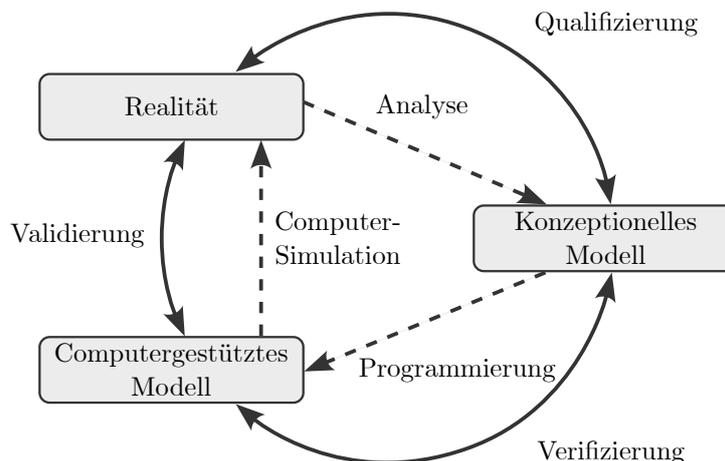


Abbildung 2.7: Prozessdarstellung der Verifizierung und Validierung im Kontext computergestützter Simulationen. Eigene Darstellung, in Anlehnung an SCS (1979) aus dem engl.

aus der Bewertung von Fehlern und den Unsicherheiten zusammen. Fehler lassen sich als die Differenz zwischen einem gemessenen und einem wahren Wert bestimmen, wohingegen Unsicherheiten dazu dienen, die Streuung um einen gemessenen Wert zu charakterisieren

und daher auf stochastische Prozesse zurückzuführen sind. Fehler und Unsicherheiten können sowohl bei der Verifizierung als auch der Validierung eine Rolle spielen, weshalb die Grundlagen zur Quantifizierung in den folgenden Unterkapiteln dokumentiert werden.

Verifizierung

Die Verifizierung numerischer Simulationen hat vor allem die Schwerpunkte, den iterativen Fehler und den Diskretisierungsfehler sowie die Unsicherheiten zu quantifizieren.

Modellspezifische Fehler aus der iterativen numerischen Lösung der diskretisierten Gleichungen können lediglich abgeschätzt werden (ASME, 2009). Für den iterativen Fehler gibt es i. d. R. keinen spezifischen Wert, den es zu erreichen gilt, sondern lediglich Richtlinien. EÇA und HOEKSTRA (2002) empfehlen, dass dieser gegenüber dem Diskretisierungsfehler um zwei bis drei Größenordnungen kleiner sein sollte. Für die Um- und Durchströmung von Gebäuden empfehlen TOMINAGA et al. (2008) und FRANKE und BAKLANOV (2007), die relevanten Größen über die Iterationen aufzutragen und den stationären Zustand zu belegen. FRANKE (2006) empfiehlt konkreter, dass die normierten Residuen um mindestens vier Größenordnungen fallen müssen und ein Residuum von $1 \cdot 10^{-3}$ als nicht mehr ausreichend zu bewerten ist.

Den Diskretisierungsfehler und die damit assoziierten Unsicherheiten zu bestimmen, ist dagegen komplexer. Zur Fehlerabschätzung wird am häufigsten die von ROACHE (1998) entwickelte Methode der Richardson-Extrapolation verwendet (ASME, 2009), die in Abbildung 2.8 visualisiert ist. Die Richardson-Extrapolation nutzt aus, dass eine unendlich feine Diskretisierung der stetigen Gleichungen zur exakten Lösung führt. Mit mindestens drei unterschiedlich feinen Netzen wird eine Funktion an die Datenpunkte der Zielgröße gefittet und das extrapolierte Ergebnis für ein theoretisch unendlich feines Netz berechnet. Eine ideale CFD konvergiert bei Verwendung der FEM mit linearer Ansatzfunktion maximal mit zweiter Ordnung genau. In der Praxis ist die Konvergenzrate jedoch meist geringer, weshalb anhand der unterschiedlichen Netzfeinheiten die modellspezifische Konvergenzrate p_{num} berechnet wird:

$$p_{\text{num}} = \left[\frac{1}{\ln(r_{21})} \right] \left[\ln \left| \frac{\epsilon_{32}}{\epsilon_{21}} \right| + f(p_{\text{num}}) \right], \quad (2.30)$$

$$f(p_{\text{num}}) = \ln \left(\frac{r_{21}^{p_{\text{num}}} - s}{r_{32}^{p_{\text{num}}} - s} \right), \quad (2.31)$$

$$s = \text{sign} \left(\frac{\epsilon_{32}}{\epsilon_{21}} \right), \quad (2.32)$$

wobei $r \geq 1,3$ die Netzverfeinerungsrate und ϵ die Differenz der Zielgrößen ist. Das Gleichungssystem lässt sich z. B. mit dem Bisektionsverfahren lösen.

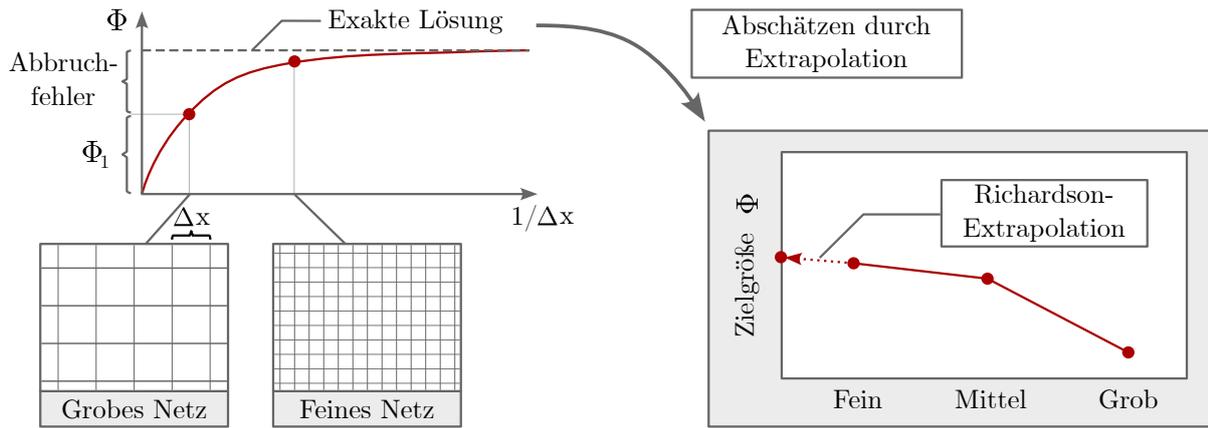


Abbildung 2.8: Schematische Darstellung des Diskretisierungsfehlers (links) und der Richardson-Extrapolation (rechts) (eigene Darstellung)

Das mittels Potenzfunktion extrapolierte Ergebnis Φ_{ext} liefert die Abschätzung des Zielwertes auf einem unendlich feinen Rechenetz für die modellspezifische Konvergenzrate p_{num} :

$$\Phi_{\text{ext}} = \frac{r_{21}^{p_{\text{num}}} \Phi_1 - \Phi_2}{r_{21}^{p_{\text{num}}} - 1}. \quad (2.33)$$

Die Unsicherheit der Methode kann quantifiziert werden, indem ein *Grid Convergence Index* (GCI) ermittelt wird:

$$\text{GCI}_{\text{fein}}^{21} = \frac{F_S \cdot e_R^{21}}{r_{21}^{p_{\text{num}}} - 1}. \quad (2.34)$$

Der Sicherheitsfaktor F_S ist für eine Untersuchung mit drei oder mehr Netzen konservativ mit $F_S = 1,25$ anzunehmen (ROACHE, 1998). Anhand empirischer Untersuchungen berichten COLEMAN und STERN (1997) sowie STERN et al. (2001), dass die berechnete Unsicherheit damit in einem 95 % Intervall liegt. Diese Unsicherheit unterliegt allerdings keinen statistischen Annahmen, wie einer Gaußverteilung. Für die weitere Verwendung kann daher die Umrechnung in die Form einer Standardabweichung σ erforderlich sein:

$$u_{\text{num}} = \frac{U_{\text{num}}}{C_R} = \frac{\text{GCI}_{\text{fein}}^{21}}{C_R}. \quad (2.35)$$

Der Koeffizient C_R hängt vom Konvergenzverhalten der Lösungswerte ab. Bei oszillierender Konvergenz gilt $C_R = 2,00$, bei monotoner Konvergenz gilt $C_R = 1,10$ bis $1,15$. CELIK et al. (2008) beschreiben, wie sich anhand der Zielgrößen der unterschiedlich feinen Rechenetze der Konvergenzmodus bestimmen lässt.

Validierung

Die zentrale Aufgabe der Validierung ist es, den Abgleich von Simulation und Realität zu

quantifizieren. Dabei stützt sich der Prozess maßgeblich auf Messergebnisse. Der Abgleich besteht wiederum aus der Bestimmung von Fehlern und Unsicherheiten.

OBERKAMPF und TRUCANO (2000) entwickelten für die Quantifizierung der allgemeinen Übereinstimmung eine Validierungsmetrik VM:

$$VM = 1 - \frac{1}{N} \sum_{i=1}^N \tanh \left| \frac{u_{\text{ref}}(x_i) - \tilde{u}(x_i)}{\tilde{u}(x_i)} \right|, \quad (2.36)$$

wobei $u_{\text{ref}}(x_i)$ der beobachtete Wert und $\tilde{u}(x_i)$ der Vorhersagewert durch die Simulation ist.

Spezifischer ist die Analyse bestehender Metriken von CHANG und HANNA (2004). Für die Bewertung systematischer Fehler wird der *geometric mean bias* (MG) und *fractional bias* (FB) empfohlen. Unsystematische Fehler können mittels *normalized mean square error* (NMSE) und *geometric variance* (VG) identifiziert werden. Der *factor of two* (FAC2) quantifiziert große Abweichungen zwischen Modell und Realität. Grenzwerte der Metriken können bspw. PATRYL und GALERIU (2011) entnommen werden:

$$MG = \exp \left[\overline{\ln(u_{\text{ref}})} - \overline{\ln(\tilde{u})} \right], \quad 0,7 < MG < 1,3, \quad (2.37)$$

$$FB = \frac{\left(\overline{u_{\text{ref}}} - \overline{\tilde{u}} \right)}{0,5 \left(\overline{u_{\text{ref}}} + \overline{\tilde{u}} \right)}, \quad |FB| < 0,3, \quad (2.38)$$

$$NMSE = \frac{\overline{(u_{\text{ref}} - \tilde{u})^2}}{\overline{u_{\text{ref}}} \overline{\tilde{u}}}, \quad NMSE < 1,5, \quad (2.39)$$

$$VG = \exp \left[\overline{(\ln(u_{\text{ref}}) - \ln(\tilde{u}))^2} \right], \quad VG < 4,0, \quad (2.40)$$

$$FAC2 = \text{Teil der Daten die } 0,5 \leq \frac{\tilde{u}}{u_{\text{ref}}} \leq 2,0 \text{ erfüllen,} \quad FAC2 > 0,5. \quad (2.41)$$

CHANG und HANNA (2004) bewerten den Korrelationskoeffizienten als Bewertungsmaßstab für ungeeignet, da dieser einen linearen Zusammenhang impliziert und gegenüber wenigen großen abweichenden Wertepaaren empfindlich ist. Dennoch lässt sich der Korrelationskoeffizient als Validierungsmetrik in vielen Arbeiten für die Simulation der Durch- oder Umströmung von Gebäuden finden (BUSTAMANTE et al., 2013; BLANES-VIDAL et al., 2008; YOSHIE, MOCHIDA, TOMINAGA et al., 2007).

Zusätzlich ergeben sich Unsicherheiten aus den Modellparametern. Hierzu zählen die dem Modell aufgeprägten Randbedingungen oder die Variation geometrischer Details. Anhand

einer Sensitivitätsanalyse kann die Abweichung von minimaler und maximaler Parametereinstellung ermittelt werden. Ebenso gilt es geeignete Turbulenzmodelle mit den experimentellen Ergebnissen zu vergleichen (s. Abbildung 2.9), da z. B. die Parameter der in Kapitel 2.2.3 benannten Turbulenzmodelle für ein möglichst breites Anwendungsspektrum entwickelt wurden und daher keine eindeutige anwendungsspezifische Zuordnung der Modelle erfolgt.

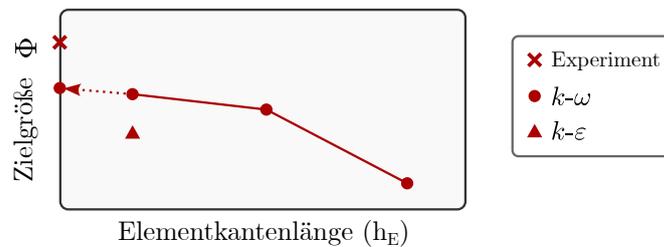


Abbildung 2.9: Darstellung der grafischen Aufbereitung einer Sensitivitätsanalyse des Turbulenzmodells (eigene Darstellung)

Der Übersichtsarbeit von BLOCKEN et al. (2011) können weiterführende Informationen zum Stand der Validierung von Simulationen im Kontext von Gebäuden entnommen werden.

2.3 Grundlagen der statistischen Versuchsplanung

Um das digitale Modell für eine Optimierung des physischen Objektes einzusetzen, gilt es, unterschiedliche Randbedingungen zu prüfen, wofür eine Vielzahl numerischer Simulationen erforderlich wird. Das Ziel der statistischen Versuchsplanung (engl. *Design of experiments* (DoE)) besteht darin, diesen experimentellen Aufwand zu minimieren und dennoch eine aussagekräftige und mathematisch begründete Suche nach dem Parametereinfluss oder Optimum im Untersuchungsraum durchzuführen. Hierzu zählt insbesondere auch der effiziente Entwurf numerischer Experimente, um den nach heutiger Rechenleistung hohen numerischen Aufwand zu reduzieren. Dazu werden zunächst die Eingangsgrößen des Systems identifiziert, welche in der statistischen Versuchsplanung als Parameter bezeichnet werden. Die Teilmenge der zu untersuchenden Parameter wird hingegen als Faktor bezeichnet. Im Rahmen der Anwendung auf numerische Simulationen ist die erforderliche gezielte und reproduzierbare Einstellung der Faktoren unproblematisch. Die Einstellungen der Faktoren werden in Stufen unterteilt. Wenn das Ergebnis des Versuchsplans lineare Zusammenhänge aufdecken soll, reichen zwei Stufen je Faktor aus. Für quadratische Zusammenhänge sind je Faktor drei Stufen im Versuchsplan erforderlich, wobei SIEBERTZ, BEBBER und HOCHKIRCHEN (2017) festhalten, dass die Nichtlinearität der Zusammenhänge oft überschätzt wird. Die erhöhte Anzahl der Stufen und das Aufdecken der quadratischen Zusammenhänge bedeuten eine höhere Anzahl an Durchläufen im Versuchsplan, wobei nach SIEBERTZ, BEBBER und HOCHKIRCHEN (2017) mit einem geringen Stufenabstand und linearen Modellen oft gute Ergebnisse

zu erzielen sind. Die Wechselwirkungen der Faktoren werden nach SIEBERTZ, BEBBER und HOCHKIRCHEN (2017) hingegen oft unterschätzt. Der Aufwand für einen vollfaktoriellen Versuchsplan lässt sich wie folgt berechnen:

$$N = n^i, \tag{2.42}$$

wobei N die Anzahl der Versuchsdurchläufe, n die Anzahl der Stufen je Faktor und i die Anzahl der zu untersuchenden Faktoren darstellt.

Ein vollfaktorieller Versuchsplan für bspw. 14 Faktoren auf zwei Stufen führt somit zu 16.384 Versuchsdurchläufen. Die statistische Versuchsplanung reduziert diesen Aufwand effektiv, indem orthogonale und ausgewogene Versuchspläne generiert werden, welche sicherstellen, dass keine der Kombinationen aus jeweils zwei Spalten miteinander korrelieren und die Einstellung der Stufen jedes Faktors gleich oft vorkommt. Diese Vorgehensweise ermöglicht es, trotz gleichzeitiger Variation der Faktoren, eine eindeutige Zuordnung der Effekte vorzunehmen. Ein weiterer Vorteil liegt darin, dass die Umstellvorgänge nicht in einem Eckpunkt des Faktorraumes beginnen, sondern diese gleichmäßig darin verteilt sind.

Abbildung 2.10 stellt linksseitig einen vollfaktoriellen, mittig einen teilfaktoriellen und rechts einen Box-Behnken-Versuchsplan dar.

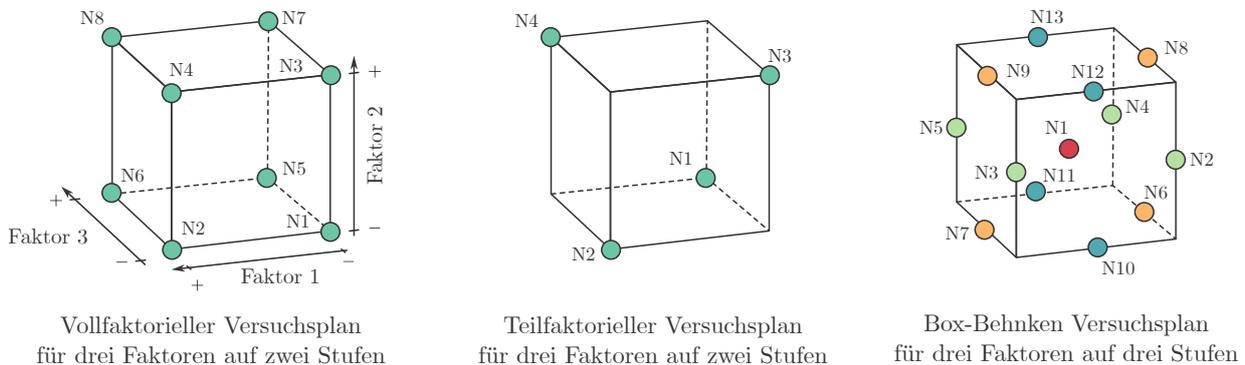


Abbildung 2.10: Darstellung des Faktorraumes eines vollfaktoriellen (links), teilfaktoriellen (mittig) und Wirkungsflächen-Versuchsplans nach Box-Behnken (rechts)

Der teilfaktorielle Versuchsplan zeigt die Reduktion der benötigten Versuchsdurchläufe von acht auf vier Durchläufe auf, wohingegen der Box-Behnken-Versuchsplan quadratische Zusammenhänge in dreizehn anstatt 27 Versuchsdurchläufen aufdeckt. Der zusätzliche Zentralpunkt im Box-Behnken-Versuchsplan ($N1$) kann auch in den (teil)faktoriellen Versuchsplänen implementiert werden, um die Nichtlinearität der Faktoren abzuschätzen.

Für die vorgeschaltete Identifikation der relevanten Parameter eignen sich dagegen *Screening*-Versuchspläne, welche (Mehrfach-)Wechselwirkungen mit den Haupteffekten zusammenlegen. Eine eindeutige Zuordnung von Haupteffekten und Wechselwirkungen ist hierbei erschwert. Die Einteilung der Versuchspläne in Qualitätsgruppen (auch Auflösung genannt)

erfolgt über die benannte Zusammenlegung von Wechselwirkungen und Haupteffekten (Tabelle 2.1).

Tabelle 2.1: Auflösung von Versuchsplänen nach SIEBERTZ, BEBBER und HOCHKIRCHEN (2017)

Auflösung	Beschreibung
III	Haupteffekte sind mit Zweifachwechselwirkungen vermengt. Das Feld ist dicht besetzt und nur zum Screening geeignet.
IV	Haupteffekte sind mit Dreifachwechselwirkungen vermengt und Zweifachwechselwirkungen untereinander. Das Feld ist geeignet, um Haupteffekte sicher zu bestimmen, Zweifachwechselwirkungen lassen sich nicht eindeutig zuordnen.
V	Haupteffekte sind mit Vierfachwechselwirkungen vermengt und Dreifachwechselwirkungen mit Zweifachwechselwirkungen.
V+	Haupteffekte und Zweifachwechselwirkungen sind praktisch unvermengt.

Ein bis heute weitverbreiteter Ansatz für Screening-Versuchspläne geht auf PLACKETT und BURMAN (1946) zurück. Bei den irregulären Plackett-Burman-Versuchsplänen ist das Produkt zweier Spalten nur zu diesen Spalten orthogonal, nicht aber zum Rest des Feldes. Daraus ergibt sich der Vorteil, dass die Zweifachwechselwirkung dieser Spalten nicht auf die Haupteffekte selbst wirkt, allerdings wirkt diese Zweifachwechselwirkung abgeschwächt auf die anderen Haupteffekte (SIEBERTZ, BEBBER und HOCHKIRCHEN, 2017). Ein solcher Versuchsplan lässt sich maximal mit $N - 1$ Faktoren belegen. Sollte die Auflösungsstufe III nicht ausreichen, lässt sich das Feld mittels Falten und Invertieren in eine Auflösungsstufe IV überführen. Daraus wird auch ersichtlich, dass die bisherigen Versuchsreihen weiterhin Anwendung finden, der Aufwand ergänzt sich lediglich um die gefalteten und invertierten Durchläufe.

Aus der Auswertung des Versuchsplans hinsichtlich einer Zielgröße ergeben sich die Effekte der Faktoren. Der Effekt berechnet sich für lineare Zusammenhänge aus der Differenz der Mittelwerte bei den hohen und niedrigen Einstellungen, welche mit + und – kodiert werden:

$$E_P = \frac{1}{N} \sum^N u_+ - \frac{1}{N} \sum^N u_- , \quad (2.43)$$

wobei E_P die Effektstärke, u_+ die hohe und u_- die niedrige Faktoreinstellung kennzeichnet.

Aus den Effektstärken der Faktoren kann abschließend eine Regressionsgleichung gebildet werden, um für eine beliebige Einstellung der Faktoren eine zahlenwertige Antwort zur Zielgröße zu erhalten.

2.4 Grundlagen (Physik-informierter) künstlicher neuronaler Netze

Im Gegensatz zu Regressionsgleichungen haben künstliche neuronale Netze die Fähigkeit, sich eigenständig an die gegebenen Datenpunkte anzupassen und sogar Grundfunktionen höherer Ordnung zu bilden oder das gesamte Lösungsfeld zu visualisieren. Künstliche neuronale Netze setzen dies über die Adaption eines vereinfachten biologischen Aufbaus der Neuronen um und gehen auf die Ideen von MCCULLOCH und PITTS (1943) zurück. Beim biologischen Vorbild werden die im synaptischen Spalt verstärkten oder gehemmten Signale mittels Dendriten an den Nervenzellkörper geleitet. Die weitergeleiteten Signale summieren sich am Axonhügel (zwischen Soma und Axon). Wird ein Schwellwert erreicht, erfolgt die Signalweiterleitung über die Axonterminale (s. Abbildung 2.11, links). Ein künstliches Neuron besteht aus den Eingangssignalen (x_i), die jeweils gewichtet werden (w_i), einer Summierung der gewichteten Eingänge und anschließender Schwellwertfunktion. Die Schwellwertfunktion entscheidet darüber, wie hoch das Aktivitätslevel (a) des Neurons ist und welcher Ausgangswert an das folgende künstliche Neuron weitergeleitet wird (s. Abbildung 2.11, rechts). Im Folgenden wird auf die zusätzliche Kennzeichnung „künstlich“ verzichtet, da diese Arbeit ausschließlich künstliche neuronale Netze behandelt.

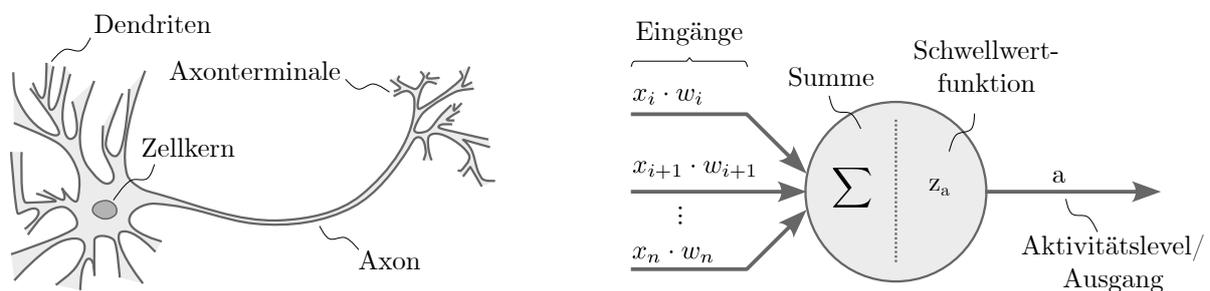


Abbildung 2.11: Vereinfachte Darstellung eines biologischen Neurons links und die Darstellung der künstlichen Adaption rechts (eigene Darstellung)

2.4.1 Vom Neuron zum neuronalen Netz

Die **Aktivierungsfunktionen** typisieren das Aktivitätslevel des einzelnen Neurons, indem z. B. lineare (ggf. mit Schwellwert), binäre, sigmoidale oder normalverteilte Aktivierungsfunktionen eingesetzt werden. Häufig wird eine Art der Aktivierungsfunktion auf das gesamte Netz angewandt, obwohl sich unterschiedliche Arten je Schicht oder gar Neuron einsetzen ließen. Am häufigsten werden sigmoidale Aktivierungsfunktionen eingesetzt (REY und WENDER, 2018). Diese können wiederum in logistische Funktionen und den Tangens hyperbolicus (tanh) unterteilt werden. Logistische Funktionen nehmen Werte von 0 bis +1 an, wohingegen der tanh von -1 bis +1 reicht. Der Funktionsverlauf entspricht keinem Sprung und

ist damit biologisch plausibel sowie differenzierbar. Die Differenzierbarkeit der Aktivierungsfunktion spielt für die spätere Optimierung eine Rolle.

Ergänzend kann dem Neuron eine Grundaktivität zugewiesen werden, indem ein Bias hinzugefügt wird:

$$a_i = z_a(a_j \cdot w_{ij} + b_i), \tag{2.44}$$

wobei a_i das Aktivitätslevel des empfangenden Neurons ist, z_a die (sigmoidale) Aktivierungsfunktion, a_j das empfangene Ausgangssignal des sendenden Neurons, b_i der Bias des empfangenden Neurons und w_{ij} die Gewichtung zwischen dem empfangenden und sendenden Neuron ist.

Ein einzelnes Neuron kann lediglich einfache boolesche Operationen abbilden. Das Theorem von PINKUS (1999) besagt aber, dass ein neuronales Netz mit nur einer verdeckten Schicht, ausreichend vielen Neuronen sowie nicht linearer Aktivierungsfunktion jede beliebig komplexe Funktion und deren Ableitungen approximieren kann. Dazu müssen neuronale Netze mit mindestens einer verdeckten Schicht gebildet werden, indem die in Abbildung 2.11 (rechts) dargestellten Neuronen vervielfältigt und miteinander verknüpft werden. Abbildung 2.12 stellt ein vollständig verknüpftes Netz dar.

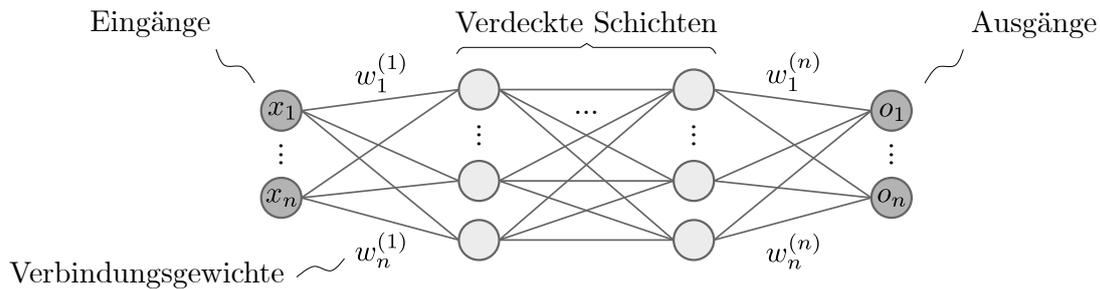


Abbildung 2.12: Aufbau eines vollständig verknüpften künstlichen neuronalen Netzes (eigene Darstellung)

Die entstehende **Netztopologie** kann sich unterscheiden, wobei vollständig miteinander verknüpfte Netze (engl. *fully connected neural network*) die Basis bilden. Das Netz lässt sich in Teilbereiche gliedern und besteht aus der Eingangsschicht, den verdeckten Schichten und der Ausgangsschicht. Jede der Schichten kann eine unterschiedliche Anzahl von Neuronen beinhalten.

2.4.2 Berechnung, Fehlerbestimmung und Optimierung

Analog zu Gleichung 2.44 lässt sich die Matrixschreibweise für ein neuronales Netz mit mehreren Neuronen auf mehreren Schichten aufstellen:

$$\mathbf{a}_k = \sigma(\mathbf{W}_{k-1}^T \mathbf{a}_{k-1} + \mathbf{b}_{k-1}), \tag{2.45}$$

wobei \mathbf{a}_k der Aktivitäts-Vektor der Schicht k , \mathbf{W}_{k-1}^T die transponierte Matrix der Gewichte der vorangehenden Schicht $k - 1$, \mathbf{a}_{k-1} das Aktivitätslevel der vorangehenden Schicht und \mathbf{b}_{k-1} der Bias-Vektor der vorangehenden Schicht ist. Die Berechnung bis zur Ausgangsschicht eines neuronalen Netzes gemäß Gleichung 2.45 wird als **feedforward pass** bezeichnet. Die Matrixschreibweise zeigt, dass eine stark parallelisierte Verarbeitung mittels Computer umsetzbar ist.

Inwiefern die Ausgaben aus dem *feedforward pass* mit dem wahren Lösungswert übereinstimmen, kann anhand unterschiedlicher Metriken quantifiziert werden. Am gebräuchlichsten ist der *mean squared error* (MSE):

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^N |u_{\text{ref}i} - \tilde{u}_i|^2, \quad (2.46)$$

wobei u_{ref} der erwartete und \tilde{u} der vorhergesagte Wert ist.

Ziel ist es, die Verbindungsgewichte und Biases des neuronalen Netzes so einzustellen, dass die Fehlermetriken möglichst gering werden. Die Werte sind jedoch nicht a priori bekannt, weshalb diese mittels iterativen Optimierungsalgorithmen angepasst werden müssen. Die Kombination vieler Parameter und nicht linearer Funktionen zeigt auch auf, dass die resultierende zu optimierende Funktion $f(\Theta)$ i. d. R. nicht strikt konvex ist. Das Gros der Frameworks implementiert gradientenbasierte Optimierer für diese Aufgabe, welche die Parameter Θ entgegen dem Gradienten $\nabla f(\Theta)$ anpassen (s. Abbildung 2.13).

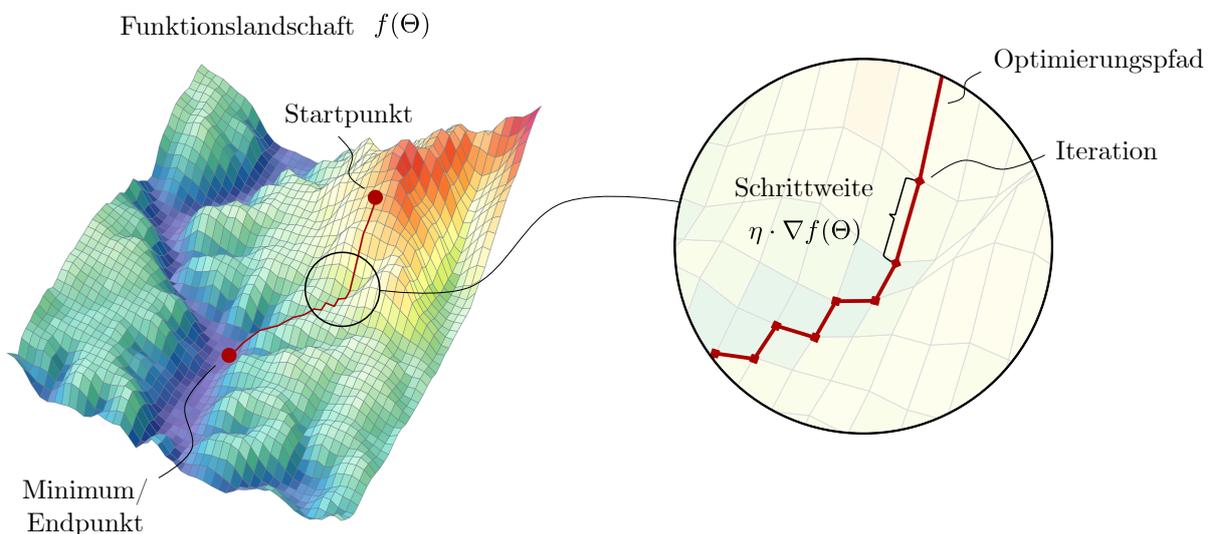


Abbildung 2.13: Gradientenbasierte Optimierung der Parameter Θ einer Funktion $f(\Theta)$. Eigene Darstellung, unter Verwendung von MATPLOTLIB (HUNTER, 2007)

Gradientenbasierte Algorithmen berechnen die aktualisierten Parameter Θ_i mittels Lernrate η , welche die Schrittweite beschreibt, mit der entgegen der Richtung des Gradienten „gegangen“ wird, um die Parameter für die nächste Iteration zu berechnen:

$$\Theta_i = \Theta_{i-1} - \eta \cdot \nabla f(\Theta) . \quad (2.47)$$

Die Abwägung der Größe des Datensets, nach dem eine Iteration des Optimierers durchgeführt wird, ist für die Optimierung entscheidend: Die Optimierung über das gesamte Set ist langsam, wohingegen stochastische Optimierer (Iteration nach jedem Datensatz) zu Fluktuationen neigen. Diese Oszillation im Optimierungsprozess zeigt sich aufgrund der starken Krümmung der Oberflächen in unterschiedlichen Dimensionen (SUTTON, 1986). Eine Möglichkeit dem zu begegnen ist die Unterteilung des Datensatzes in *mini-batches*, um die Vorteile beider Verfahren zu kombinieren.

Die Methode des Momentums dämpft diese Oszillation, indem ein Teil (Faktor C_β) des vorigen Optimierungsvektors \mathbf{o}_{i-1} in die aktuelle Iteration übertragen wird (QIAN, 1999):

$$\begin{aligned} \mathbf{o}_i &= C_\beta \mathbf{o}_{i-1} + \eta \cdot \nabla f(\Theta) \\ \Theta_i &= \Theta_{i-1} - \mathbf{o}_i . \end{aligned} \quad (2.48)$$

Ergänzend sei hier auf das von NESTEROV (1983) entwickelte *Nesterov accelerated gradient* verwiesen, wobei die Ergebnisse des Verfahrens bei kleinen Schrittweiten mit den Ergebnissen der Momentum-Methode übereinstimmen.

Eine auf diesen Methoden aufbauende und oft verwendete Variante ist die *Adaptive Moment Estimation* (Adam) (KINGMA und BA, 2015): Hierbei wird zusätzlich ein exponentiell abklingender Durchschnitt der (quadrierten) Gradienten der vorangegangenen Iterationen einbezogen und sowohl das erste, als auch das zweite Momentum abgeschätzt ($m_i^{(1)}$, $m_i^{(2)}$). Eine Korrektur des Momentums erster und zweiter Ordnung minimiert die anfängliche Tendenz des Optimierers, Werte nahe null einzustellen ($\hat{m}_i^{(1)}$, $\hat{m}_i^{(2)}$):

$$\begin{aligned} m_i^{(1)} &= C_{\beta_1} m_{i-1}^{(1)} + (1 - C_{\beta_1}) \nabla_i ; & \hat{m}_i^{(1)} &= \frac{m_i^{(1)}}{1 - C_{\beta_1}^i} , \\ m_i^{(2)} &= C_{\beta_2} m_{i-1}^{(2)} + (1 - C_{\beta_2}) \nabla_i^2 ; & \hat{m}_i^{(2)} &= \frac{m_i^{(2)}}{1 - C_{\beta_2}^i} , \\ \Theta_{i+1} &= \Theta_i - \frac{\eta}{\sqrt{\hat{m}_i^{(2)} + \epsilon}} \hat{m}_i^{(1)} . \end{aligned} \quad (2.49)$$

Die Bildung der für die Optimierung erforderlichen Gradienten ist eine Herausforderung bei einem neuronalen Netz mit verdeckten Schichten und mehreren Neuronen. Es benötigt dazu eine spezialisierte Form der automatischen Differentiation, die im Kontext neuronaler Netze als **backpropagation** bezeichnet wird. Hierbei wird die Kettenregel angewandt, um den Einfluss der einzelnen Gewichte und Biases auf den Fehler aus dem *feedforward pass*

rückwärts gerichtet durch das Netz zu propagieren. Abbildung 2.14 visualisiert den Ablauf der Systematik.

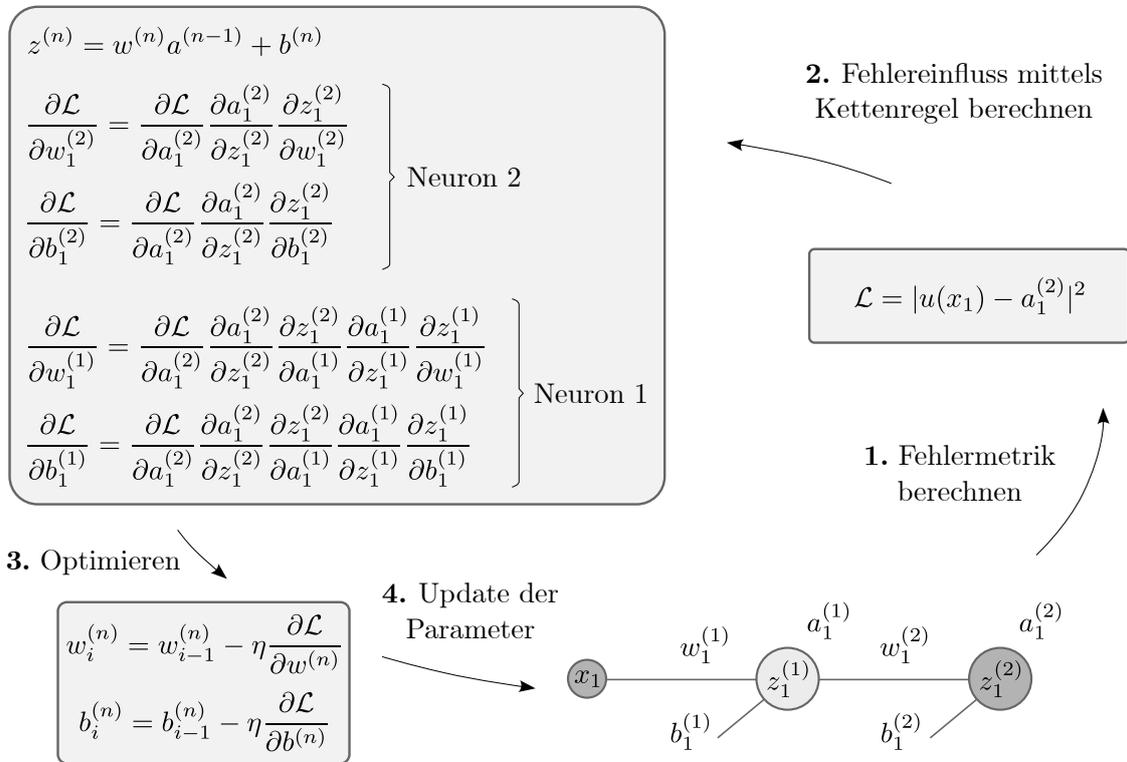


Abbildung 2.14: Schema der *backpropagation* am Beispiel eines neuronalen Netzes mit einer verdeckten Schicht und je einem Neuron pro Schicht sowie Aktualisierung der Parameter mittels Gradientenabstieg (eigene Darstellung)

Bei verdeckten Schichten mit mehreren Neuronen wird die Berechnung im Computer nur unwesentlich komplexer. Hierfür ergibt sich je Neuron auf der Schicht $n - 1$ eine Summierung über die gesamte folgende Schicht n , wobei die Anwendung der o. g. Systematik je Term erfolgt. Die benötigten Gradienten werden daher bereits im *feedforward pass* berechnet und die jeweiligen Terme (Ketten) können über eine Matrizenmultiplikation effizient bestimmt werden. Diese Aktualisierung wird bis zum Erreichen des Abbruchkriteriums iterativ durchgeführt.

2.4.3 Physik-informierte neuronale Netze

Das zuvor vorgestellte Verfahren des maschinellen Lernens mittels neuronalen Netzen birgt das Problem, dass zur Anpassung der Gewichte und Biases eine umfangreiche Datenbasis vorhanden sein muss, um die Fehlermetriken zu berechnen und die *backpropagation* durchzuführen. Bei realen ingenieurstechnischen Anwendungen müssen dazu viele aufwändige Experimente oder Simulationen durchgeführt werden. Physik-informierte neuronale Netze

(PINNs) umgehen dieses Problem, indem eine zusammengesetzte Fehlermetrik \mathcal{L} entwickelt wird, die um einen physikalischen Verlustterm ergänzt ist. Der physikalische Verlustterm inkorporiert die Randbedingungen (RB) und die das System beschreibenden partiellen Differentialgleichungen (PDEs) als Residuum. Durch die Einbeziehung der zusammengesetzten Verlustfunktion in das Training von PINNs ist das resultierende neuronale Netz nicht nur in der Lage, die Lösung der PDE vorherzusagen, sondern auch die zugrunde liegenden physikalischen Prinzipien einzuhalten:

$$\mathcal{L} = \mathcal{L}_d + \mathcal{L}_f + \mathcal{L}_b, \quad (2.50)$$

mit dem Verlustterm der Daten \mathcal{L}_d , der PDE \mathcal{L}_f und den Randbedingungen \mathcal{L}_b . Bei Anwendung einer MSE-Metrik ergeben sich die Terme zu:

$$\mathcal{L}_d = \frac{1}{N_u} \sum_{i=1}^{N_u} |u_{\text{ref},d}(t_i, x_i) - \tilde{u}_i|^2, \quad (2.51)$$

$$\mathcal{L}_f = \frac{1}{N_f} \sum_{i=1}^{N_f} |f(t_i, x_i)|^2, \quad (2.52)$$

und

$$\mathcal{L}_b = \frac{1}{N_b} \sum_{i=1}^{N_b} |u_{\text{ref},b}(t_i, x_i) - \tilde{u}_i|^2, \quad (2.53)$$

wobei N_d , N_f und N_b die Anzahl der Punkte sind, an denen Daten vorliegen, die PDE ausgewertet und Randbedingungen angewandt werden. Dieser Ansatz wurde, in spezieller Form, erstmals von PSICHOGIOS und UNGAR (1992) und verallgemeinerter von LAGARIS, LIKAS und FOTIADIS (1998) publiziert. Erst RAISSI, PERDIKARIS und KARNIADAKIS (2019) griffen die Methode erneut auf, um zeitabhängige und nicht lineare PDEs zu lösen.

Die Grundidee zur Bildung des physikalischen Verlustterms besteht darin, die in der PDE enthaltenen Gradienten mittels *backpropagation* durch das neuronale Netz zu approximieren. Dies geschieht, indem die Gewichte und Biases so angepasst werden, dass beispielsweise für einen Ausgabewert \tilde{u} des neuronalen Netzes in Bezug auf den Eingang x_i gilt:

$$\frac{\partial u_{\text{ref}}}{\partial x_i} \approx \frac{\partial \tilde{u}}{\partial a_n^{(n)}} \cdot \dots \cdot \frac{\partial z_1^{(1)}}{\partial x_i}. \quad (2.54)$$

Letztlich ist dies die Ergänzung des in Abbildung 2.14 dargestellten Schemas: Ausgehend vom Ausgang des neuronalen Netzes wird die Kettenregel bis zum Eingang x_i fortgeführt. Daraus geht außerdem hervor, dass auch Ableitungen höherer Ordnung durch die mehrfache Anwendung der Kettenregel approximiert werden können.

Die Verwendung kleinerer Datensätze bietet zusätzliche Möglichkeiten bei der Auswahl der Optimierungsverfahren. Waren zuvor Verfahren höherer Ordnung zu berechnungsintensiv, können diese für PINNs eingesetzt werden. Ein Beispiel ist das L-BFGS-Verfahren (D. C. LIU und NOCEDAL, 1989), bei dem die Ableitungen zweiter Ordnung approximiert werden. Zwar argumentiert LU et al. (2021), dass das L-BFGS-Verfahren eher zu lokalen Optima strebt, aber die Kombination klassischer Verfahren und folgender Optimierung mittels L-BFGS zu einer optimierten Parameteraktualisierung führt.

2.5 Stand der Forschung

Im Rahmen dieser Arbeit sollen die Grundlagen der numerischen Strömungssimulationen eingesetzt werden, um den Einfluss unterschiedlicher Faktoren auf das Tierwohl, die Emissionen und Belüftung des Stalls systematisch zu untersuchen. Das Kapitel 2.5.1 analysiert daher Arbeiten mit Bezug zur numerischen Strömungssimulation landwirtschaftlicher Gebäude. Mittels PINN sollen die Ergebnisse der Strömungssimulation in ein echtzeitfähiges Modell überführt werden, weshalb das Kapitel 2.5.2 die Literatur der PINNs dahingehend einordnet. Ein digitaler Zwilling soll dieses echtzeitfähige Modell implementieren und einen umfangreichen Datenaustausch aller Komponenten ermöglichen, weshalb das Kapitel 2.5.3 auf den Stand der Forschung zu digitalen Zwillingen in der Landwirtschaft eingeht.

2.5.1 Strömungssimulation landwirtschaftlicher Gebäude

Eine valide Simulation ermöglicht es, jeden Teilbereich detailliert zu visualisieren und damit ein tiefgreifendes Verständnis über die Strömungscharakteristik in einem Gebäude zu erlangen, das über die Erkenntnisse direkter Messergebnisse hinausgeht (BUSTAMANTE et al., 2013). Die Anwendung der CFD-Grundlagen auf die Strömungssimulationen landwirtschaftlicher Gebäude unterliegt jedoch speziellen Herausforderungen. Dies liegt zum einen an den von Stall zu Stall unterschiedlichen Lüftungselementen, zum anderen an der Anforderung, das Vieh in der Simulation adäquat abzubilden. Ergänzend ist der Emissions-, Luftfeuchte- und Wärmetransport von zentraler Bedeutung für das Tierwohl und den Emissionsausstoß. Damit diese abgebildet werden können, müssen geeignete Randbedingungen in der Simulation definiert und die zugehörigen Transportgleichungen in einer gekoppelten Berechnung gelöst werden.

Tabelle 2.2 kategorisiert die Analyse der mit dieser Arbeit thematisch verwandten Studien. Die Art der Simulation ist in eine Berücksichtigung von Fluid-, Wärme- und Stofftransport gegliedert. Die in der Simulation einbezogene Umgebung ist unterschiedlich detailliert ausgeführt. Einige Studien untersuchen nur bestimmte Segmente eines Stalls, während andere das Innere des Stalls und ggf. ergänzend die Umgebung betrachten. Es ist zudem eine Unterscheidung nach Einzelanalyse oder Parameterstudie aufgeführt. Die Abbildung des Viehs

in der Simulation teilt sich in die Darstellung über ein poröses Material, eine vereinfachte oder eine komplexe Geometrie auf. Abschließend folgt eine Einordnung der Validierung und Verifizierung.

Tabelle 2.2: Einordnung der Literatur zu numerischen Modellen von Stallungen

	Sim.			Umg.		Par.	Vieh			V&V	
	Fluidtransp.	Wärmetransp.	Stofftransp.	Segment Stall	Stall und Umg.	Einzelanalyse Parameterst.	Poröses Mat.	Simple Geom.	Komplexe Geom.	Verifizierung	Validierung
(NORTON, GRANT et al., 2009)	•	•			•	•				•	• ^{b,c}
(NORTON, GRANT et al., 2010)	•	•			•	• ^a		•		•	• ^b
(BJERG und ANDERSEN, 2010)	•	•	•	•		• ^a	•				
(RONG, NIELSEN und G. ZHANG, 2010)	•	•	•	•		• ^a				•	• ^b
(BUSTAMANTE et al., 2013)	•				•	•					•
(SAPOUNAS, DOOREN und SMITS, 2013)	•				•	• ^a					
(SAHA, FIEDLER, AMON et al., 2014)	•				•	• ^a				•	• ^b
(MONDACA und CHOI, 2016)	•	•		•		• ^a	•	•	•		•
(DREWRY, CHOI et al., 2018)	•	•	•	•		• ^a		•			•
(DREWRY, MONDACA et al., 2018)	•	•	•		•	• ^a	•			•	•
(X. WANG, G. ZHANG und CHOI, 2018)	•	•		•		• ^a			•	•	• ^c
(ZHOU et al., 2019)	•	•		•		• ^a			•	•	•
(TOMASELLO et al., 2019)	•				•	• ^a				•	•
(YAO et al., 2019)	•	•	•	•		• ^a			•	•	•
(MONDACA, CHOI und COOK, 2019)	•				•	• ^a	•		•	•	• ^b
(DAMASCENO et al., 2020)	•	•			•	•					• ^{b,c}
(FAGUNDES et al., 2020)	•	•			•	•				•	•
(JANKE, CAIAZZO et al., 2020)	•			•		• ^a				•	• ^b
(SAHA, YI et al., 2020)	•				•	• ^a				•	• ^b
(YI et al., 2020)	•				•	•				•	• ^b
(DOUMBIA, JANKE, YI, AMON et al., 2021)	•	•		•	•	• ^a	•	•		•	• ^b
(DOUMBIA, JANKE, YI, PRINZ et al., 2021)	•	•			•	•	•			•	• ^b
(VEGA et al., 2022)	•	•			•	•				•	•
DIESE ARBEIT	•	•	•		•	•		•		•	•

^aEs wurde keine statistische Versuchsplanung eingesetzt.

^bDie Validierung wurde mit einem maßstäblichen Modell im Windkanal durchgeführt.

^cDie Validierung wurde nicht für alle simulierten Quantitäten durchgeführt.

Die Einordnung der Literatur zeigt, dass in den meisten Arbeiten der Fokus auf dem Fluidtransport liegt und bei der Simulation von Wärme- sowie Stofftransport die Simulationsdomäne beschränkt wird. NORTON, GRANT et al. (2010) ergänzen den Wärmetransport und entwickelten ein nulldimensionales Wärmeübertragungsmodell für Kälber, welches die

konvektiven und radiativen Anteile lokal bestimmt. Es wird darauf hingewiesen, dass in windgetriebenen Umgebungen auf ein solches Modell verzichtet werden kann. BJERG und ANDERSEN (2010) und RONG, NIELSEN und G. ZHANG (2010) haben den Einfluss der Luftströmung auf den Stofftransport von Ammoniak innerhalb eines Stallsegments untersucht, um die Luftführung und Lüftungselemente dahin gehend zu optimieren. DREWRY, CHOI et al. (2018) haben den Fluid-, Wärme- und Stofftransport innerhalb eines Stallsegments untersucht und DREWRY, MONDACA et al. (2018) erweitern die Betrachtung auf das Innere eines ganzen Stalls. YAO et al. (2019) haben mithilfe einer Simulation des Wärme- und Feuchtigkeitstransports den Einfluss von Axiallüftern auf das Tierwohl in einem Stallsegment analysiert.

Einige der vorliegenden Arbeiten führten Parameterstudien durch, um den Einfluss auf unterschiedliche Zielgrößen zu ermitteln. Die Untersuchung des Fluidtransports steht im Fokus der Arbeiten von SAPOUNAS, DOOREN und SMITS (2013), SAHA, FIEDLER, AMON et al. (2014), YI et al. (2020) und SAHA, YI et al. (2020), wobei unterschiedliche Dachformen und Lüftungsöffnungen von Ställen berücksichtigt werden. FAGUNDES et al. (2020) untersuchten den Einfluss von Axial- und Deckenventilatoren auf die Strömung im Inneren eines Stalls, wobei die Kühe nicht einbezogen wurden. MONDACA, CHOI und COOK (2019) ermittelten den Einfluss der Luftwechselrate auf das Mikroklima in tunnelbelüfteten Ställen. DAMASCENO et al. (2020) evaluieren die Strömungscharakteristik in *Compost bedded pack barns* unter Berücksichtigung unterschiedlicher Windrichtungen und Stallöffnungen.

Die Genauigkeit der CFD-Simulation von Fluid- und Wärmetransport mit verschiedenen Geometrien einer Kuh untersuchten MONDACA und CHOI (2016). Die Arbeit umfasst auch einen weniger lokalen Ansatz zur Approximation einer Kuhherde über ein poröses Medium, wie er auch von DREWRY, MONDACA et al. (2018), MONDACA, CHOI und COOK (2019) und mit ergänzender Anisotropie von DOUMBIA, JANKE, YI, AMON et al. (2021) publiziert ist. Jedoch stellen MONDACA, CHOI und COOK (2019) fest, dass dieser Ansatz nicht ausreichend ist, um eine detaillierte Betrachtung des Mikroklimas im Stall zu ermöglichen.

Die meisten Arbeiten belegen die Übereinstimmung des Simulationsmodells mit der Realität in einer Validierung, entweder anhand von Windkanalversuchen oder den Messungen an realen Ställen. Der Korrelationskoeffizient wird häufig als Validierungsmetrik verwendet und nur VEGA et al. (2022) nutzen die von CHANG und HANNA (2004) empfohlenen Metriken. Die Verifizierung ist selten detailliert dokumentiert und besteht i. d. R. aus einem qualitativen Vergleich der Geschwindigkeitsprofile innerhalb des Stalls.

Zusätzliche Einblicke in die Literatur zur Anwendung von CFD in der Landwirtschaft können den Arbeiten von NORTON, D.-W. SUN et al. (2007), LEE et al. (2013), BJERG, CASONE et al. (2013) und RONG, NIELSEN, BJERG et al. (2016) entnommen werden.

2.5.2 Physik-informierte neuronale Netze und Strömungsmechanik

RAISSI, PERDIKARIS und KARNIADAKIS (2019) lösten mit der Datensparsamkeit des PINN Ansatzes und dessen Anwendung auf nicht lineare zeitabhängige PDEs ein erhebliches Forschungsinteresse aus. Diese Möglichkeit zur Kompensation mangelnder Simulations- oder Messdaten ist für die oft zeitaufwändigen und kostenintensiven CFD-Simulationen und Messungen von besonderem Interesse. Es finden sich daher viele Arbeiten zu PINNs, die sich ausschließlich mit der Lösung der Navier-Stokes-Gleichungen beschäftigen. Tabelle 2.3 kategorisiert diese Arbeiten nach der Reynolds-Zahl, Ablösung der Strömung, Turbulenzmodellierung, betrachteten Geometrie und der Möglichkeit zur Parametrisierung.

Tabelle 2.3: Einordnung der Literatur Physik-informierter neuronaler Netze mit Anwendung auf die Navier-Stokes-Gleichungen

	Re	Str.	Turb.	Geom.	Par.
k. A.	$< 1 \cdot 10^2$ Re	Strömungsabriss	Ohne Klassisch PINN spezifisch	Teilgebiet Vollständig Variabel	Konstant Parametrisierbar
	$< 1 \cdot 10^3$ Re				
	$> 1 \cdot 10^4$ Re				
(Y. ZHU et al., 2019)	•		•	•	•
(L. SUN et al., 2020)	• •		•		•
(RAISSI, YAZDANI und KARNIADAKIS, 2020)	•		•	•	•
(R. WANG et al., 2020)	• ^a		•	• •	•
(JIN et al., 2021)	• •		• ^b	•	•
(ARTHURS und KING, 2021)	•		•	•	•
(H. XU, W. ZHANG und Y. WANG, 2021)		•		•	•
(Q. ZHU, Z. LIU und YAN, 2021)	•		• ^b	•	•
(WANDEL, WEINMANN und KLEIN, 2021)		•	• ^b	•	•
(ARZANI, J.-X. WANG und D'SOUZA, 2021)		•	•	•	•
(HENNIGH et al., 2021)		•	•	•	•
(LAUBSCHER und ROUSSEAU, 2021)	•		• ^b	•	•
(CHAUDHARI, KULKARNI und DAMODARAN, 2021)	•		•	•	•
(SEKAR et al., 2022)		•	•	•	•
(H. WANG, Y. LIU und S. WANG, 2022)		•	• ^b	•	•
(LI und FENG, 2022)	•		•	•	•
(MA et al., 2022)	•	•	•	•	•
(EIVAZI und VINUESA, 2022)	• •	•	• ^b	• •	•
(EIVAZI, TAHANI et al., 2022)	• • •	•	•	•	•
(XIANG et al., 2022)	•	•	• ^b	•	•
(S. XU et al., 2022)	•	•	• ^b	•	•
(HASANUZZAMAN et al., 2023)		•	•	•	•
(RYCK, JAGTAP und MISHRA, 2023)	•		• ^b	•	•
DIESE ARBEIT		•	•	• •	•

^aAls turbulentes Strömungsfeld beschrieben, ohne Angabe der Re.

^bVollständig zeitlich und räumlich aufgelöste Navier-Stokes-Gleichungen.

Viele technische Problemstellungen sind mit hohen Reynolds-Zahlen und entsprechend turbulenten Strömungen verbunden. Turbulente Strömungen führen jedoch zu hochkomplexen Strömungsfeldern sowohl im Raum als auch in der Zeit, was die Lösung der Navier-Stokes-Gleichungen mithilfe von PINNs zu einer anspruchsvollen Aufgabe macht. Dieser Umstand wird in der Einordnung der betrachteten Reynolds-Zahlen deutlich: Der Großteil der Arbeiten ist auf Reynolds-Zahlen kleiner $1 \cdot 10^3$ limitiert. Ausnahmen bilden die Arbeiten von H. XU, W. ZHANG und Y. WANG (2021), HENNIGH et al. (2021), SEKAR et al. (2022), EIVAZI und VINUESA (2022) und EIVAZI, TAHANI et al. (2022). Insbesondere der von EIVAZI, TAHANI et al. (2022) dokumentierte Ansatz ist mit einer Reynolds-Zahl von $2 \cdot 10^5$ hervorzuheben.

Die Literaturanalyse zeigt deutlich, dass mit fortschreitender Forschung zunehmend Strömungsdomänen mit einer abgelösten Strömung betrachtet werden. Häufige Anwendungsbeispiele der Methoden sind die Umströmung eines Zylinders (H. XU, W. ZHANG und Y. WANG, 2021; MA et al., 2022) und die Betrachtung des Nachlaufs hinter dem Zylinder (EIVAZI und VINUESA, 2022; XIANG et al., 2022; S. XU et al., 2022).

Der Großteil der Arbeiten berücksichtigte kein Turbulenzmodell, obwohl die Navier-Stokes-Gleichungen häufig nicht zeitlich aufgelöst wurden. Diese vereinfachte Annahme ist nur bei laminarer Strömung zulässig, weshalb diese Ansätze mit steigender Reynolds-Zahl zu hohen relativen Fehlern führen (SEKAR et al., 2022). Für die häufig eingesetzten RANS-Simulationen zeigen HENNIGH et al. (2021) einen ersten Ansatz, indem das Mischlängenmodell für das Training von PINNs integriert wird. H. XU, W. ZHANG und Y. WANG (2021) und EIVAZI, TAHANI et al. (2022) stellen Ansätze zur Turbulenzmodellierung vor, die spezifisch für PINNs entwickelt wurden und ohne zusätzliche Transportgleichungen auskommen.

Ein Themenschwerpunkt der Arbeiten liegt in der Betrachtung eines Ausschnitts der Geometrie. So analysieren z. B. SEKAR et al. (2022), EIVAZI und VINUESA (2022) und HASANUZZAMAN et al. (2023) insbesondere die Grenzschichten von Strömungen. Ein ebenso häufig vertretenes Anwendungsgebiet ist die Analyse hämodynamischer Strömungen in einem Ausschnitt der Gefäße (L. SUN et al., 2020; RAISSI, YAZDANI und KARNIADAKIS, 2020; ARTHURS und KING, 2021; ARZANI, J.-X. WANG und D'SOUZA, 2021). Nur wenige Arbeiten beschreiben eine Methode für eine variable Geometrie der um- oder durchströmten Objekte (L. SUN et al., 2020; ARTHURS und KING, 2021; WANDEL, WEINMANN und KLEIN, 2021; HENNIGH et al., 2021) oder eine Möglichkeit, ergänzende Parameter variabel zu gestalten (Y. ZHU et al., 2019; WANDEL, WEINMANN und KLEIN, 2021; HENNIGH et al., 2021).

Weitere Analysen der Literatur zur Anwendung von PINNs in der Strömungsmechanik können der Arbeit von CAI et al. (2021) entnommen werden.

2.5.3 Digitale Zwillinge in der Landwirtschaft

GROHER, HEITKÄMPER und UMSTÄTTER (2020) zeigen in der Analyse zum Stand der Digitalisierung in der Landwirtschaft auf, dass in der Milchviehhaltung vorwiegend die Sensorik und Analyse zum Endprodukt fokussiert wird, nicht aber die Produktionsbedingungen selbst. Hier gilt es sowohl die landwirtschaftlich Beschäftigten zu sensibilisieren als auch geeignete Konzepte zu entwickeln. Auch NEETHIRAJAN und KEMP (2021) resümieren, dass weitere Fallstudien zu digitalen Zwillingen in der Tierhaltung erforderlich sind. Um den Forschungsbedarf weiter einzugrenzen, folgt in Tabelle 2.4 eine Analyse und Kategorisierung der Literatur digitaler Zwillinge mit Bezug zur Landwirtschaft. Die Stufe der Realisierung ist nach Konzept, Software und Hardware gestaffelt. Die Betrachtung der Schnittstellen ermittelt proprietäre und nach *Internet of Things* (IoT) standardisierte Konzepte oder Umsetzungen. Eine thematische Einordnung sortiert nach Wassermanagement, Pflanzenzucht und Stallungen. Letztlich wird die Kommunikation in offline und online basierte Konzepte aufgeteilt.

In der analysierten Literatur sind nur wenige Arbeiten zu finden, die alle Aspekte des digitalen Zwillings vom Konzept über die Software bis zur Hardware abdecken und eine umfassende Methode für die Umsetzung bieten (JANS-SINGH et al., 2020; CHAUX, SANCHEZ-LONDONO und BARBIERI, 2021). Während CHAUX, SANCHEZ-LONDONO und BARBIERI (2021) eine grundlegende Beschreibung der teilweise implementierten IoT-Schnittstelle liefern, wird die Kommunikation von JANS-SINGH et al. (2020) über proprietäre Schnittstellen realisiert.

Im Bereich der Stallungen gibt es mehrere Arbeiten, die Teile des digitalen Zwillings entwickeln oder umsetzen. JO, D.-H. PARK, H. PARK und S.-H. KIM (2018) stellen eine Kommunikationsarchitektur für Schweineställe vor, die JO, D.-H. PARK, H. PARK, KWAK et al. (2019) um Ergebnisse verschiedener Simulationen erweitern, die in Zukunft in einen digitalen Zwilling integriert werden sollen. ERDÉLYI und JÁNOSI (2019) präsentieren ein Modell zur Vorhersage des Energiebedarfs und des Gewichtszuwachses in der Schweinemast, mit dem Ziel, es in Zukunft für einen digitalen Zwilling einzusetzen. MOSHREFZADEH, MACHL und GACKSTETTER (2020) entwickeln ein standardisiertes Konzept für die Organisation von gesammelten Daten verteilter digitaler Zwillinge von Ställen und belegen die Funktionalität der Software. Das umfangreichste Konzept für digitale Zwillinge in der Landwirtschaft wird von VERDOUW et al. (2021) vorgestellt, wobei zusätzlich die IT-Sicherheit berücksichtigt wird.

Keine der Arbeiten präsentiert eine vollumfängliche Realisierung des digitalen Zwillings für Stallungen oder stellt ein flexibles Konzept bereit, welches den Einsatz sowohl offline als auch online ermöglicht.

Über die dargelegte Einordnung hinaus bestehen zahlreiche Literaturanalysen zu digitalen Zwillingen in der Landwirtschaft (SREEDEVI und KUMAR, 2020; PYLIANIDIS, OSINGA und ATHANASIADIS, 2021; SLOB und HURST, 2022; NASIRAHMADI und HENSEL, 2022; ABBASI, MARTINEZ und AHMAD, 2022; PURCELL und NEUBAUER, 2023). Ergänzend sei auf die

Einordnungen von NEGRI, FUMAGALLI und MACCHI (2017), KRITZINGER et al. (2018) und JONES et al. (2020) verwiesen (ohne Bezug zur Landwirtschaft).

Tabelle 2.4: Kategorisierte Literatur zu digitalen Zwillingen in der Landwirtschaft

	Real.			Sst.		Th.			Komm.	
	Konzept	Software	Hardware	Proprietär	IoT	Wassermanag.	Pflanzen	Stallungen	Offline	Online
(MONTEIRO et al., 2018)			•		•		•			•
(JO, D.-H. PARK, H. PARK und S.-H. KIM, 2018)	•							•		
(PARGMANN, EUHAUSEN und FABER, 2018)		•	•		•					•
(ERDÉLYI und JÁNOSI, 2019)		•			•			•		
(JO, D.-H. PARK, H. PARK, KWAK et al., 2019)	•							•		
(ALVES et al., 2019)	•	•		•	•	•				•
(ANGIN et al., 2020)	•	•			•		•			•
(JANS-SINGH et al., 2020)	•	•	•	•			•			•
(MOSHREFZADEH, MACHL und GACKSTETTER, 2020)	•	•			•		•	•		•
(CHAUX, SANCHEZ-LONDONO und BARBIERI, 2021)	•	•	•	•	•		•			•
(VERDOUW et al., 2021)	•				•		•	•		•
(MONTEIRO et al., 2022)	•	•		•	•		•			•
(HAN et al., 2022)		•						•		
(YANG et al., 2022)		•			•				•	
(KNIBBE et al., 2022)		•			•	•	•	•		•
DIESE ARBEIT	•	•	•		•		•		•	•

2.6 Wissenschaftlicher Beitrag der Arbeit

Aus dem aktuellen Stand der Forschung zur numerischen Strömungssimulation landwirtschaftlicher Gebäude geht hervor, dass die Mehrheit der Studien den stationären Zustand mithilfe von RANS-Simulationen und dem k - ε -Modell betrachtet. Obwohl die Bedeutung von Emissionen für das Tierwohl und die Umwelt immer wieder betont wird, befassen sich nur wenige Arbeiten mit der Simulation des Stofftransports. Wenn der Stofftransport berücksichtigt wurde, war das betrachtete Berechnungsgebiet oft eingeschränkt. Die Untersuchung des Einflusses von Parametern wird zwar in den Arbeiten häufig berücksichtigt, jedoch selten systematisch mit statistischen Methoden bearbeitet. Die Komplexität der Abbildung des Viehs in der Simulation geht mit der Größe des betrachteten Berechnungsgebiets einher. Keine der Arbeiten führte eine Verifizierung gemäß der Richardson-Extrapolation durch, sondern basierte auf einem qualitativen Vergleich der Geschwindigkeitsprofile. Einige Arbeiten führten keine Verifizierung durch. Die Validierung wurde auf unterschiedliche Weise durchgeführt,

sowohl im Windkanal als auch an realen Ställen. Einige Arbeiten validierten nicht alle untersuchten Größen oder verwendeten Benchmark-Ergebnisse zur Validierung. Im Gegensatz zu den in Kapitel 2.2.4 gegebenen Empfehlungen von CHANG und HANNA (2004) wird häufig der Korrelationskoeffizient zur Bewertung der Modellgüte verwendet. Daraus ergibt sich die erste Forschungsfrage, die in dieser Arbeit beantwortet wird.

Forschungsfrage 1: Lassen sich die multiphysikalischen Effekte von Fluid-, Stoff- und Wärmetransport in und um einen frei belüfteten Kuhstall mittels CFD valide simulieren und der Parametereinfluss auf das Tierwohl, den Emissionsausstoß und die Luftwechselrate über eine statistische Methode systematisch bestimmen?

Der aktuelle Forschungsstand zu PINNs zeigt, dass der Einsatz zur Approximation des Lösungsfeldes der Navier-Stokes-Gleichungen hauptsächlich auf die vollständig räumlich und zeitlich aufgelösten Gleichungen beschränkt ist. Für ingenieurtechnische Anwendungen ist es jedoch oft effektiver, die RANS-Gleichungen zu lösen. Darüber hinaus konzentrieren sich die meisten Studien auf laminare Strömungen oder sehr niedrige Reynolds-Zahlen, da die Approximation von Strömungen mit hohen Reynolds-Zahlen problematisch sein kann (SEKAR et al., 2022; L. SUN et al., 2020). Ergänzend besteht die Problematik komplexer Optimierungslandschaften bei PINNs zur Lösung der Navier-Stokes-Gleichungen, da große konvektive Terme in den Gleichungen zu unphysikalischen Lösungen führen können (KRISHNAPRIYAN et al., 2021). Ferner ist die Anwendung von PINNs auf komplexere Geometrien oder die Verwendung als Metamodell mit variablen Parametern bisher wenig erforscht.

Forschungsfrage 2: Kann die Methode der PINNs dahingehend erweitert werden, dass echtzeitfähige Strömungssimulationen für komplexe Geometrien und variable Eingangsparameter mittels der für technische Anwendungen oft genutzten RANS-Gleichungen, insbesondere für hohe Reynolds-Zahlen, implementiert werden können?

Obwohl es viele Ansätze für digitale Zwillinge im Maschinenbau gibt, befassen sich nur wenige Forschungsarbeiten speziell mit der Landwirtschaft und insbesondere mit der Tierhaltung und den daraus resultierenden speziellen Anforderungen. Ebenso wurde die Bereitstellung oder Integration von Metamodellen in nur wenigen Arbeiten betrachtet, obwohl diese eine vorausschauende Bewertung des Zustands ermöglichen könnten. Zukünftige Konzepte müssen auch in Betracht ziehen, dass es Betriebe gibt, die ein Offline-Konzept bevorzugen. Die Plattform muss auch berücksichtigen, dass landwirtschaftlich Beschäftigte i. d. R. mobil auf dem Gelände unterwegs sind und daher einen Zugriff von mobilen Endgeräten benötigen. Weiterhin müssen Anbieter von Hardwaresystemen für das Stallklima einen standardisierten Zugriff auf die Ergebnisse der Metamodelle erhalten, um die Akteure vorausschauend und optimal einzustellen.

Forschungsfrage 3: Lässt sich ein Kommunikationsschema für frei belüftete Kuhställe entwickeln, das die o. g. echtzeitfähigen Simulationen implementiert, somit den digitalen Zwilling ermöglicht, den Zugriff flexibel off- und online erlaubt sowie die Problematik der Datensicherheit und Mensch-Maschine-Interaktion berücksichtigt?

Experimentelle Ergebnisse

Um die Validität einer Simulation belegen zu können, sind experimentelle Ergebnisse der relevanten Größen unverzichtbar. Daher werden in den folgenden Unterkapiteln die Messungen und Auswertungen von Windgeschwindigkeit, Windrichtung, Lufttemperatur, Luftfeuchtigkeit und Methankonzentration innerhalb eines Versuchsstalls und dessen Umgebung dokumentiert. Kapitel 3.1 beschreibt den Versuchsstall, in dem die Messungen durchgeführt wurden, während Kapitel 3.2 sowohl den Auswahlprozess für die Messgeräte als auch detaillierte technische Informationen zu den Sensoren enthält. Die Positionierung der Sensoren während der Messkampagnen ist im Kapitel 3.3, und die Ergebnisse der Messungen sind in Kapitel 3.4 dokumentiert.

3.1 Versuchsstall

Der frei belüftete Kuhstall wurde im Jahr 2014 geplant und gebaut und befindet sich beim Breitengrad $51^{\circ} 50' 3,57''$ Nord und Längengrad $6^{\circ} 34' 6,4092''$ Ost auf einer Höhe von 21 m über Normalnull. Der Stall ist 15° um seine Hochachse rotiert und befindet sich etwa 70 m westlich von Wohngebäuden. Auf der Südseite des Stalls ist ein Silo platziert, dessen Silage je nach Füllstand 6 m bis 15 m zur Außenwand des Stalls entfernt ist. Etwa 40 m südlich des Stalls und nördlich der Straße befindet sich eine Baumreihe mit Sträuchern. Die Nord- und Westseite der Bodenplatte grenzt an weitläufige Felder, die zum Zeitpunkt der Messungen mit Gras bedeckt waren. Abbildung 3.1 stellt den Stall und die Umgebung anhand einer Karte dar.

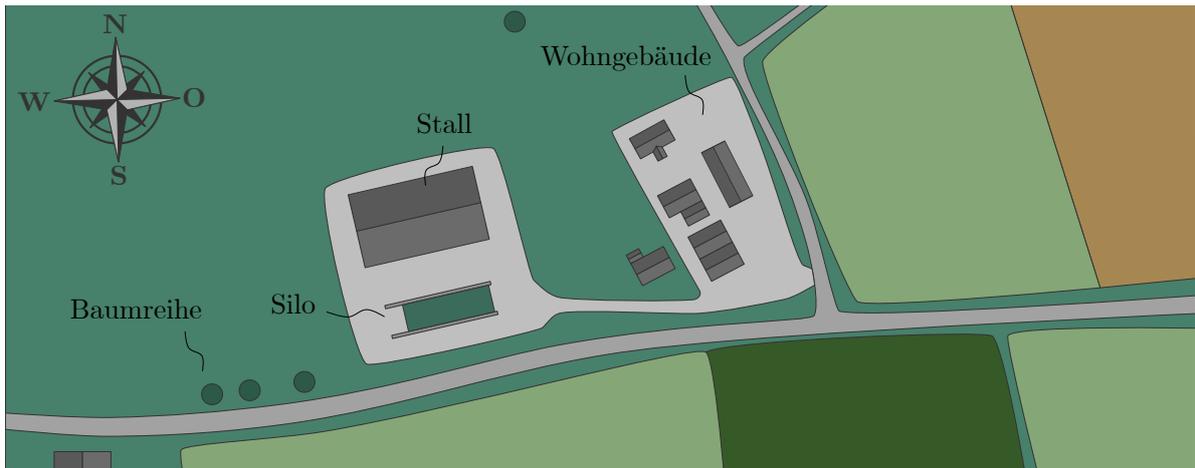


Abbildung 3.1: Karte der Umgebung des Versuchsstalls ($51^{\circ} 50' 3,57''\text{N } 6^{\circ} 34' 6,4092''\text{O}$) mit Kompass und Markierungen relevanter Objekte (eigene Darstellung)

Der Versuchsstall hat eine Breite von 39,10 m und eine Länge von 47,22 m sowie eine Traufhöhe von 4,40 m, was bei einem Dachneigungswinkel von 24° und zusätzlicher Firstlüftung zu einer Gebäudehöhe von 13,40 m führt. Abbildung 3.2 stellt die vereinfachten Geometrien und eine Luftaufnahme des Stalls dar.

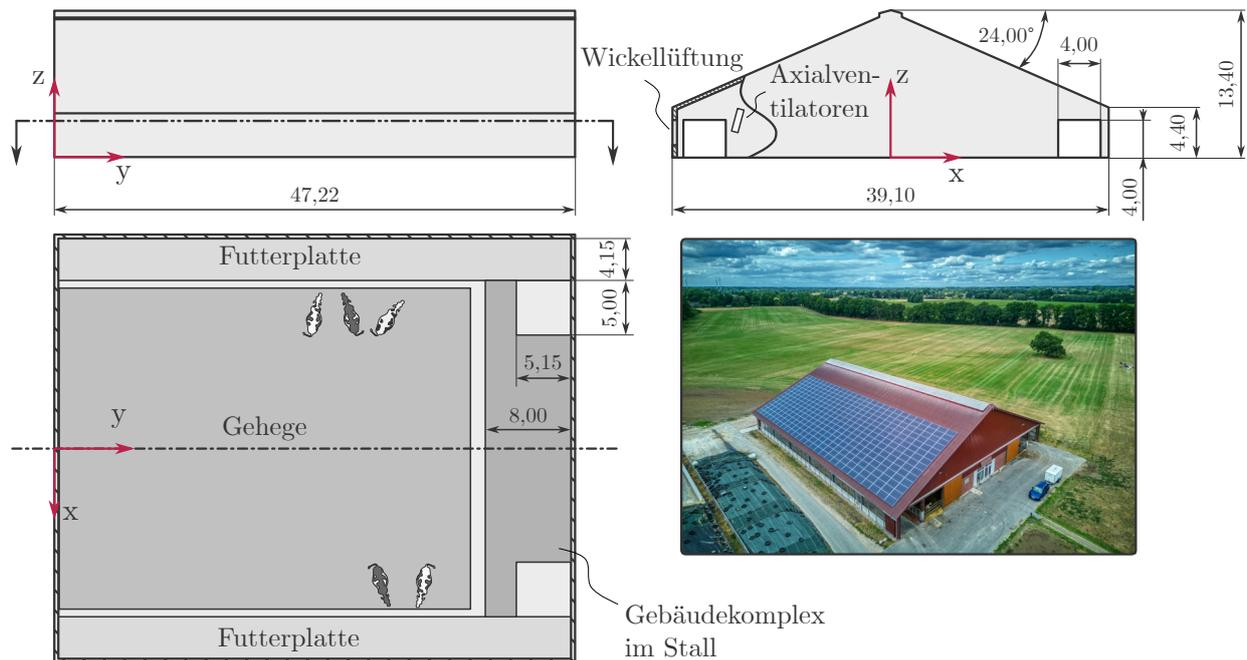


Abbildung 3.2: Aufbau und Luftaufnahme des Versuchsstalls, mit freundlicher Genehmigung der Firma WELSING MILCHVIEH (eigene Darstellung)

Im Inneren des Gebäudes befinden sich Funktionsräume, mit zwei daran angrenzenden Melkstationen, die wiederum an der Nord- und Südseite an die außen liegenden Futterplatten grenzen (s. Abbildung 3.3, rechts). Die Futterplatten sind von beiden Stirnseiten (Ost und West)

über Tore zugänglich (s. Abbildung 3.3, links). Die Liegeboxen im Gehege sind durch Betonwände begrenzt und mit Stroh ausgelegt (s. Abbildung 3.3, mittig). Das Lagerungssystem für den Dung ist unterhalb des Spaltenbodens im Keller des Gebäudes angelegt. Sparren und Pfetten aus Holz bilden die Stützkonstruktion des Dachs. Das Gebäudevolumen, abzüglich der Funktionsräume und der Stützkonstruktion, beträgt $15.396,20 \text{ m}^3$. Die Wickellüftung ist an der Gebäudelängsseite verbaut und öffnet von oben nach unten. Die Wickellüftung kann nicht über die gesamte Traufhöhe geöffnet werden, da eine Wand mit der Höhe von $0,80 \text{ m}$ oberhalb der Bodenplatte die Querbelüftung versperrt. Die Firstlüftung kann über Klappen geöffnet werden, die eine Fläche mit einer Höhe von $0,33 \text{ m}$ über die gesamte Gebäudelänge freigeben. Aktive Lüftungselemente in Form von insgesamt zwölf Axiallüftern sind an der Nord- und Südseite längs des Gebäudes in einem Abstand von je $9,00 \text{ m}$ platziert. Die Oberkante der Axiallüfter schließt mit der Traufhöhe ab. Im Jahresmittel kann ein Besatz von etwa 120 Kühen angenommen werden.



Abbildung 3.3: Fotos des Versuchsstalls, mit freundlicher Genehmigung der Firma WELSING MILCHVIEH

3.2 Messinstrumente

Im Folgenden wird der Prozess der Messinstrumentenauswahl dokumentiert und dabei detaillierte technische Informationen bereitgestellt. Zusätzlich ist die Entwicklung und Kalibrierung des Methansensors erläutert.

3.2.1 Strömungssensoren

Vor der Auswahl der Messinstrumente wurde eine vereinfachte Strömungssimulation durchgeführt. Dabei wurde festgestellt, dass in Tierställen bei einer örtlichen Windgeschwindigkeit von $3,5 \frac{\text{m}}{\text{s}}$, die dem Median der Windgeschwindigkeit der Jahre 2019 bis 2022 entspricht, mit Strömungsgeschwindigkeiten der freien Strömung im Stall bis hinunter auf $0,5 \frac{\text{m}}{\text{s}}$ zu rechnen ist. Staudrucksensoren erreichen üblicherweise nicht die genannten niedrigen Windgeschwindigkeiten. Die hohe Staubbelastung in den Ställen schließt den betriebssicheren Einsatz von

Hitzdrahtanemometern aus. Zudem sind für die *Particle Image Velocimetry* hohe Anschaffungskosten und ein hoher Aufwand zur Installation im realen Versuchsstall zu erwarten. Aus diesen Gründen ist das Messprinzip der Anemometer am besten für diskrete Geschwindigkeitsmessungen in Tierställen geeignet (HEIDENREICH et al., 2008; FIEDLER et al., 2013). Schalenkreuzanemometer ermitteln lediglich die ebenen Geschwindigkeitskomponenten, weshalb insbesondere Ultraschallanemometer (USA) für dreidimensionale Geschwindigkeitsmessungen geeignet sind. In Tabelle 3.1 sind die technischen Eigenschaften der für diese Arbeit ausgewählten und eingesetzten 3D-USA aufgeführt.

Tabelle 3.1: Spezifikationen des 3D-USA (Modell 4.3830.22.300) der Firma ADOLF THIES

Quantität	Messbereich		Auflösung	Unsicherheit
	min.	max.		
Windgeschw. in $\frac{m}{s}$	0	85	0,1	$\pm(0,1 + 1 \%)$
Windrichtung in $^\circ$	0	360	1,0	$\pm 1,0$
Virtuelle Temperatur in $^\circ C$	-50	80	0,1	$\pm 0,5$

Die Ermittlung der Windgeschwindigkeit erfolgt mittels der Superposition von Wind- und Schallgeschwindigkeit, wobei die Schallgeschwindigkeit mithilfe von zwei gegenüberliegenden Sonden emittiert und gemessen wird. Durch die bidirektionale Messung wird die Einflussnahme der Temperatur auf die Messung mithilfe der akustischen virtuellen Temperatur eliminiert. Wenn die Messstrecken senkrecht zueinander ausgerichtet sind, können die Geschwindigkeitskomponenten u , v und w differenziert werden. Der Aufbau und die Messstrecken des für die Arbeit ausgewählten und eingesetzten 3D-USA sind in Abbildung 3.4 dargestellt.

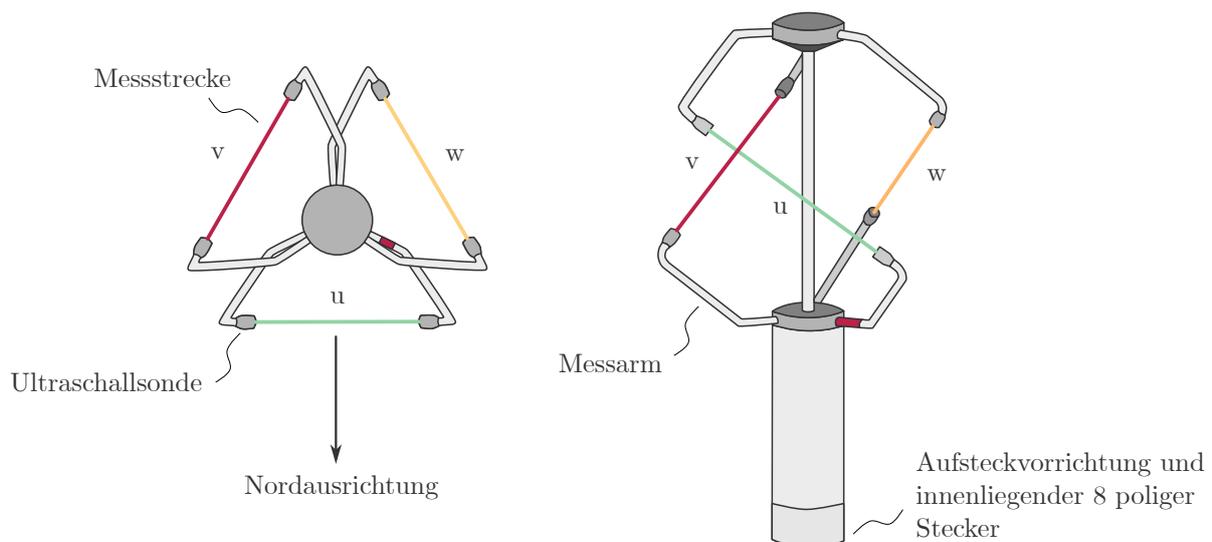


Abbildung 3.4: Aufbau des 3D-USA und Darstellung der Messstrecken (eigene Darstellung)

Aus der akustischen virtuellen Temperatur für trockene Luft $T_{\text{virt,ref}}$ kann die vorherrschende Lufttemperatur berechnet werden. Die resultierende Temperatur wird aufgrund der indirekten Berechnung weiterhin mit T_{virt} gekennzeichnet:

$$T_{\text{virt}} = T_{\text{virt,ref}} \cdot \left(1 + 0,329 \frac{e_d}{p}\right), \quad (3.1)$$

wobei e_d dem Dampfdruck und p dem Luftdruck entspricht. Über die Luftfeuchte φ und den Sättigungsdampfdruck e_s kann der Dampfdruck berechnet werden:

$$e_d = \frac{\varphi}{100} \cdot e_s. \quad (3.2)$$

3.2.2 Methansensoren

Laut Messungen von STANLEY et al. (2018) liegt der Methangehalt in der Atmosphäre bei etwa 2,0 *parts per million* (ppm). Dies schränkt die Auswahl geeigneter Sensoren ein, da viele Methansensoren für den Explosionsschutz entwickelt und kalibriert wurden ($\text{UEG}_{\text{CH}_4} = 44.000$ ppm). Die Messung der atmosphärischen Konzentration kann mit Infrarotpyrometern oder Gaschromatographen durchgeführt werden. Diese Methoden sind jedoch aufgrund ihrer hohen Kosten und der Anfälligkeit gegenüber Staub für einen langfristigen Einsatz in Ställen nur bedingt geeignet. EUGSTER und KLING (2012) und EUGSTER, LAUNDRE et al. (2020) haben gezeigt, dass Metalloxid-Halbleiter-Sensoren (MOX-Sensoren) zur Messung der atmosphärischen Konzentration geeignet sind und auch unter extremen Wetterbedingungen eingesetzt werden können. Der TGS 2600 MOX-Sensor von FIGARO ENGINEERING ist für einen Betriebsbereich von 1,0 bis 100,0 ppm ausgelegt, wobei der optimale Betriebsbereich bei 1,0 bis 30,0 ppm liegt. BJERG, G. ZHANG et al. (2011) belegen in einer Messreihe eine zu erwartende Konzentration in frei belüfteten Kuhställen von 0,2 bis 30,5 ppm, welche durch den optimalen Betriebsbereich des TGS 2600 weitestgehend abgedeckt wird.

Die Charakteristik des TGS 2600 Sensors zeigt eine nicht lineare Beziehung zwischen der Gaskonzentration und der Leitfähigkeit des Sensor-Substrats. Die Gaskonzentration wird durch das Verhältnis des gemessenen Widerstands R_s zum Widerstand bei frischer Luft R_0 berechnet. Das Verhältnis hängt von der Lufttemperatur und Luftfeuchtigkeit ab. Ein 16 Bit Analog-Digital-Wandler (ADS1115) wurde verwendet, um den Spannungswert zu digitalisieren, der über den MOX-Widerstand gemessen wurde. Die Spannungsversorgung und Signalverarbeitung wurden von einem RASPBERRY PI ZERO 2 W übernommen. Ein Präzisionswiderstand R_L mit $10 \text{ k}\Omega \pm 0,1 \%$ ist gemäß Datenblatt im elektrischen Schaltkreis implementiert. Der Sensor wurde in atmosphärischer Konzentration und mit Prüfgasen bei Konzentrationen von 10,0 ppm und 100,0 ppm Methan in synthetischer Luft kalibriert. Ein einstellbarer Druckminderer sorgte für einen konstanten Volumenstrom in der Prüfkammer. Aufgrund des Messprinzips benötigt der Sensor eine Aufheizzeit von ein bis zwei Stunden,

die je nach Konzentration variiert. Abbildung 3.5 zeigt den Schaltplan und den Aufbau der Kalibrierungsmessungen sowie die Messung während der Aufheizphase des Sensors in atmosphärischer Konzentration. Die Messung zeigt, dass Böen trotz der Aufstellung des Sensors in einem windberuhigten Bereich einen erheblichen Einfluss auf die Messung eines MOX-Sensors haben können. Daher wurden die Sensoren für die Messungen im Versuchsstall in einem Gehäuse verbaut.

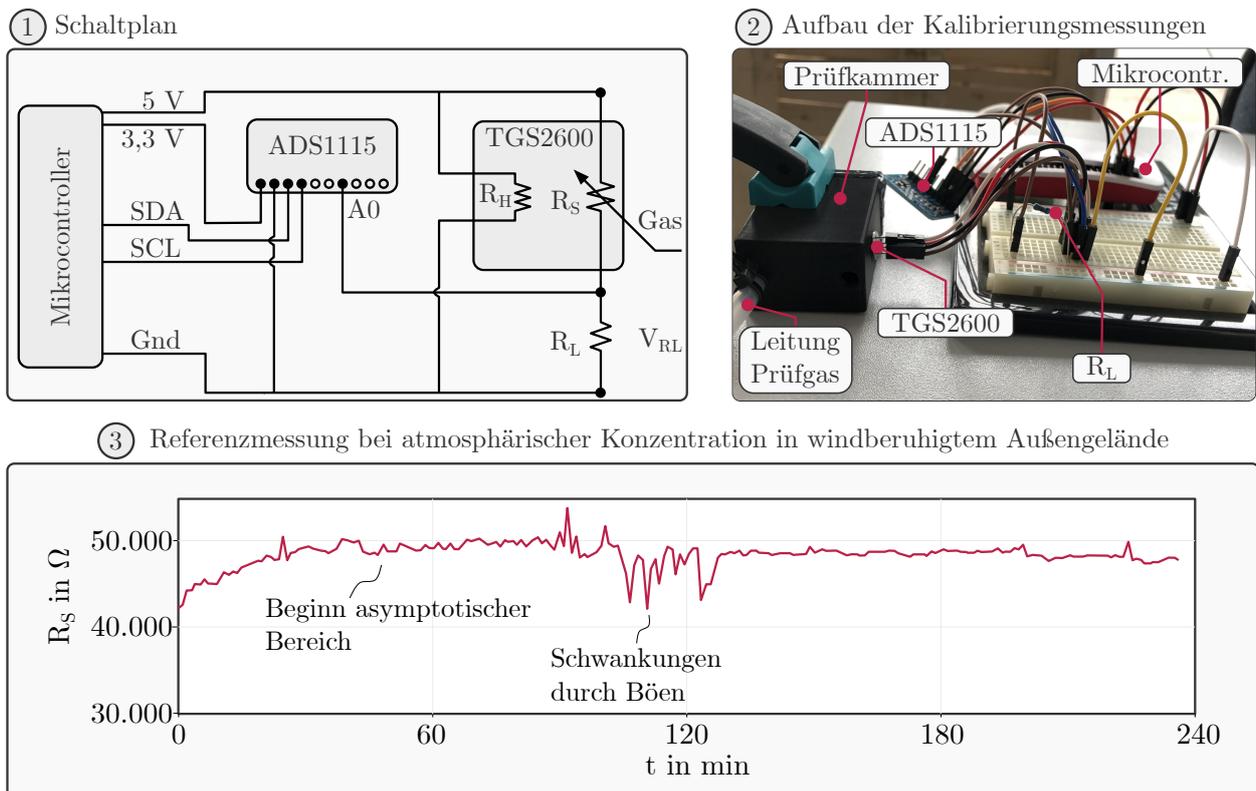


Abbildung 3.5: Markierung **1** zeigt den Schaltplan eines TGS 2600 MOX-Sensors und der Auswerteeinheit, **2** stellt den Aufbau der Kalibrierungsmessungen und **3** das Diagramm der Messwerte einer Kalibrierungsmessung bei atmosphärischer Konzentration dar (eigene Darstellung)

3.2.3 Lubrateg SmartBox

Der Hersteller des Kombisensors gibt für die in dieser Arbeit verwerteten Messungen der Luftfeuchte und Lufttemperatur die in Tabelle 3.2 gelisteten Spezifikationen an. Für die Messungen der SMARTBOX wird ein SENSIRION SCD30 Sensormodul verwendet, welches neben den benannten Quantitäten auch den CO_2 Gehalt der Luft erfasst. Die LUBRATEC SMARTBOX ist in einem Gehäuse verbaut, welches den Luftaustausch bei gleichzeitigem Staubschutz der Elektronik über ein Vlies realisiert. Die Sensorwerte werden mittels *Long Range Wide Area Network* (LoRaWAN) an einen Server des Herstellers übermittelt und sind über eine Webapplikation abrufbar.

Tabelle 3.2: Spezifikationen der LUBRATEC SMARTBOX von HUESKER SYNTHETIC

Quantität	Messbereich		Auflösung	Unsicherheit
	min.	max.		
Lufttemperatur in °C	-40,00	70,00	0,10	$\pm(0,40 + 0,023 \cdot (T - 25,00))$
Luftfeuchte in %	0,00	100,00	1,00	$\pm 3,00$

3.2.4 Wetterstation – Deutscher Wetterdienst

Die geografischen Koordinaten der nächstgelegenen Wetterstation des Deutschen Wetterdienstes (DWD) sind der Breitengrad 51° 49' 44,4" Nord und Längengrad 6° 32' 13,2" Ost, was einer Luftlinien-Entfernung von 2,12 km zum Versuchsstall entspricht. Die Wetterstation trägt die nationale Kennung „Bocholt-Liedern (Wasserwerk)“ mit der Kennnummer 00554. Zur Erfassung der Wetterdaten kommen verschiedene Geräte zum Einsatz, die in den Metadaten des *Open Data* Bereichs des *Climate Data Center* des DWD aufgeführt sind. Die Windgeschwindigkeit und -richtung werden in zehn Metern Höhe über dem Boden erfasst. In zwei Metern Höhe über dem Boden erfolgt die Messung der Lufttemperatur. Zum Schutz vor Sonneneinstrahlung und Witterungseinflüssen befinden sich die Temperatur- und Feuchte-sensoren in einer Lamellen-Wetterhütte.

Die Messung der Windgeschwindigkeit erfolgt mittels eines Schalenkreuzanemometers mit optoelektronischer Abtastung (Modell 4.3303, Firma ADOLF THIES). Eine Windfahne (Modell 4.2121, Firma ADOLF THIES) ermöglicht es, die Windrichtung zu bestimmen. Die Lufttemperatur wird mit einem Widerstandsthermometer (PT 100) gemessen, wobei laut Angaben des DWD verschiedene Hersteller zugelassen sind. Die Feuchtigkeit wird mittels eines HYGROMER MP100 der Firma ROTRONIK bestimmt, wobei die elektrische Kapazität von der relativen Luftfeuchte abhängt, da der Sensor mit einem Spezialkunststoff ummantelt ist. Eine Zusammenfassung der technischen Daten der Sensoren ist in Tabelle 3.3 aufgeführt.

Tabelle 3.3: Spezifikationen der Geräte der Wetterstation des DWD mit nationaler Kennung 00554 in Bocholt-Liedern (Wasserwerk)

Quantität	Messbereich		Auflösung	Max. Unsicherheit	Hersteller
	min.	max.			
Windgeschw. in $\frac{m}{s}$	0,3	50,0	k. A.	$\pm 0,5$	ADOLF THIES
Windrichtung in °	0,0	360,0	2,50	$\pm 2,5$	ADOLF THIES
Lufttemperatur in °C	-100,0	400,0	0,01	$\pm(0,15 + 0,05 \%)$	k. A.
Luftfeuchte in %	0,0	100,0	k. A.	$\pm 1,0$	ROTRONIK

3.2.5 Wetterstation am Versuchsstall

Im Laufe der durchgeführten Messkampagnen konnten die Daten der Wetterstation des Deutschen Wetterdienstes durch lokal am Versuchsstall erhobene Daten ergänzt werden. Die hierfür am Versuchsstall installierte Wetterstation gibt die Messgrößen zur Lufttemperatur, Windrichtung und Windgeschwindigkeit aus, welche mit einer zeitlichen Auflösung von $\Delta t = 10$ s verfügbar sind. Die Außentemperatur wird über den TMP112NAIDRLT Temperatursensor der Firma TEXAS INSTRUMENTS über den thermoelektrischen Effekt in einer Diode ermittelt. Das Schalenkreuzanemometer und die Windfahne, zur Bestimmung von Windrichtung und -geschwindigkeit, werden über Hall-Effekt-Sensoren ausgewertet. Tabelle 3.4 listet die Werte zur Auflösung und Unsicherheit der Sensoren.

Tabelle 3.4: Spezifikationen der Geräte der lokalen Wetterstation am Versuchsstall

Quantität	Messbereich		Auflösung	Max. Unsicherheit	Hersteller
	min.	max.			
Windgeschw. in $\frac{\text{m}}{\text{s}}$	0,7	50,0	0,02	$\pm 2,0$ %	k. A.
Windrichtung in $^\circ$	0,0	360,0	2,00	$\pm 2,0$	k. A.
Lufttemperatur in $^\circ\text{C}$	-40,0	125,0	0,05	$\pm 0,5$	TEXAS INSTRUMENTS

3.3 Messaufbau

Die Messung der Windgeschwindigkeiten im Versuchsstall erfolgte mithilfe von zwei 3D-USA, welche in Kapitel 3.2.1 beschrieben wurden. Einer der Sensoren wurde mittig auf einer Maststange im Stall in einer Höhe von etwa 4,40 m montiert, um ein Erreichen durch die Kühe zu verhindern. Um eine Verkabelung aus dem Gehege zu ermöglichen, wurde der erste Sensor in einer Entfernung von $\Delta y = -10,40$ m zu den Funktionsräumen im Inneren des Stalls aufgestellt. Der zweite Sensor wurde auf einem Stativ installiert und für jeweils 60 Minuten an verschiedenen Standorten auf der Nord- und Südseite des Stalls positioniert, um im Laufe des Tages sechs Messungen durchführen zu können. Die Positionen der Sensoren wurden mittels Laser-Entfernungsmessgerät bestimmt und dokumentiert.

Die am Versuchsstall in den Jahren 2019 bis 2022 am häufigsten vorkommende Windrichtung war eine Anströmung aus Süden. Daher wurden Methansensoren an Luv und Lee platziert, um die Werte der dem Wind zu- und abgewandten Seite messen zu können. Die LUBRATEC SMARTBOX von HUESKER SYNTHETIC, zur Messung der Luftfeuchte, wurden auf einer Höhe von 2,00 m angebracht, wobei acht Sensoren im Inneren des Stalls verteilt sind: Vier Sensoren befinden sich auf der Futterplatte im Südosten, Südwesten, Nordwesten und Nordosten und vier weitere Sensoren sind im Gehege in identischer Himmelsrichtung platziert. Die örtliche

Wetterstation am Versuchsstall ist auf der Ostseite des Stalls in Firsthöhe montiert. Abbildung 3.6 zeigt die Positionen der Sensoren für die drei durchgeführten Messkampagnen.

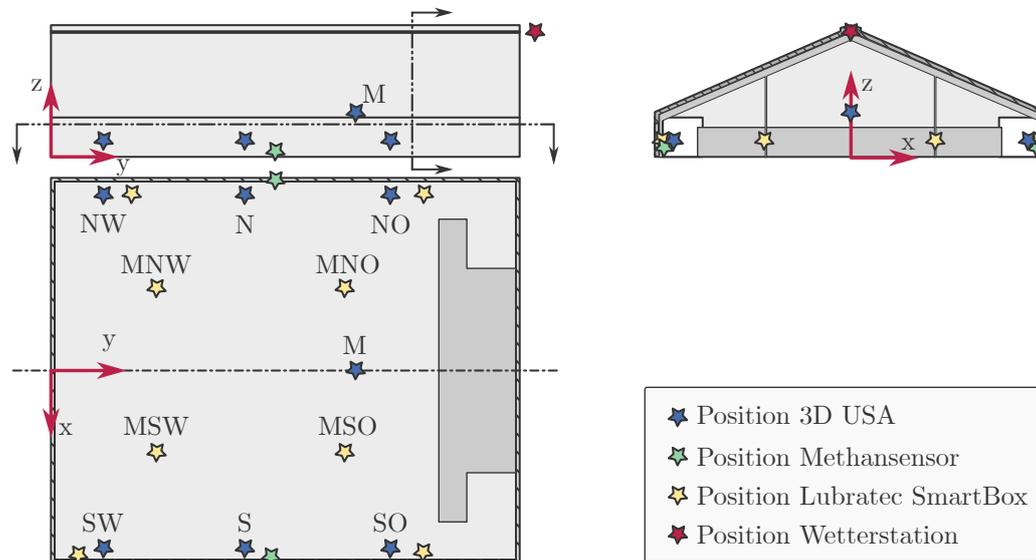


Abbildung 3.6: Übersicht der Positionen aller Sensoren am und im Stall (eigene Darstellung)

Da sich die exakte Position des 3D-USA auf dem Stativ und der Methansensoren je Messkampagne unterschied, listet Tabelle 3.5 die genauen Positionsangaben der Sensoren über alle Messkampagnen.

Tabelle 3.5: Positionsangaben der Sensoren im lokalen Koordinatensystem des Versuchsstalls

Sensor	Pos.	①			②			③		
		x in m	y in m	z in m	x in m	y in m	z in m	x in m	y in m	z in m
3D-USA	SO	19,1	40,9	1,7	19,1	32,1	1,7	19,1	28,0	1,7
	S	19,1	24,5	1,7	19,1	21,0	1,7	19,1	18,0	1,7
	SW	19,1	14,1	1,7	19,1	13,5	1,7	19,1	12,1	1,7
	M	0,0	28,2	4,5	0,0	28,2	4,5	0,0	28,2	4,5
	NO	-19,1	29,6	1,7	-19,1	33,0	1,7	-19,1	33,7	1,7
	N	-19,1	21,0	1,7	-19,1	24,2	1,7	-19,1	25,7	1,7
	NW	-19,1	4,3	1,7	-19,1	8,9	1,7	-19,1	11,4	1,7
Wetterstation					48,1	0,0	12,8	48,1	0,0	12,8
Methansensoren	S	19,3	28,6	0,3	19,3	27,1	0,3	19,3	27,2	0,3
	N	-19,1	15,6	0,3	-19,3	15,1	0,3	-19,3	15,0	0,3
LUBRATEC SMARTBOX	SO	18,8	34,8	2,0	18,8	34,8	2,0	18,8	34,8	2,0
	SW	18,8	12,1	2,0	18,8	12,1	2,0	18,8	12,1	2,0
	NO	-18,8	34,8	2,0	-18,8	34,8	2,0	-18,8	34,8	2,0
	NW	-18,8	12,1	2,0	-18,8	12,1	2,0	-18,8	12,1	2,0
	MSO	8,3	34,8	2,0	8,3	34,8	2,0	8,3	34,8	2,0
	MSW	8,3	12,3	2,0	8,3	12,3	2,0	8,3	12,3	2,0
	MNO	-8,3	12,3	2,0	-8,3	12,3	2,0	-8,3	12,3	2,0
MNW	-8,3	34,8	2,0	-8,3	34,8	2,0	-8,3	34,8	2,0	

3.4 Messkampagnen und Auswertung

Äquivalent zu BJERG, G. ZHANG et al. (2011) und JANKE, WILLINK et al. (2020) ist die Messung in Messkampagnen für die Jahreszeiten Sommer, Übergang und Winter aufgeteilt, um ein saisonales klimatisches Spektrum abzubilden. Die Sommermesskampagne fand am 27.07.2022 (KW 30), die Übergangsmesskampagne am 22.10.2022 (KW 42) und die Wintermesskampagne am 31.01.2023 (KW 5) statt. Die Messkampagnen sind durch ①, ② und ③ für Sommer, Übergang und Winter gekennzeichnet. In den folgenden Unterkapiteln werden die einzelnen Messgrößen ausgewertet und ergänzende statistische Kenngrößen dokumentiert.

3.4.1 Lüftungselemente

Da die Einstellung der passiven und aktiven Lüftungselemente im Versuchsstall die Messungen der Windgeschwindigkeiten maßgeblich beeinflussen kann, sind im Folgenden die Einstellungen für die drei Messkampagnen dokumentiert.

Die Wickellüftung im Versuchsstall war während der Messkampagne ① vollständig geöffnet, wobei dies einer Resthöhe der Versperrung von 0,80 m ab dem Grund entspricht. Die Axiallüfter variierten entsprechend den in Tabelle 3.6 aufgeführten Zeitpunkten der veränderten Einstellungen zwischen 0 % und 100 % der Last. Während der Messkampagne war die Firstlüftung vollständig geöffnet.

Tabelle 3.6: Zeitfolge der Lüftungseinstellungen für die Axiallüfter der ersten Messkampagne

Uhrzeit	Einstellung in %
10:27:00	0
12:15:00	40
12:57:00	100
13:11:00	40
13:45:00	100
13:57:00	40

Während der Messkampagne ② war die Wickellüftung vollständig geöffnet und die Axiallüfter über die gesamte Dauer der Messung abgeschaltet. Die Firstlüftung des Versuchsstalls wurde zwischen 13:05 Uhr und 14:00 Uhr geschlossen, da Niederschlag detektiert wurde.

Im Zuge der Messkampagne ③ wurde die Wickellüftung zunächst zu 75 % bis auf 1,10 m oberhalb der Bodenplatte heruntergefahren. Ab 13:10 Uhr und bis zum Ende der Messung der Messkampagne ③ ist die Wickellüftung auf 1,55 m Höhe oberhalb der Bodenplatte verstellt

worden, was einer Öffnung von 64,8 % entspricht. Die Axiallüfter waren über die gesamte Dauer der Messung abgeschaltet. Die Firstlüftung schloss sich mit einsetzendem leichtem Regen ab 13:10 Uhr und öffnete wieder ab 15:05 Uhr.

3.4.2 Strömungsgeschwindigkeiten

3D-USA

Die Messfrequenz wurde für beide 3D-USA auf 1,0 Hz eingestellt. Das Messprotokoll übertrug die Komponenten der Windgeschwindigkeit Azimuth und Elevation sowie die Windrichtung. Die Geschwindigkeitskomponenten wurden über eine Koordinatentransformation in das lokale Koordinatensystem des Stalls überführt und korrespondierend zur x -, y - und z -Achse mit u , v und w bezeichnet. Die Abbildung 3.7 stellt den Median, den Interquartilsabstand (IQR), zwischen dem ersten Quartil Q_1 und dritten Quartil Q_3 sowie die Antennen für die Messgrößen in einem Boxplot dar. Gemäß Tukeys ursprünglicher Definition sind die Antennen mit $Q_1 - 1,5 \cdot \text{IQR}$ und $Q_3 + 1,5 \cdot \text{IQR}$ ausgewertet.

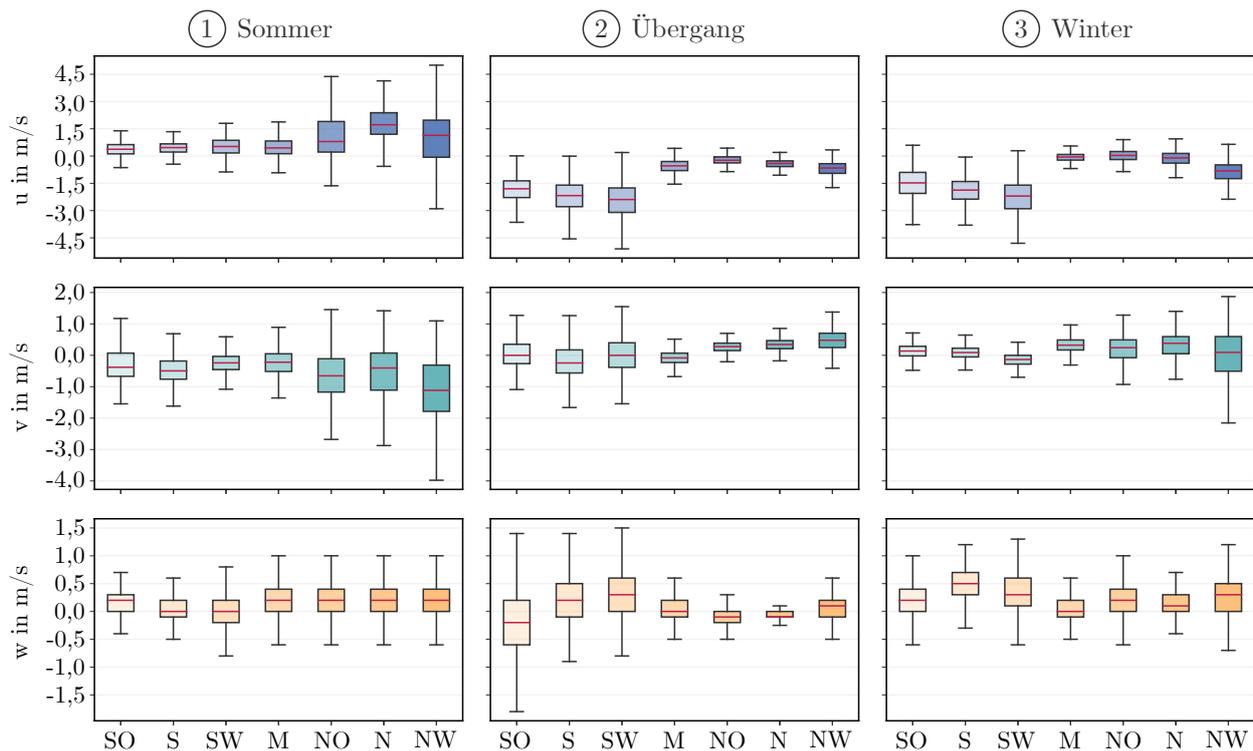


Abbildung 3.7: Boxplot der Geschwindigkeitskomponenten u , v und w im lokalen Koordinatensystem des Stalls für alle Sensorpositionen und Messkampagnen (eigene Darstellung)

Die Boxplots der Sensoren zeigen über alle Messkampagnen für die Hauptgeschwindigkeitskomponente u größere Antennen an der dem Wind zugewandten Seite (Luv), wohingegen

diese an der dem Wind abgewandten Seite (Lee) kleiner ausfallen. Die Sensoren an der Luv-Seite der Messkampagne ① weisen im Vergleich größere Antennen auf. Da außerdem ein dezentraler Median zu erkennen ist, deutet dies z. B. auf eine zeitliche Veränderung der Windrichtung oder Windgeschwindigkeit hin. Ein Vergleich der Hauptgeschwindigkeitskomponenten u der Messkampagnen ② und ③ zeigt an, dass bei gleicher Windrichtung (Süd bis Südost) reproduzierbare Messergebnisse erzielt werden. Im Anhang A sind zusätzlich die Mittelwerte und Turbulenzgrößen in Tabellenform aufgeführt.

Wetterstationen

Sowohl die Wetterstation des DWD als auch die am Versuchsstall betriebene lokale Wetterstation werden im Folgenden ausgewertet. Die Station des DWD liefert gemittelte Windgeschwindigkeiten über eine Stunde. Die lokale Wetterstation, die im Rahmen der Messkampagne ② in Betrieb genommen wurde, ist mit einer Frequenz von 0,1 Hz aufgelöst. In Abbildung 3.8 werden die Windrichtungen und -geschwindigkeiten außerhalb des Stalls mithilfe von Windrosen visualisiert. Messkampagne ① weist mit $\bar{\alpha}_{\text{DWD}} = 345,73^\circ$ einen Nord-Nord-Westwind bei mittlerer Geschwindigkeit Azimut von $\bar{U}_{\text{DWD}} = 2,81 \frac{\text{m}}{\text{s}}$ auf. Während der Messkampagne ② ist die mittlere Windrichtung nach DWD $\bar{\alpha}_{\text{DWD}} = 209,00^\circ$, wohingegen die lokale Wetterstation eine mittlere Windrichtung von $\bar{\alpha}_{\text{Hof}} = 186,8^\circ$ misst. Die Windgeschwindigkeiten Azimut unterscheiden sich mit $\bar{U}_{\text{DWD}} = 4,69 \frac{\text{m}}{\text{s}}$ und $\bar{U}_{\text{Hof}} = 2,12 \frac{\text{m}}{\text{s}}$ ebenfalls. Die Auswertung der Messkampagne ③ weist für die Windrichtungen $\bar{\alpha}_{\text{DWD}} = 235,67^\circ$ und $\bar{\alpha}_{\text{Hof}} = 227,20^\circ$ auf. Mit $\bar{U}_{\text{DWD}} = 6,63 \frac{\text{m}}{\text{s}}$ und $\bar{U}_{\text{Hof}} = 3,56 \frac{\text{m}}{\text{s}}$ weisen die mittleren Windgeschwindigkeiten der dritten Messkampagne die höchsten Werte aller drei Messkampagnen auf.

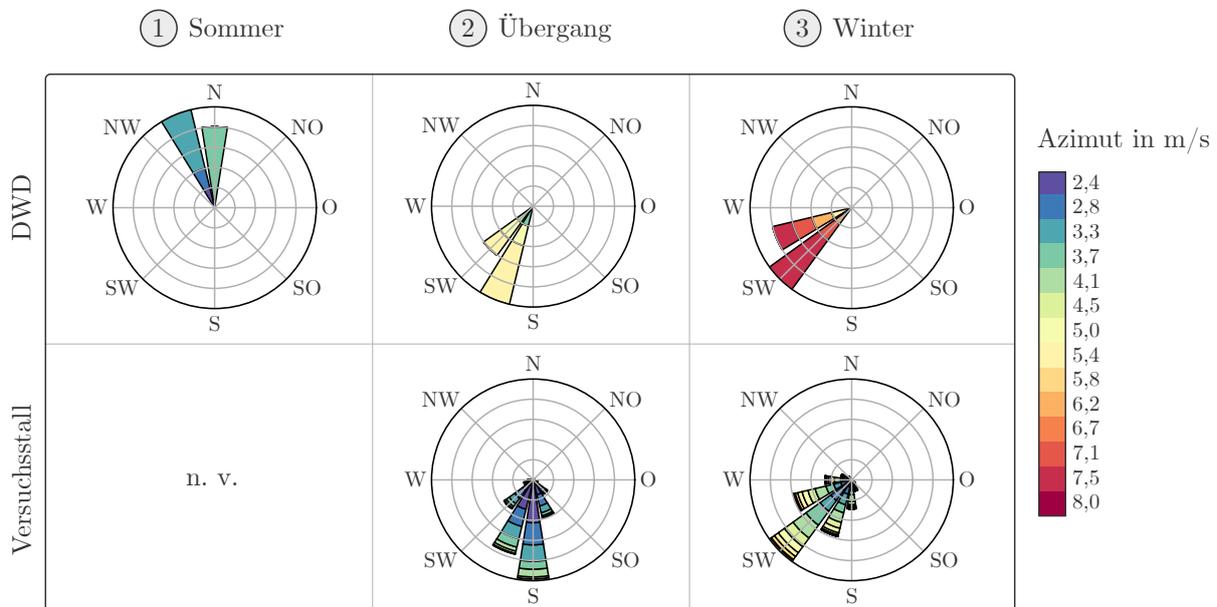


Abbildung 3.8: Windrosen der Messstation des DWD und am Versuchsstall über alle Messkampagnen (eigene Darstellung)

3.4.3 Lufttemperatur

3D-USA

Die virtuellen Lufttemperaturen für trockene Luft wurden im Stall an den Positionen der Strömungsmessungen der 3D-USA aufgezeichnet. Aus der Korrektur nach Gleichung 3.1 folgen die Boxplots der virtuellen Temperaturen in Abbildung 3.9.

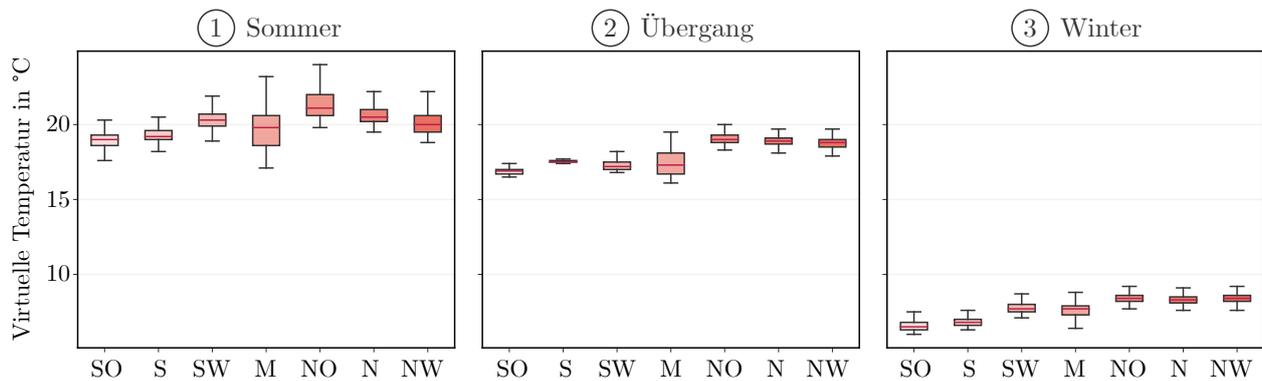


Abbildung 3.9: Boxplot der virtuellen Lufttemperatur gemäß den 3D-USA für alle Sensorpositionen und Messkampagnen (eigene Darstellung)

Die größten Antennen weisen jeweils die Auswertungen des in der Stallmitte platzierten Sensors auf. Die dezentrale Lage des Medians im IQR deutet auf eine zeitliche Veränderung der Lufttemperatur im Inneren des Stalls hin. Zudem ist zu erkennen, dass die Temperaturmessung an der Luv-Seite während Messkampagne ① gegenüber den Messkampagnen ② und ③ größere Antennen aufweist.

Wetterstationen

Die Lufttemperaturen der Messstation des DWD wurden stündlich gemittelt, wohingegen die Messwerte der lokalen Station am Versuchsstall mit einer Frequenz von 0,1 Hz aufgelöst sind. Für die einzelnen Messkampagnen ergaben sich die in Tabelle 3.7 aufgelisteten mittleren Temperaturen und zugehörigen Standardabweichungen. Die Differenz zwischen dem Messwert des DWD und der lokalen Wetterstation belegt einen jeweils höheren Messwert der Lufttemperatur am Versuchsstall.

Tabelle 3.7: Mittlere Temperaturen der Wetterstationen des DWD und am Versuchsstall für die durchgeführten Messkampagnen

	①	②	③
\bar{T}_{DWD} in $^{\circ}\text{C}$	18,90	16,15	5,62
$\sigma_{T_{\text{DWD}}}$ in $^{\circ}\text{C}$	$\pm 1,50$	$\pm 1,20$	$\pm 0,74$
\bar{T}_{Stall} in $^{\circ}\text{C}$	n. v.	17,55	6,84
$\sigma_{T_{\text{Stall}}}$ in $^{\circ}\text{C}$	n. v.	$\pm 0,79$	$\pm 0,68$

3.4.4 Luftfeuchte

Die klimatischen Bedingungen der Luftfeuchte wurden während der Messkampagnen sowohl an der Wetterstation des DWD als auch an den LUBRATEC SMARTBOXEN im Innenbereich erfasst. Die Werte der Messstation des DWD werden stündlich gemittelt, während die LUBRATEC SMARTBOXEN die Werte alle zehn Minuten mitteln. In Tabelle 3.8 sind die gemittelten Luftfeuchtigkeiten der einzelnen Kampagnen aufgeführt. Daten, die aufgrund von Übertragungsfehlern des Sensors an die Datenbank des Herstellers fehlen, werden als nicht vorhanden markiert. Im Vergleich zum Messwert der Wetterstation des DWD ergeben die Messwerte der Sensoren für die Messkampagne ① und ② im Stall eine erhöhte Luftfeuchte, die im mittleren einstelligen %-Bereich liegt. Während der Messkampagne ③ liegen die Messwerte des DWD und der Sensoren im Stall auf einem ähnlichen Niveau.

Tabelle 3.8: Mittlere Luftfeuchte der Wetterstation und LUBRATEC SMARTBOXEN für die durchgeführten Messkampagnen

in %	$\bar{\varphi}_{\text{DWD}}$	$\bar{\varphi}_{\text{SO}}$	$\bar{\varphi}_{\text{SW}}$	$\bar{\varphi}_{\text{NO}}$	$\bar{\varphi}_{\text{NW}}$	$\bar{\varphi}_{\text{MSO}}$	$\bar{\varphi}_{\text{MSW}}$	$\bar{\varphi}_{\text{MNO}}$	$\bar{\varphi}_{\text{MNW}}$
①	52,00	56,09	56,33	58,36	57,48	60,00	59,69	57,74	56,87
②	78,63	84,19	84,44	n. v.	85,66	n. v.	84,11	n. v.	n. v.
③	88,67	88,50	89,37	87,46	85,77	n. v.	n. v.	n. v.	n. v.

3.4.5 Methangehalt

Die Messfrequenz der Methansensoren ist auf 1 Hz eingestellt. Abbildung 3.10 zeigt die Liniendiagramme der ausgewerteten Methansensoren über alle Messkampagnen anhand des gleitenden Mittels über fünf Minuten (links) sowie den zugehörigen Boxplot (rechts). Die Auswertung der Luv-Seite weist gegenüber der Lee-Seite über alle Messkampagnen hinweg stärkere Streuungen des Messwertes auf, was sich in den größeren Standardabweichungen in Tabelle 3.9 zeigt und an den größeren Antennen der Boxplots in Abbildung 3.10 ablesen lässt.

Zu Beginn der Messkampagne ② musste ein defekter Methansensor an der Nordseite ausgetauscht werden. Die Platine des Ersatz-Sensors ist zusätzlich mit einem Spannungsregler versehen, woraus ein unterschiedlicher Referenzwiderstand \bar{R}_0 zwischen dem Sensor der Nord- und Südseite resultiert. Daher beginnt die Aufzeichnung der Messwerte für die Methankonzentration an der Nordseite der Messkampagne ② um 15:00 Uhr.

Die Veränderung des gemessenen Widerstands zwischen 12:00 Uhr und 12:30 Uhr an der Südseite der Messkampagne ② fällt mit einem einsetzenden Niederschlag und daraus resultierender veränderter Luftfeuchte in Bodennähe zusammen. Abgesehen vom ausgetauschten Sensor wurden die Referenzwiderstände jeweils vor Messbeginn der Kampagne an frischer

Luft und nach einer Aufheizphase von einer Stunde ermittelt, bevor die Sensoren am Messstandort montiert wurden. Anhand der Referenzwiderstände und der Mittelwerte der Sensorwiderstände wurden die mittleren Konzentrationen Methan in ppm berechnet.

Tabelle 3.9: Gemittelte Messwerte der Methansensoren für alle Messkampagnen

Kampagne	\bar{R}_0 in Ω		\bar{R}_S in Ω		σ_{R_S} in Ω		\bar{c}_{CH_4} in ppm	
	N	S	N	S	N	S	N	S
①	20.060	20.060	20.060	17.016	± 2.341	± 912	0,99	18,43
② ^a	12.080	16.220	10.231	16.054	± 400	± 1.599	18,67	3,11
③	16.050	16.050	13.915	16.007	± 721	± 1.125	14,93	2,87

^aUnvollständiger Datensatz durch Sensorausfall.

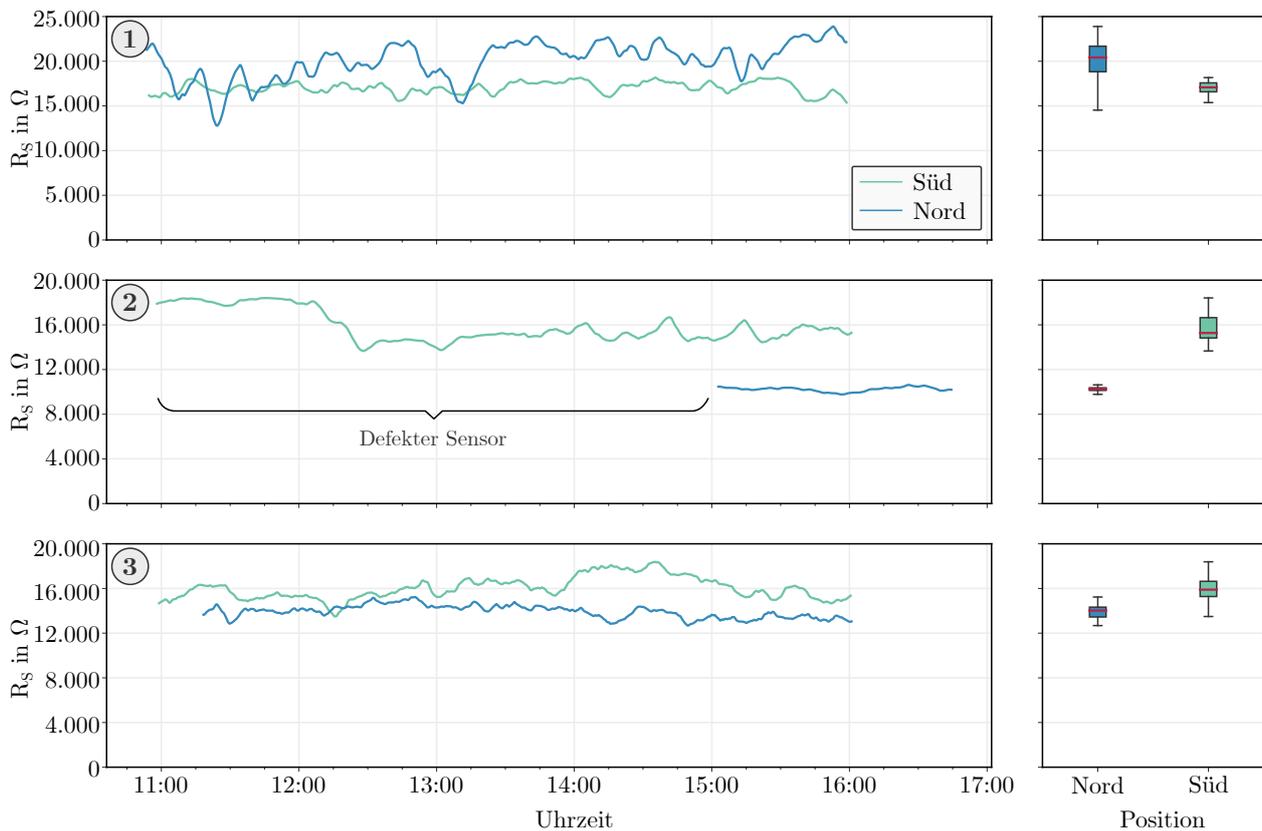


Abbildung 3.10: Auswertung der Methansensoren in einem linksseitig dargestellten Liniendiagramm und rechtsseitigen Boxplot (eigene Darstellung)

Numerische Ergebnisse

Ein Ziel der vorliegenden Arbeit besteht darin, ein verifiziertes und validiertes Simulationsmodell aufzubauen, um auf dieser Grundlage eine Analyse der Parametereinflüsse durchzuführen und die gewonnenen Daten zur Erstellung eines Metamodells zu verwenden. Um eine statistische Versuchsplanung durchführen zu können, ist es notwendig, die Parameter zu definieren, die in den verschiedenen Versuchsreihen variiert werden sollen. Daher beschreibt Kapitel 4.1 die Festlegung der geometrischen Parameter und die Entwicklung eines parametrisierbaren CAD-Modells. In den Kapiteln 4.2 bis 4.4 wird die Diskretisierung des Modells dokumentiert und die Randbedingungen sowie Einstellungen des Löser erläutert. Kapitel 4.5 schätzt die Fehler und Unsicherheiten des Simulationsmodells ab, bevor in Kapitel 4.6 die Systematik zur Untersuchung des Parametereinflusses beschrieben wird.

4.1 Parametrisierbares CAD-Modell

Das vorliegende Kapitel beschreibt die Erhebung der geometrischen Parameter, die Modellierungsstrategie sowie die resultierende Geometrie des parametrisierbaren CAD-Modells. In Teilen basieren die nachfolgenden Ausführungen auf der Arbeit von SCHMEING (2021).

4.1.1 Geometrische Parameter

Die Identifikation der geometrischen Parameter erfolgte mittels Besichtigungen von vier Höfen und einem Experten-Interview (s. Anhang B). Basierend darauf wurde ein Standard-Stall abgeleitet, der in Abbildung 4.1 dargestellt ist. Der parametrisierbare Stall variiert in der Breite zwischen min. 24 m und max. 60 m und in der Länge zwischen min. 30 m und max. 120 m. Diese Werte stellen keine Extreme in der Auslegung dar, sondern sind typische Grenzwerte bei Standardbauweisen. Die Traufhöhe eines frei belüfteten Kuhstalls liegt üblicherweise zwischen min. 2,7 m und max. 5,0 m, woraus sich mit Angabe des Dachneigungswinkels (min. 15° und max. 25°) die Gebäudehöhe ergibt. Für die Lüftungselemente können Axial-, Deckenventilatoren und Schlauchlüftungen installiert werden. Ergänzend ist die Wickellüftung geteilt und kann nach oben und unten verfahren werden, um die Querbelüftung zu steuern. Der Wert für die Wickellüftungen ist mit je 90 % limitiert, um zu berücksichtigen, dass die aufgewickelten Windschutznetze die Querbelüftung versperren können. Ein externer Parameter stellte sich durch häufig im Umfeld der Stallungen errichtete Nachbargebäude dar, woraus sich ein Faktor für die Entfernung zu diesem Gebäude ergab. Da diese Gebäude üblicherweise ebenfalls Stallungen sind, wurden die Abmessungen im Rahmen der Arbeit mit dem Hauptgebäude identisch ausgelegt. Abbildung 4.1 kennzeichnet die geometrischen Parameter des CAD-Modells für die Arbeit.

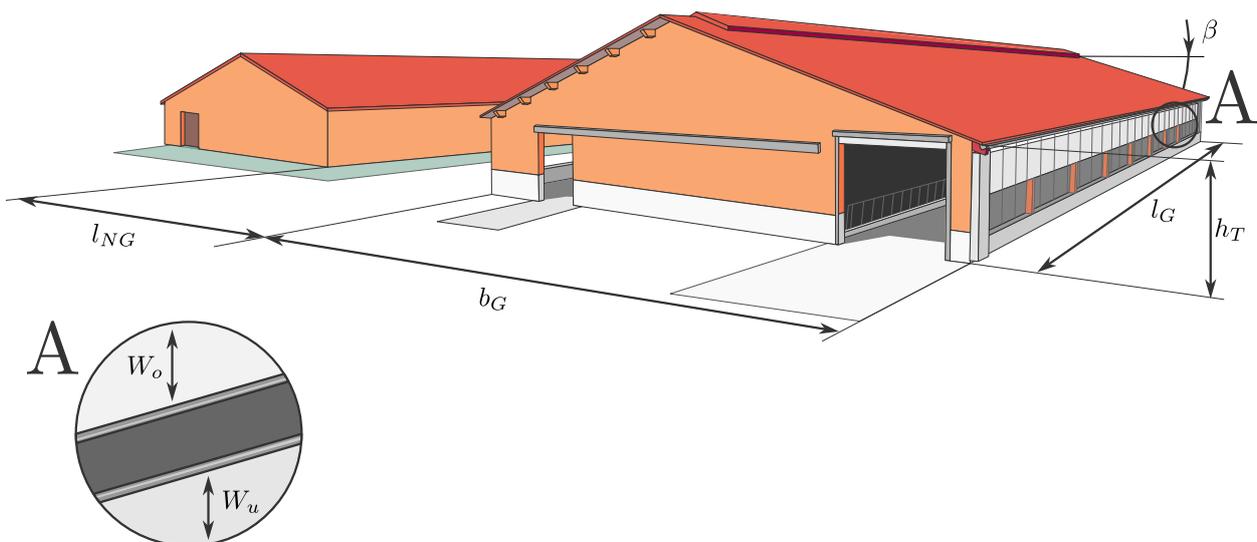


Abbildung 4.1: Ein frei belüfteter Kuhstall und das Nachbargebäude mit markierten geometrischen Parametern (eigene Darstellung)

Die in Tabelle 4.1 zusammengefassten geometrischen Parameter wurden im CAD-Programm als variable Eingangsgröße angelegt, sodass bei Änderung der Parameter ein automatisierter Neuaufbau des CAD-Modells durchgeführt wurde, inklusive der regelbasiert gemusterten Geometrien von bspw. Stützen und Lüftungselementen, die an die geänderten Parameter angepasst wurden. Die spezifische Beschreibung der Geometrien und Musterung einzelner Elemente des CAD-Modells folgt in Kapitel 4.1.3.

Tabelle 4.1: Geometrische Modellparameter für den Versuchsplan

Parameter	Formelzeichen	Grenzwert hoch (+)	Grenzwert niedrig (–)
Entfernung Nachbargebäude	l_{NG}	15 m	nicht vorhanden
Axialventilatoren	V_A	vorhanden	nicht vorhanden
Deckenventilator	V_D	vorhanden	nicht vorhanden
Schlauchlüftung	V_S	vorhanden	nicht vorhanden
Traufhöhe	h_T	5,0 m	2,7 m
Dachneigungswinkel	β	25°	15°
Gebäudelänge	l_G	120 m	30 m
Gebäudebreite	b_G	60 m	24 m
Wickellüftung oben	W_o	90 % offen	vollständig geschlossen
Wickellüftung unten	W_u	90 % offen	vollständig geschlossen

4.1.2 Modellierungsstrategie

Beim Aufbau des Modells wurde die sog. *Resilient Modelling* Strategie angewendet (GEBHARD, 2013). Diese Strategie erfordert, dass die *Features* des Modells nach einer vorgegebenen Ordnerstruktur einsortiert werden. Dies gewährleistet einen konsistenten und vom Modell unabhängigen Aufbau. CAMBA, CONTERO und COMPANY (2016) weisen in mehreren Experimenten eine Zeitersparnis des *Resilient Modelling* gegenüber anderen Modellierungsstrategien für parametrisierte CAD-Modelle nach.

Abbildung 4.2 stellt die beim *Resilient Modelling* einzuhaltende Ordnerstruktur dar. Referenzobjekte, wie z. B. Bilder, Linien, Punkte, Skizzen, Ebenen oder Koordinatensysteme, bilden die Grundlage des Modells und werden im Ordner *1-Referenz* angelegt. 3D-Kurven oder Oberflächen sind darauf aufbauend dem Ordner *2-Konstruktion* zuzuweisen. Folgend werden die *3-Kernelemente* des Modells mittels Extrudieren oder Rotieren der Konstruktionselemente erstellt. Vervollständigende Details der Kernelemente können in der Gruppe *4-Detail* hinterlegt werden und z. B. Löcher oder Gewinde beinhalten. Die Elemente der Gruppe *4-Detail* dürfen mit den Elementen anderer Gruppen, nicht aber innerhalb der Gruppe selbst referenziert werden. Das Mustern oder Spiegeln der Elemente wird als Modifikation in Gruppe *5-Modifizieren* einsortiert. Abschließend können flüchtige Konstruktionsmerkmale in der Gruppe *6-Quarantäne* angewandt werden. Darunter fallen bspw. Fasen oder Radien.

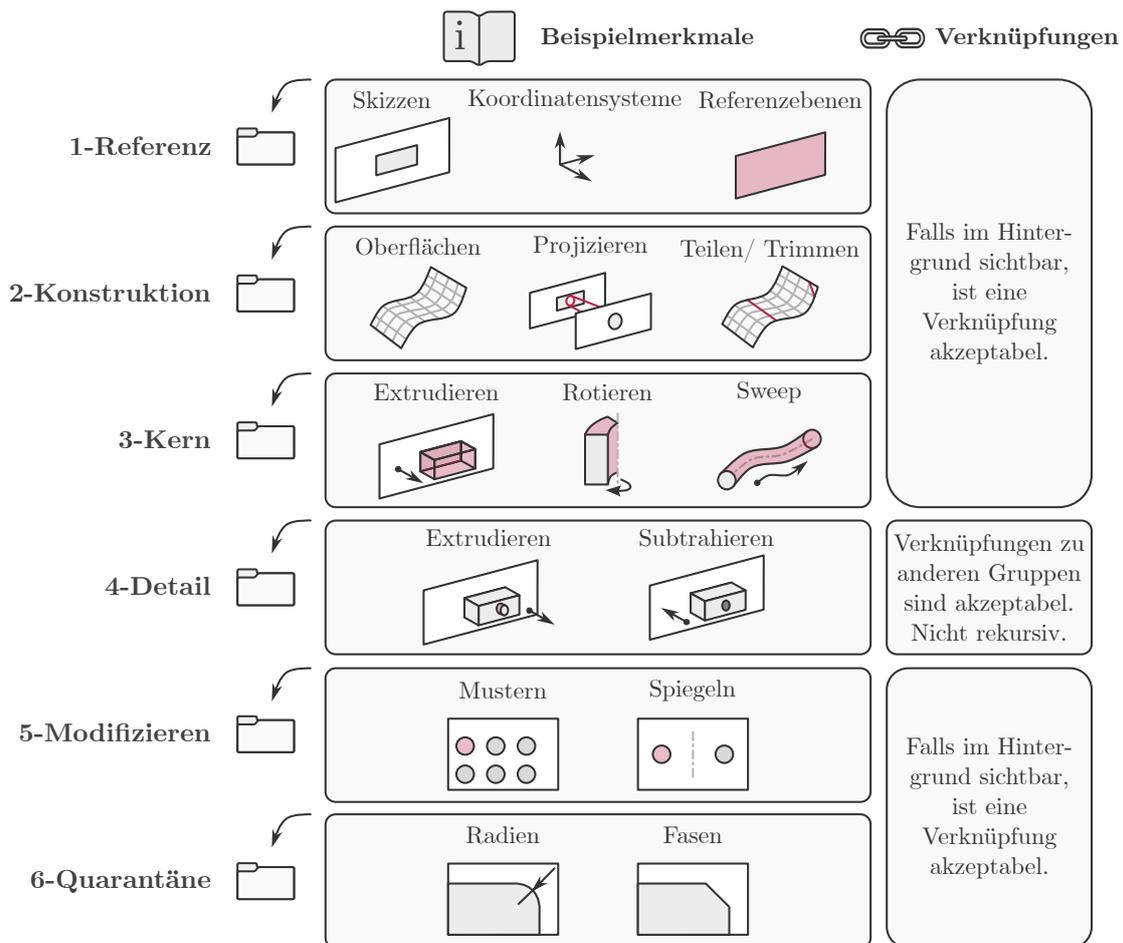


Abbildung 4.2: Gruppierung beispielhafter Konstruktionselemente und deren Verknüpfung gemäß *Resilient Modeling* Strategie (eigene Darstellung)

4.1.3 CAD-Geometrien

Die Modellierung der Geometrie für CFD-Simulationen unterliegt der Besonderheit, dass der vom Medium durchströmte Raum, nicht aber die Festkörper zu modellieren sind (Energiedurchgänge zu Festkörpern ausgenommen). In diesem Fall gilt es daher, einen Strömungsraum um das zu betrachtende Gebäude anzulegen, der in den Abmessungen ausreichend groß ist, damit eine Beeinflussung der Strömung durch die äußeren Begrenzungsflächen möglichst ausgeschlossen werden kann. Die von TOMINAGA et al. (2008) entwickelte AIJ-Richtlinie stellt in sieben Windkanalversuchen und anschließendem Abgleich mit CFD-Simulationen heraus, dass bei der Simulation eines einzelnen Gebäudes ein Abstand zu oberer und lateraler Begrenzung von $5h_G$ einzuhalten ist, wobei h_G der Höhe des Gebäudes entspricht. Damit der infolge der Grenzschichtablösungen auftretende Nachlauf hinter dem Gebäude ausreichend berücksichtigt werden kann, sollte die Begrenzungsfläche am Auslass des simulierten Gebietes $10h_G$ hinter dem Gebäude liegen. FRANKE (2006) gibt mit einem Wert von $15h_G$ einen konservativeren Vorschlag zur Auslegung des Abstands zum Auslass, wohingegen

YOSHIE, MOCHIDA und TOMINAGA (2006) bei zu großer Dimensionierung des Strömungsraumes unrealistische Ergebnisse erzielten. Benachbarte Gebäude im Umkreis eines Radius von h_G bis $2h_G$ sind ebenfalls detailliert zu modellieren. Gebäude, die sich außerhalb dieses Umkreises, jedoch innerhalb des Strömungsraumes befinden, können mittels vereinfachter Geometrie (z. B. Quader, Zylinder o. Ä.) oder über eine Oberflächenrauheit berücksichtigt werden. Abbildung 4.3 stellt die Abstände im Strömungsraum gemäß der Richtlinie dar. Die Platzierung des Stalls in der Ecke der Strömungsdomäne erlaubt es, den Einlass in der CFD-Simulation auf zwei unterschiedliche Flächen zu setzen (hier Süd und West), ohne die Geometrie zu verändern oder die Vernetzung des Modells erneut durchführen zu müssen.

Der äußere Strömungsraum bildet das erste zentrale Konstruktionselement. Hinsichtlich einer geeigneten Diskretisierung des Strömungsraumes für die CFD-Simulation ist der äußere Strömungsraum im Umfeld des Gebäudes unterteilt worden, indem ein Quader um das Stallgebäude gelegt wurde, dessen Abstand zu den äußeren Abmessungen des Gebäudes h_G betrug. Der durchströmte Raum im Inneren des Stalls stellt das letzte Kernelement des CAD-Modells dar. Abbildung 4.3 visualisiert die drei Kernelemente separiert und kombiniert.

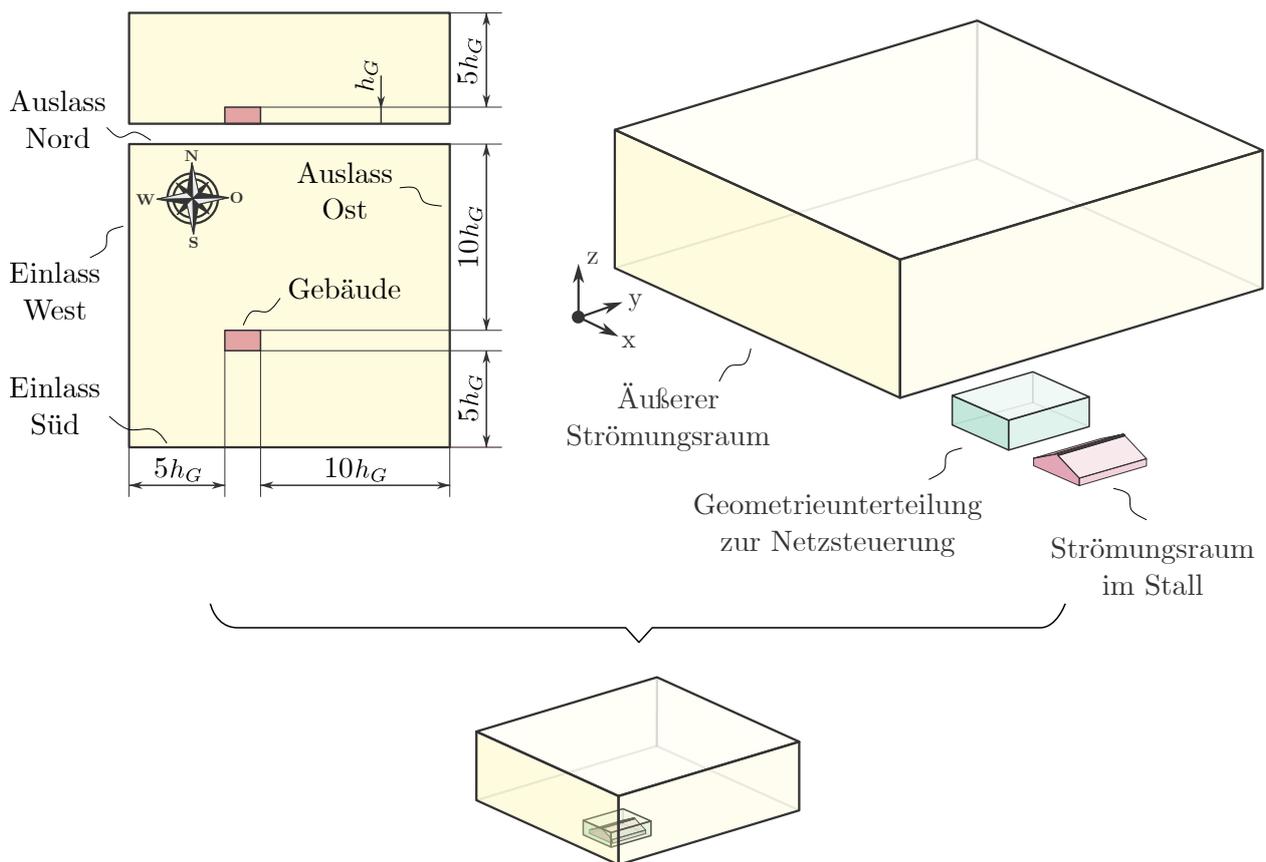


Abbildung 4.3: Darstellung des Strömungsraumes zur CFD-Simulation mit Gebäuden gemäß TOMINAGA et al. (2008) und Anwendung der Richtlinie auf die Kernelemente des CAD-Modells bei mittlerer Einstellung der geometrischen Parameter (eigene Darstellung)

Der innere Strömungsraum ist durch die Öffnungen der Wickellüftung, Firstlüftung, Tore oder Türen mit dem äußeren Strömungsraum verbunden. Die Firstlüftung, Tore und Türen werden durch die landwirtschaftlich Beschäftigten i. d. R. nicht verändert und sind daher als konstant angenommen und modelliert. Die Tore ermöglichen den Zugang zur Futterplatte, z. B. für mobile Fütterungssysteme. Kuhställe können in ihrer Bauweise nach außen oder innen liegenden Futterplatten unterschieden werden. Bei außen liegender Futterplatte ergibt sich ein Gehege, das von beiden Futterplatten zugänglich ist, wohingegen die innen liegende Futterplatte den Stall in zwei Gehege teilt. Für die vorliegende Arbeit wurde das Modell auf außen liegende Futterplatten beschränkt. Da jede Futterplatte von beiden Stirnseiten des Gebäudes zugänglich ist, ergeben sich daraus vier Tore, die, abgesehen von stürmischen Ereignissen, ganzjährig geöffnet sind. Die gewählten Abmessungen wurden gemäß der in Abbildung 4.4 dargestellten Skizze parametrisiert im Modell implementiert. Die Bezugsebene der Tore ist die hier linksseitig dargestellte Außenwand des Stalls. Die gegenüberliegenden Tore sind durch Spiegel des Referenz-Tors erzeugt.

In den Ställen sind oft Büroräume, Lagerräume, Melkanlagen etc. untergebracht. Diese Funktionsräume und -einheiten sind üblicherweise an einer der Stirnseiten des Gebäudes platziert und sind über einen separaten Eingang von außen zugänglich. Für die vorliegende Arbeit wurden die komplexeren Geometrien dieser Bauteile vereinfacht und zusammengefasst, um eine homogene Diskretisierung im Stall zu gewährleisten. Hierfür diente ein real existierender Stall als Vorbild (s. auch Kapitel 3). Abbildung 4.4 zeigt dies an der positiven Stallgeometrie auf, die später vom Strömungsraum subtrahiert wurde. Die Höhe der Räume wurde auf 3 m festgelegt.

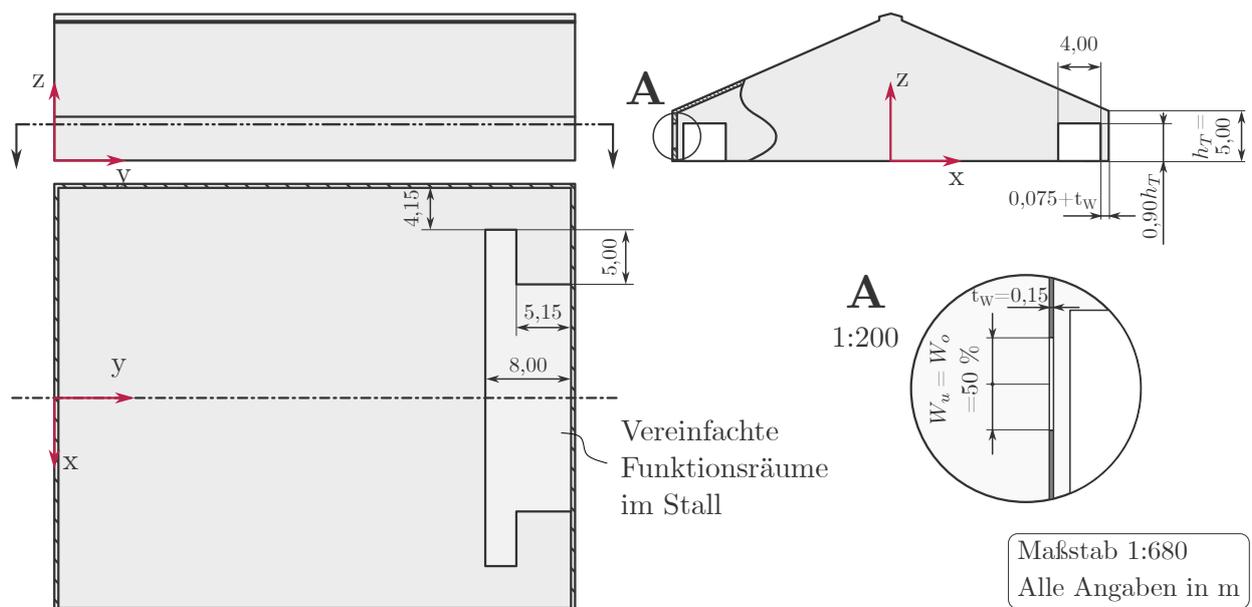


Abbildung 4.4: Teilweise Schnittdarstellung des Stalls mit den Abmessungen der parametrisierten Geometrien und Funktionsräume (eigene Darstellung)

Des Weiteren ist der innere Strömungsraum bei geöffneten Wickellüftungen mit dem äußeren Strömungsraum verbunden. Die Wickellüftungen sind mittig zwischen der unteren Begrenzungsfläche und der Traufhöhe geteilt. Über eine prozentuale Angabe der Versperung können die untere und obere Wickellüftung separat eingestellt werden. Abbildung 4.4 zeigt dies an der vom Strömungsraum zu subtrahierenden Positiv-Geometrie für je zu 50 % geöffnete Wickellüftungen auf. Die Wandstärke der Wickellüftungen entspricht $t_W = 0,15$ m und ist identisch mit den Wänden des Gebäudes. Dies reduziert kleinere geometrische Stufen im Modell, um eine bessere Diskretisierung zu ermöglichen. Die Referenzebene der Wickellüftungen ist an der äußeren Wandfläche positioniert. Durch ein Spiegeln der Geometrie an der z - y -Ebene wurden die gegenüberliegenden Wickellüftungen erzeugt.

Die Statik des Gebäudes ist zudem durch eine Balkenkonstruktion verstärkt. Die Abmessungen der Balken sind ausreichend groß, um eine Beeinflussung der Strömung anzunehmen und diese in der CFD-Simulation zu berücksichtigen. Damit eine Änderung der äußeren Abmessungen des Stalls weiterhin ein automatisiert neu aufgebautes Modell erzeugt, sind die Stützen zunächst auf einer Ebene skizziert, extrudiert und abschließend gemustert. Abbildung 4.5 zeigt die Abmessungen der Stützkonstruktion und Musterung auf. Die Musterung erfolgte über die Vorgabe des Abstandes der Stützen (analog zum Bauplan des Versuchstalls). Die Anzahl der Stützen ergab sich aus $n_{S,x} = \text{floor}(\frac{b_G}{2 \cdot \Delta x})$ bzw. $n_{S,y} = \text{floor}(\frac{l_G - 2t_W}{\Delta y})$, wobei $\Delta x = 11,0$ m bzw. $\Delta y = 7,5$ m.

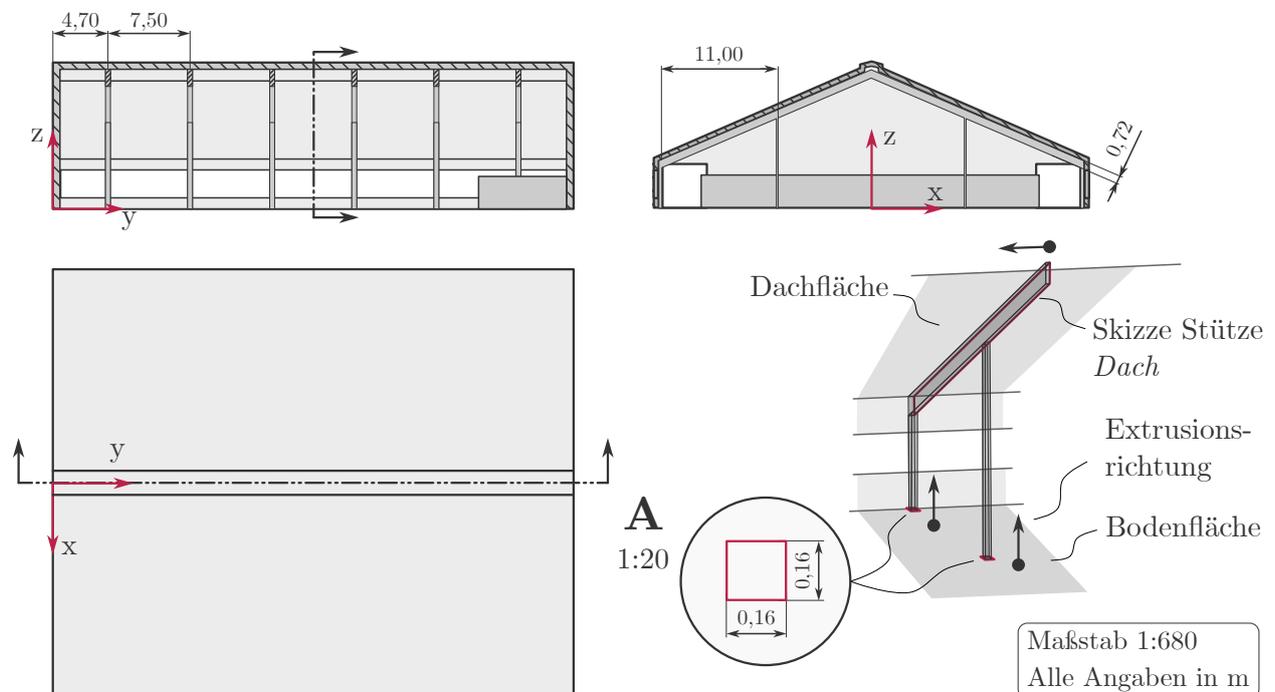


Abbildung 4.5: Zeichnung des Stalls in Schnitt- und Vollarstellungen, inkl. der Detailbetrachtung der Skizzen für die Extrusion und Musterung der Stützen im CAD-Programm (eigene Darstellung)

Das Gehege wurde im CAD-Modell unterteilt, um eine separate Steuerung des Rechnetzes zu ermöglichen. An den zur Futterplatte liegenden Begrenzungsflächen des Geheges befindet sich in realen Ställen ein Betonsockel, der die Strömung beeinflusst und daher im Modell als Quader vereinfacht implementiert wurde. Aufgrund des genutzten CFD-Programms konnten die im Gehege stehenden und liegenden Kühe nicht über ein anisotrop poröses Material approximiert werden, da die Turbulenzinteraktion zwischen porösem Material und freier Strömung lediglich für sehr einfache Turbulenzmodelle implementiert ist. Stattdessen wurde die nach MONDACA und CHOI (2016) vorgeschlagene Approximation als Zylinder vorgenommen. Über das Jahr gemittelt kann für den modellierten Stall eine Herde von 120 Kühen angenommen werden. Die meisten stehenden Kühe sind in Richtung der Futterplatten orientiert oder liegen in selbiger Ausrichtung in den Liegeboxen (s. auch Anhang C). Anderweitige Orientierungen wurden nicht berücksichtigt, auch wenn diese bei zeitlich bedingter Positionsveränderung der Kühe vereinzelt vorkommen. Die Kühe wurden im CAD-Modell äquidistant gemustert. Abbildung 4.6 stellt das Gehege, den Betonsockel, die Geometrie und Musterung der Kühe im Stall dar. Die Liegeboxen und Gestänge im Gehege wurden als Geometrie nicht in das CAD-Modell aufgenommen. Die Anzahl der zu mustern Kühe an der Futterplatte ergab sich in y -Richtung zu $n_{K1,y} = \text{floor}(\frac{l_G - 10,15 \text{ m}}{\Delta y})$, wobei $\Delta y_{K1} = 1,875 \text{ m}$. Eine Spiegelung an der z - y -Ebene vervollständigt die Platzierung der außen stehenden Kühe. Für die Anzahl der im Gehege liegenden Kühe ergaben sich für die y -Achsenrichtung $n_{K2,y} = \text{floor}(\frac{l_G - 10,15 \text{ m}}{\Delta y})$, mit $\Delta y_{K2} = 2,50 \text{ m}$ und für die symmetrische Musterung der zwei mittig gruppierten Kühe entlang der x -Achsenrichtung ergab sich die Anzahl zu $n_{K2,x} = \text{floor}(\frac{b_G - 5,00 \text{ m}}{\Delta x})$, wobei $\Delta x_{K2} = 7,50 \text{ m}$.

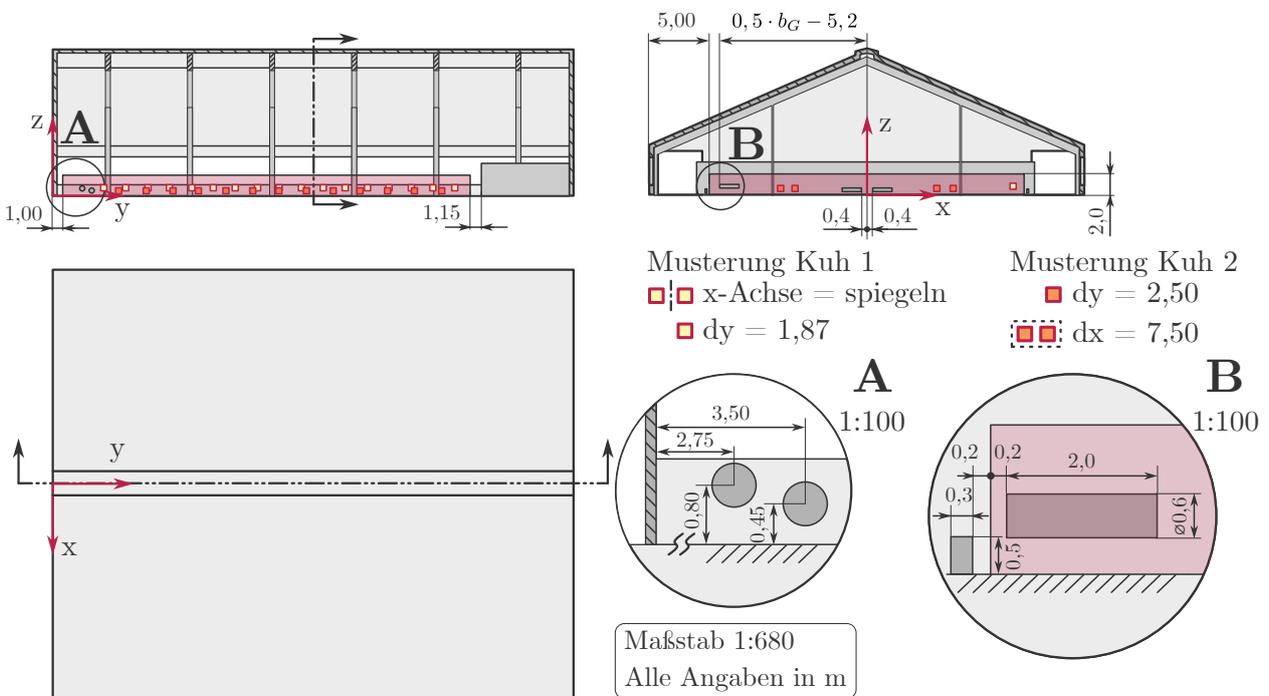


Abbildung 4.6: CAD-Geometrien des Stalls, inkl. Detailbetrachtungen der Skizzen zur Extrusion und Musterung der Kühe (eigene Darstellung)

Bei den Lüftungselementen wurden drei häufig verbaute Systeme für die frei belüftete Milchviehhaltung in die Untersuchung aufgenommen. Die Axialventilatoren, Deckenventilatoren und Schlauchbelüftungssysteme sind regelbasiert im CAD-Modell implementiert. Die Axialventilatoren und der Deckenventilator sind als Zylinder vereinfacht, um dort die Randbedingungen in der CFD-Simulation aufzuprägen. Die Schlauchbelüftung wurde als Zylinder konstruiert, mit abgesetzter Fläche für die Öffnung. Abbildung 4.7 stellt die Skizzen der Systeme dar, die wie gekennzeichnet im Stall platziert wurden. Die um 15° geneigten Axialventilatoren sind entlang der y -Achse mit der Anzahl $n_{Ax} = \text{floor}(\frac{l_G - 2t_w - 8,00 \text{ m}}{\Delta y})$ gemustert und an der Symmetrieebene $z-y$ gespiegelt, wobei $\Delta y_{Ax} = 6,00 \text{ m}$. Die Axialventilatoren schließen an der Oberkante mit der Traufhöhe h_T ab. Der Durchmesser des Deckenventilators entspricht $\varnothing d = 7,00 \text{ m}$ und wurde in einer Höhe von $h_D = h_G - 4,00 \text{ m}$ als einzelnes Lüftungselement zentral über dem Gehege positioniert, wobei h_G die Gesamthöhe des Gebäudes ist. Die Schlauchbelüftungen wurden auf einer Höhe von $z = 5,50 \text{ m}$ positioniert und über eine Länge von $l_S = l_G - 12,00 \text{ m}$ extrudiert. Die Musterung der Belüftungsschläuche ist in x -Achsenrichtung ausgelegt und die Anzahl ergab sich zu $n_S = \text{floor}(\frac{b_G - 10,00 \text{ m}}{2 \cdot \Delta x})$, wobei $\Delta x = 8,00 \text{ m}$ entsprach. Die Musterung der Belüftungsschläuche wurde an der $z-y$ -Ebene gespiegelt.

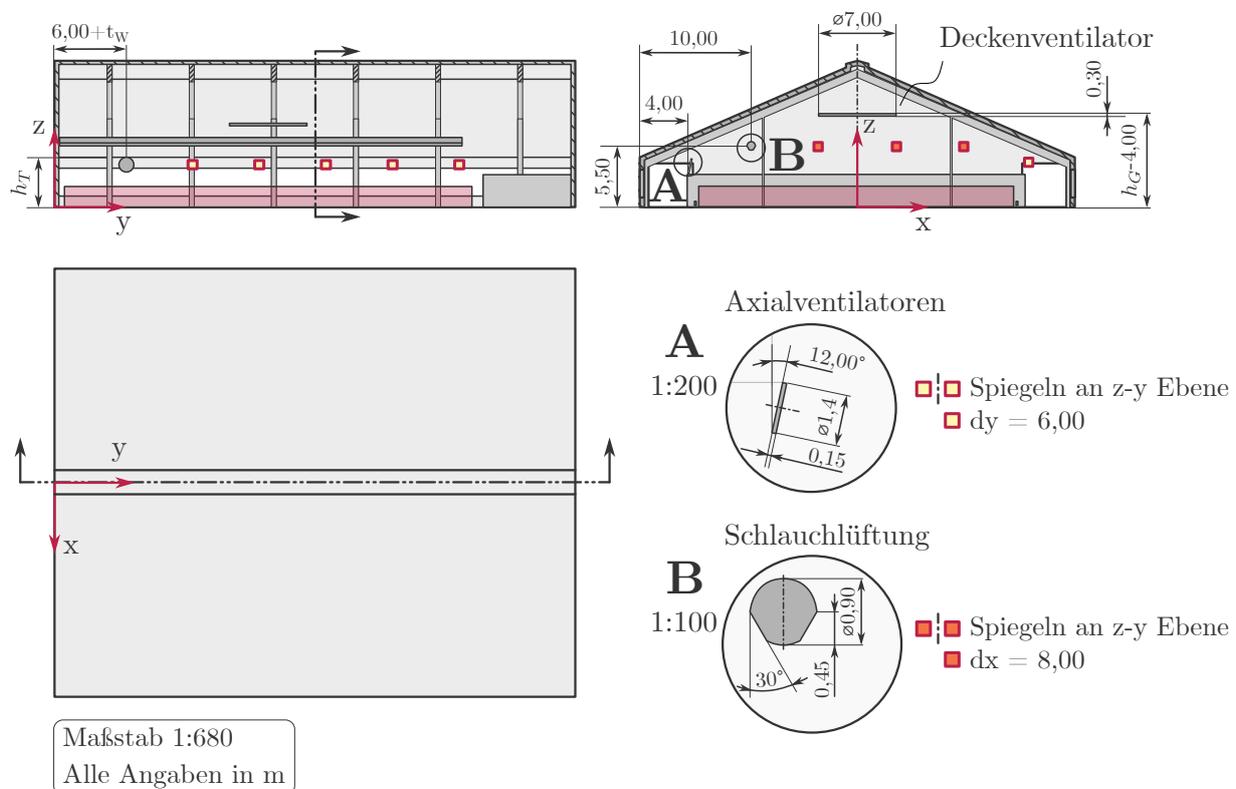


Abbildung 4.7: Schnitt- und Vollendarstellung des Stalls, inkl. Detailbetrachtung der implementierten Lüftungselemente und deren Musterung (eigene Darstellung)

4.2 Diskretisierung

Die Diskretisierung des Strömungsraumes wurde basierend auf den Richtlinien von TOMINAGA et al. (2008) und FRANKE und BAKLANOV (2007) durchgeführt (AIJ und COST Action 732). Daraus ergab sich, dass bei komplexeren Geometrien die Verwendung von unstrukturierten Netzen legitim ist. Hierbei gilt es zu beachten, dass die Randschichtvernetzung in Prismen erfolgt. Auf den Prismen können direkt die im Inneren des Strömungsraumes verwendeten Tetraeder angeordnet werden und am seitlichen Rand der Prismen mittels Pyramiden ein Übergang generiert werden.

Für die optimale Netzsteuerung im Umfeld des Gebäudes empfiehlt sich nach FRANKE und BAKLANOV (2007) die Kantenlänge der Elemente an den Gebäudewänden auf 3 % der Gebäudehöhe einzustellen. Da im Rahmen der Verifizierung drei unterschiedliche Netzfeinheiten mit möglichst isotroper Netzverfeinerung anzulegen sind, ergeben sich daraus unterschiedliche Einstellungen der Elementkantenlängen auf den Wänden für die drei Varianten, die mit $h_{max\ 1,2,3} \in \{0,2; 0,35; 0,85\}$ m eingestellt sind und 1,5 %, 2,6 % und 6,3 % der Höhe des Gebäudes entsprechen. Die maximalen Elementkantenlängen im äußeren Volumen sind mit $h_{max\ 1,2,3} \in \{7,0; 10,0; 22,0\}$ m limitiert. Die maximale Element-Wachstumsrate ist in der Randschicht über fünf Elemente auf $G_{\partial} = 1,20$ und im verbleibenden Volumen auf $G_V = 1,15$ begrenzt. Daraus ergibt sich die in Abbildung 4.8 dargestellte grobe, in Abbildung 4.9 dargestellte mittlere und in Abbildung 4.10 dargestellte feine Diskretisierung.

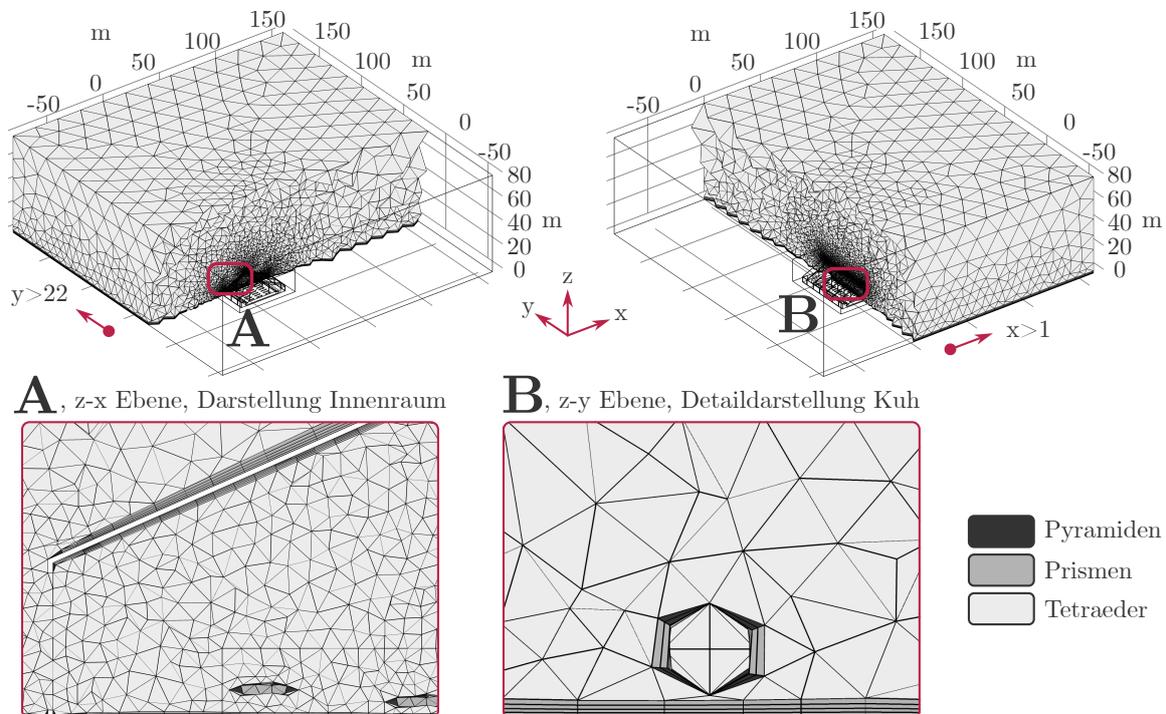


Abbildung 4.8: Grobe Diskretisierung des Strömungsraumes und der Stallgeometrie gemäß FRANKE und BAKLANOV (2007). Ansicht der z-x und z-y Schnittebenen bei $y > 22,0$ m und $x > 1,0$ m (eigene Darstellung)

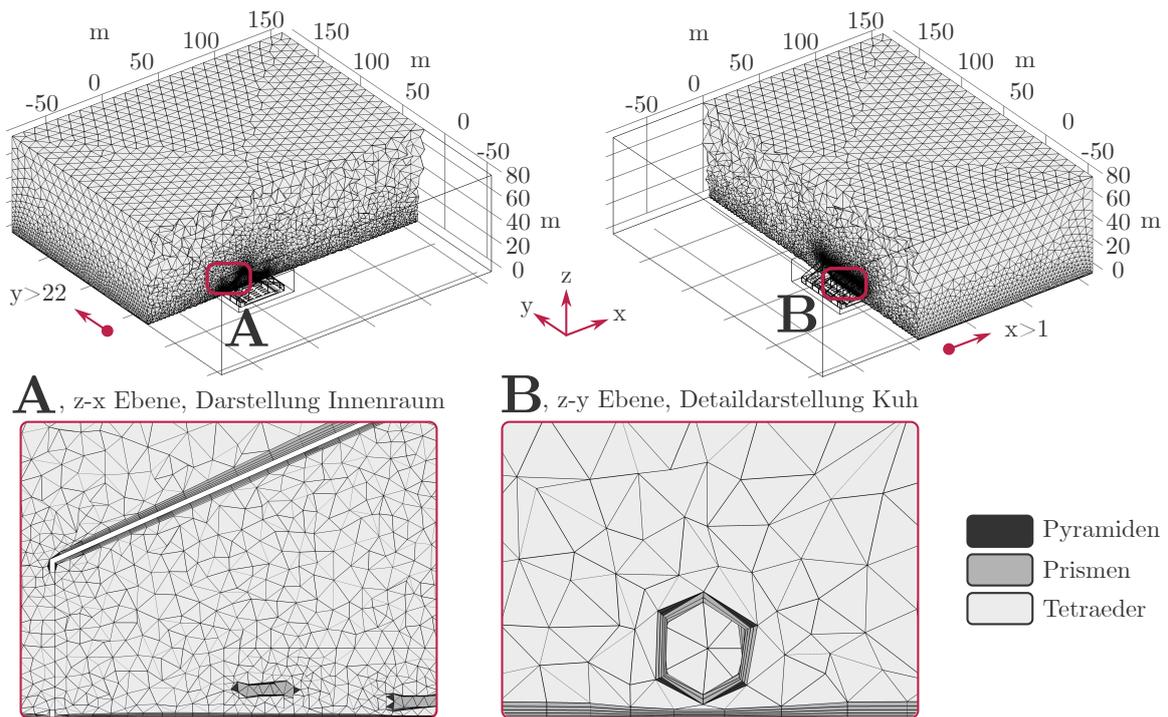


Abbildung 4.9: Mittlere Diskretisierung des Strömungsraumes und der Stallgeometrie gemäß FRANKE und BAKLANOV (2007). Ansicht der $z-x$ und $z-y$ Schnittebenen bei $y > 22,0$ m und $x > 1,0$ m (eigene Darstellung)

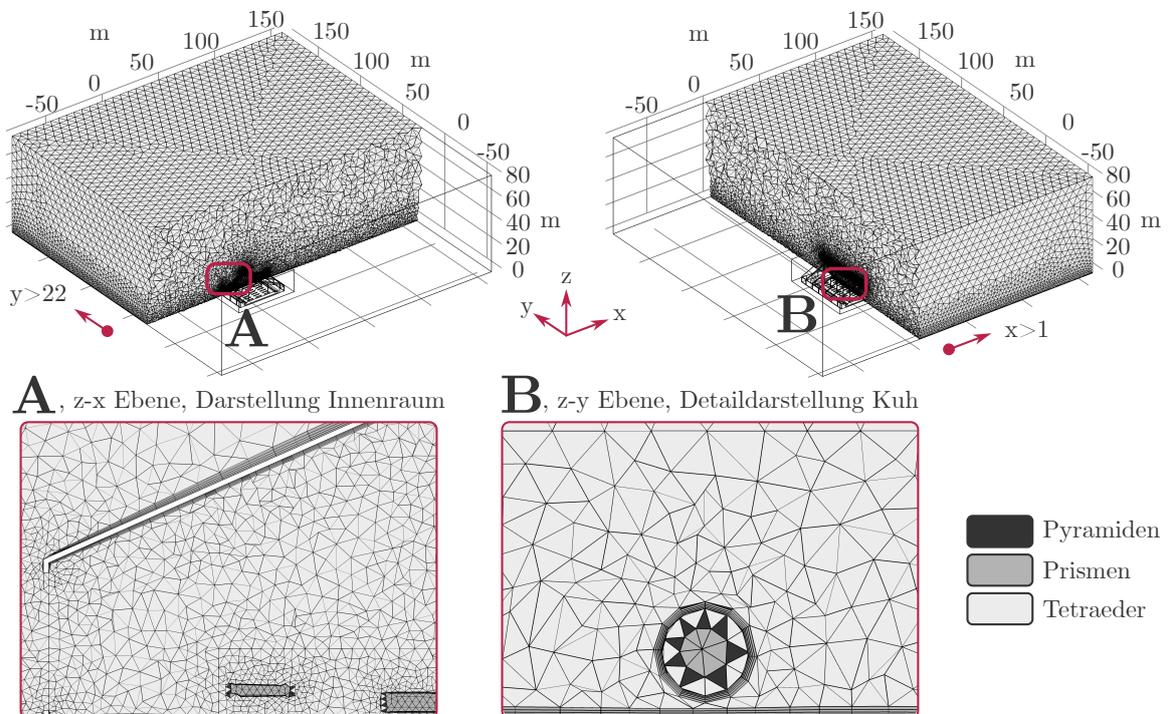


Abbildung 4.10: Feine Diskretisierung des Strömungsraumes und der Stallgeometrie gemäß FRANKE und BAKLANOV (2007). Ansicht der $z-x$ und $z-y$ Schnittebenen bei $y > 22,0$ m und $x > 1,0$ m (eigene Darstellung)

Die Anzahl der resultierenden Elemente der unterschiedlichen Netzfeinheiten ist mit dem Gros der Arbeiten im Kapitel 2.5.1 vergleichbar und liegt für die Konstellation der geometrischen Parameter des Versuchsstalls bei $N \in \{1.206.578; 2.694.251; 5.980.829\}$.

Die Elementqualitäten der unterschiedlich feinen Diskretisierungen sind mit Bewertung der Schiefe durchschnittlich $\bar{q}_{E1} = 0,676$, $\bar{q}_{E2} = 0,673$ und $\bar{q}_{E3} = 0,667$. Für die Bewertung nach maximalem Winkel ergeben sich $\bar{q}_{E1} = 0,762$, $\bar{q}_{E2} = 0,761$ und $\bar{q}_{E3} = 0,757$. Wobei \bar{q}_{E1} der Qualität des feinen, \bar{q}_{E2} der Qualität des mittleren und \bar{q}_{E3} der Qualität des groben Rechenetzes entspricht. Diese Werte sind normiert und für das verwendete Programm gilt ein Wert von 1,0 als ideales Netz.

4.3 Randbedingungen

Fluidtransport

Die Randbedingung der numerischen Simulation des Fluidtransportes ist am Einlass der Strömungsdomäne über eine erzwungene Konvektion angelegt. Aufgrund der Anströmung des Stalls in atmosphärischer Grenzschicht, genauer der bis etwa 100 m geltenden Prandtl-Schicht, ist diese gemäß nationaler Norm für die Betrachtung von Gebäuden bei Windlast generiert (DIN1991-1-4/NA:2010-12, 2010). Daraus resultiert anstelle des üblicherweise vorgeschlagenen logarithmischen Windprofils ein Potenzprofil, mit dem Vorteil der Gültigkeit über die Prandtl-Schicht hinaus ($z_{max} = 300$ m), was bei der Simulation hoher Gebäude und entsprechend hoher Strömungsdomänen relevant sein kann:

$$c_r = \frac{v_m(z)}{v_m(10)} = 0,19 \cdot \left(\frac{z_0}{0,05}\right)^{0,07} \ln\left(\frac{10}{z_0}\right) \left(\frac{z}{10}\right)^{\alpha_W}, \quad (4.1)$$

wobei c_r der Rauheitsbeiwert, $v_m(z)$ die mittlere Windgeschwindigkeit der Höhe z über dem Grund, z_0 die Rauheitslänge und α_W der Profilexponent ist. Bei Annahme der Geländekategorie II für ein „landwirtschaftliches Gebiet“, entspricht die Rauheitslänge $z_0 = 0,05$ m und der Profilexponent $\alpha_W = 0,16$. Die Gültigkeit des Potenzprofils ist erst ab $z_{min} = 4,0$ m gegeben. Bei $z \leq 4,0$ m gilt $v_m = 0,86 \cdot v_b$. Die Basisgeschwindigkeit wurde mit $v_b = v_m(10)$ angenommen und je nach betrachtetem Szenario implementiert.

Ergänzend wurde die Turbulenzintensität am Einlass als Funktion der Höhe nach DIN1991-1-4/NA:2010-12 (2010) berechnet:

$$I_v(z) = 0,19 \cdot \left(\frac{z}{10}\right)^{-0,16}, \quad (4.2)$$

wobei $I_v(z)$ der Turbulenzintensität der Höhe z entspricht. Auch dieses Profil ist für die Geländekategorie II erst ab $z_{min} = 4,0$ m zulässig. Unterhalb dessen gilt $I_v = 0,22$. Die ergänzenden Auslegungen für besondere Geländetopografien waren bei der Lage des Stalls in

ebenem Binnenland gemäß der Norm nicht anzuwenden. Betrachtungen der Böengeschwindigkeit entfielen für die stationären numerischen Simulationen.

Der Auslass der Strömungsdomäne wurde über eine Druckrandbedingung aufgeprägt, indem der statische Druck der Auslassfläche konstant und gleich dem Umgebungsdruck gesetzt wurde, womit eine voll ausgebildete Strömung erzwungen wird.

Die lateralen Begrenzungsflächen entlang der Hauptströmungsrichtung sowie die obere Begrenzungsfläche wurden als glatte Wände ohne Reibung ausgelegt, um der dort in der Realität erweiterten Domäne Rechnung zu tragen. Die Flächen des Stalls, der Kühe, weiterer Gebäudeelemente und der Boden der Strömungsdomäne wurden mit einer reibungsbehafteten Wandfunktion definiert. Diese Funktion approximiert das Geschwindigkeitsprofil in der viskosen Unterschicht bis zum ersten Knoten in \hat{e}_\perp und reduziert so die erforderliche Anzahl der Elemente in der Randschicht normal zur Wand. Die äquivalente Sandrauheit der Wände wurde aus der Rauheitslänge z_0 gemäß FRANKE und BAKLANOV (2007) mit $k_s = z_0 \exp(\kappa \cdot 8,5)$ berechnet. Die reibungsbehafteten Wände im Strömungsraum sind anhand der Rauheitslänge z_0 für einen roh verschalteten Beton mit einer äquivalenten Sandrauheit von $k_s \approx 0,03$ m eingestellt. Gemäß FRANKE und BAKLANOV (2007) belegt ein y^+ -Wert zwischen 30 und 500 die Gültigkeit der verwendeten Wandfunktionen für die Strömungssimulation urbaner Gebiete.

Die Lüftungselemente sind über eine Aus- und Einlassbedingung mit erzwungener Konvektion angelegt. Hierbei kann der Volumenstrom nach Herstellerangabe durch die im Modell erstellte Geometriefläche geteilt und daraus die jeweilige Geschwindigkeit der Randbedingungen ermittelt werden. Für die implementierten Lüftungselemente gibt der Installateur (HUESKER SYNTHETIC) folgende Angaben der oberen Grenzwerte:

- Axiallüfter: $35.000 \frac{\text{m}^3}{\text{h}}$
- Deckenventilator: $500.000 \frac{\text{m}^3}{\text{h}}$
- Schlauchbelüftung: $800 \frac{\text{m}^3}{\text{h}}$ je m Stalllänge

Energietransport

Am Einlass der Strömungsdomäne wurde dem Medium Luft eine für die Umgebung konstant vorherrschende Temperatur T zugewiesen. Die lateralen Begrenzungsflächen und Gebäudeflächen sowie die obere und untere Begrenzungsfläche wurden als adiabate Wände festgelegt. Entgegen den z. T. detaillierten Betrachtungen von Stall-Segmenten wurde die Radiation für die vorliegende Arbeit vernachlässigt. Dies wurde als zulässig bewertet, da laut GEBREMEDHIN (1987) und MONDACA, ROJANO et al. (2013) die konvektiven Effekte mit einem Anteil von bis zu 80 % dominieren. Dies gilt insbesondere mit zunehmender Windgeschwindigkeit und deutlicher für liegende Kühe. Als Energieeintrag wurden über die Flächen der approximierten Kühe eine maximale Wärmeleistung je Kuh in Anlehnung an GEBREMEDHIN und B. WU (2002) mit $\dot{Q} \approx 1000$ W vorgegeben.

Ergänzende Wärmequellen durch zusätzliche Verbraucher, welche ebenfalls Energie in das System eintragen, sind in der vorliegenden Arbeit nicht berücksichtigt.

Stofftransport

Der durch die Milchviehhaltung erzeugte Ausstoß an Methan (CH_4) lässt sich grundsätzlich in zwei Kategorien einteilen. Ein Teil ist dem Lagersystem des Dungs zuzuordnen, der andere Anteil ergibt sich aus der enterischen Fermentation der Kühe. Für die Berechnung wurde die Richtlinie des *Intergovernmental Panel on Climate Change* (IPCC) herangezogen (EGGLESTON et al., 2006) und gemäß der *Tier 2*-Methode durchgeführt. Mit Annahme der in Anhang E.1 hinterlegten Berechnungen ergaben sich für den Versuchsstall mit durchschnittlich 120 Kühen Besatz:

- Enterische CH_4 -Emissionsquelle: $5,29 \cdot 10^{-04} \frac{\text{kg}}{\text{s}}$
- Dung-Lagersystem-bedingte CH_4 -Emissionsquelle: $2,00 \cdot 10^{-04} \frac{\text{kg}}{\text{s}}$

Die in der Umgebung enthaltene Konzentration von $X_{\text{CH}_4} = 2,0$ ppm (STANLEY et al., 2018) lässt sich gemäß nachstehender Formel für die Simulationsrandbedingung umrechnen:

$$c_{\text{CH}_4} = \frac{0,1 \cdot M \cdot p_{\text{ref}} \cdot X_{\text{CH}_4}}{R_G \cdot T_{\text{ref}}}, \quad (4.3)$$

wobei die Molmasse $M = 16,04 \frac{\text{g}}{\text{mol}}$, der Referenzdruck $p_{\text{ref}} = 1013,25$ mbar, die Referenztemperatur $T_{\text{ref}} = 283,85$ K und die Gaskonstante $R_G = 8,314472 \frac{\text{kJ}}{\text{kmol} \cdot \text{K}}$ mit den entsprechenden Konstanten bzw. den Jahresmittelwerten für den Standort des Versuchsstalls angenommen wurden. Daraus ergab sich für den Einlass der Strömungsdomäne $c_{\text{CH}_4} = 8,587 \cdot 10^{-5} \frac{\text{mol}}{\text{m}^3}$. Der Diffusionskoeffizient von CH_4 in Luft ist gemäß LIDE (2008) mit $D_{\text{CH}_4} = 0,21 \frac{\text{cm}^2}{\text{s}}$ implementiert.

Die am Einlass vorherrschende Luftfeuchte der Umgebung wurde als konstant angenommen. Ergänzend wurde die von der nassen Oberfläche des Dungs emittierte Luftfeuchte über eine Randbedingung am Boden mit 100 % Luftfeuchte approximiert. Die über die Kühe abgegebene Feuchte beträgt laut LANTSOVA, V. VTORYI und S. VTORYI (2015) zwischen $\dot{m} = 8,7$ und $13,4 \frac{\text{kg}}{\text{Tag}}$ und ist von der Raumtemperatur und dem Gewicht der Kühe abhängig. Da der durchschnittliche Besatz im Versuchsstall mit ca. 650 kg Gewicht hoch einzuordnen ist, wurde der nach LANTSOVA, V. VTORYI und S. VTORYI (2015) maximale Wert in die numerische Simulation aufgenommen.

Abbildung 4.11 gibt einen Überblick zu den angelegten Randbedingungen im numerischen Simulationsmodell, bei einer Anströmung mit Westwind (s. auch Abbildung 4.3). Detail A zeigt die angelegten Strömungsrandbedingungen der aktiven Lüftungselemente, während Detail B die Randbedingungen im Gehege sowie für den Wärme- und Stofftransport der Kühe visualisiert.

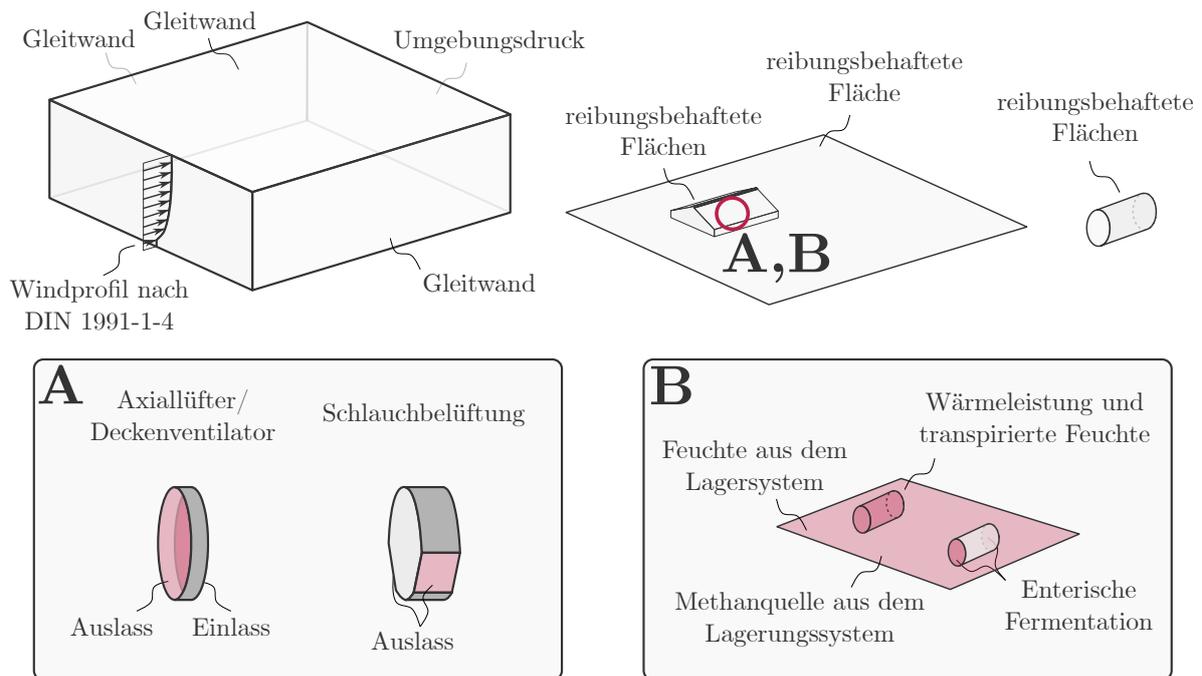


Abbildung 4.11: Darstellung der im numerischen Simulationsmodell angelegten Randbedingungen bei einer West-Ost-Anströmung des Stalls (eigene Darstellung)

4.4 Einstellungen des Löses

Damit das diskretisierte Strömungsgebiet unter Berücksichtigung der beschriebenen Randbedingungen für den Fluid-, Wärme- und Stofftransport gelöst werden konnte, war es abschließend erforderlich, die Einstellungen für den Löser zu definieren.

Zwar können die zeitlichen Fluktuationen des Windes einen Einfluss auf die Strömungsverteilung im Stall haben, dennoch dokumentieren alle Arbeiten im Kapitel 2.5.1, dass RANS-Simulationen im Abgleich mit den Messwerten der Realität gute Übereinstimmungen zeigen. Zudem sind die auszuwertenden Zielgrößen bezüglich des Tierwohls, der optimalen Durchlüftung und des Emissionsstroms über (Teil-)Volumen gemittelte Werte oder auszuwertende Flächenintegrale, die gegenüber diskreten Werten für zeitliche Schwankungen nicht in gleichem Maße anfällig sind. Eine RANS-Simulation ist für die Ziele der vorliegenden Arbeit demnach geeignet, um den Fluid-, Energie- und Stofftransport zu simulieren. Für die RANS-Simulation ist ein Turbulenzmodell erforderlich, dabei ist nach Stand der Forschung insb. das $k-\varepsilon$ -Modell zu beachten. Für die Validierung ist ergänzend das realizable- $k-\varepsilon$ -Modell und das $k-\omega$ -SST-Modell vorgesehen. Für die stationäre Lösung mittels RANS-Simulation wurde ein Pseudo-Zeitschrittverfahren eingestellt, welches sich iterativ der stationären Lösung nähert, indem die Courant-Friedrichs-Lewy-Zahl (CFL-Zahl) vorgegeben wird. Die Schrittweite $CFL = \frac{u \cdot \Delta t}{\Delta x}$ des Verfahrens während des Lösungsprozesses wurde über einen PID-Regler aktualisiert. Der Proportionalanteil war mit $P = 0,65$, der Integralteil mit $I = 0,05$ und der

Differentialteil mit $D = 0,05$ eingestellt. Das Konvergenzkriterium für den Fluidtransport ist auf den von FRANKE und BAKLANOV (2007) und TOMINAGA et al. (2008) vorgeschlagenen Wert von $1 \cdot 10^{-4}$ eingestellt. Für den Wärme- und Stofftransport ist ebenfalls ein Konvergenzkriterium von $1 \cdot 10^{-4}$ vorgegeben.

Im Kapitel 2.2.1 wurde die Herleitung für nicht isotherme Strömungen und den Stofftransport dargelegt. Es zeigte sich, dass die Diffusivität der Mechanismen untereinander in Abhängigkeit stehen kann. Das für die Simulation eingestellte Kays-Crawford-Modell löst die Abhängigkeit von Impuls-, Wärme- und Stoff-Wirbeldiffusivität über die Bestimmung einer turbulenten Prandtl-Zahl. Gemäß KAYS (1994) ergibt sich daraus eine für die meisten Fälle hinreichend genaue Übereinstimmung mit der Realität. Die Kopplung des Fluidtransportes ist als einfache (sequentielle) Kopplung mit dem Energie- und Stofftransport eingestellt. Dies reduziert den numerischen Aufwand, vernachlässigt allerdings etwaige Wechselwirkungen.

Das iterative Block-Löserverfahren unterteilt das Gesamtsystem zunächst in Gruppen abhängiger Variablen. Die aus der Systemmatrix entnommenen Blöcke bilden wiederum ein nichtlineares Gleichungssystem, welches separat gelöst wird. Anschließend werden die Gruppen des Gesamtsystems untereinander aktualisiert. Gegenüber direkten Verfahren ist diese Methode numerisch weniger robust und die Matrizen müssen sorgfältiger vorkonditioniert werden. Vorteilhaft ist der um ein Vielfaches geringere Speicherplatzbedarf, da das Verfahren weniger stark mit der Anzahl der Freiheitsgrade des Systems skaliert. Dies ermöglicht eine Einstellung spezifischer Lösereigenschaften der Subsysteme bei multiphysikalischen Problemen. Ein algebraisches Mehrgitterverfahren hilft den Konvergenzverlauf zu beschleunigen, indem die Lösung zusätzlich auf einem algebraisch erzeugtem gröberem Rechenetz ermittelt wird und somit insb. niedrigere Frequenzen im Lösungsverlauf schneller zu glätten sind. Abschließend werden die Matrizen assembliert und die Fehler bestimmt.

4.5 Verifizierung und Validierung

Um die Genauigkeit und Unsicherheit der numerischen Lösung zu beurteilen, ist es erforderlich, eine Verifizierung und Validierung durchzuführen. Dabei wird in der Verifizierung der Iterations- und Diskretisierungsfehler des erstellten Simulationsmodells quantifiziert. Die Validierung hingegen bewertet die Übereinstimmung zwischen dem Simulationsmodell und den in Kapitel 3 dokumentierten Messwerten.

4.5.1 Verifizierung

Gemäß ASME (2009) ist für die Verifizierung eine simple Einstellung der Untersuchungsparameter anzunehmen. Die geometrischen Parameter des Verifizierungsmodells entsprechen daher den geometrischen Parametern des Versuchstalls. Die Wickellüftung oben sowie unten

ist jeweils zu 90 % geöffnet, um eine typischerweise baubedingte Resthöhe zu belassen. Das Windprofil nach DIN1991-1-4/NA:2010-12 (2010) ist mit einer Basisgeschwindigkeit von $v_m(10) = 3,5 \frac{\text{m}}{\text{s}}$ berechnet und am Einlass der Strömungsdomäne des Modells aufgegeben. Dieser Wert entspricht in etwa dem Jahresmittelwert der nächstgelegenen Wetterstation zwischen 2019 und 2022. Der Wind wird aus Südrichtung aufgeprägt, was einer Abweichung von ca. $1,35^\circ$ vom Mittelwert des genannten Zeitraums entspricht. Die Luftfeuchte und Lufttemperatur sind an die Sommerzeit angelehnt und auf $\bar{\varphi} - 2\sigma \approx 50\%$ bzw. $\bar{T} + 2\sigma \approx 20^\circ\text{C}$ eingestellt.

Iterationsfehler

Der in Abbildung 4.12 dargestellte Konvergenzverlauf der äußeren Iterationen des Lösers zeigt, dass der Iterationsfehler des Modells die kritischen Werte unterschreitet. Die Beobachtung der maximalen Werte des Geschwindigkeitsbetrags U , des Drucks p , der turbulenten kinetischen Energie k und der turbulenten Dissipationsrate ε im Simulationsmodell belegt zusätzlich den stationären Zustand. Die Anforderungen an den Iterationsfehler gemäß TOMINAGA et al. (2008) und FRANKE und BAKLANOV (2007) sind somit erfüllt.

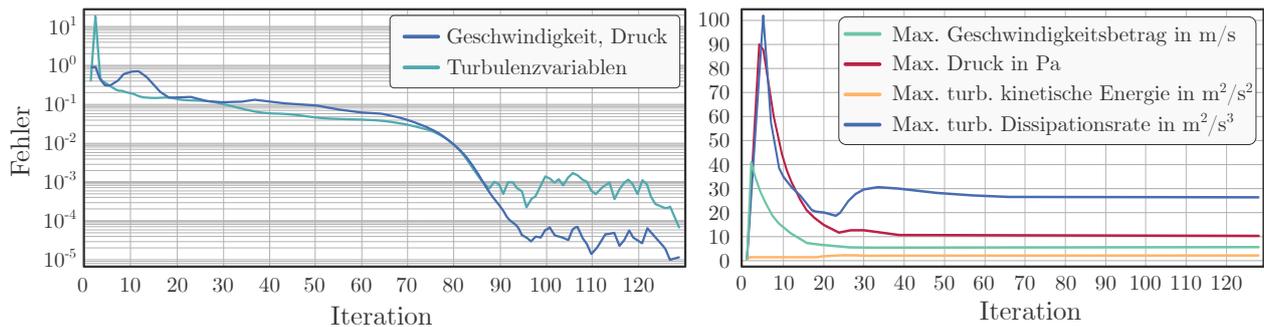


Abbildung 4.12: Konvergenzverlauf des iterativen Fehlers (links) und Beobachtung der maximalen Werte verschiedener Quantitäten im Simulationsmodell (rechts) (eigene Darstellung)

Diskretisierungsfehler

Aus den im Kapitel 4.2 vorgestellten Diskretisierungen mit der Elementanzahl von $N_{1,2,3} \in \{5.980.829; 2.694.251; 1.206.578\}$ ergeben sich nach $h_E = [(\sum_i^N \Delta V_i)/N]^{\frac{1}{3}}$ die mittleren Elementkantenlängen zu $\bar{h}_{E1,2,3} \in \{0,929; 1,212; 1,584\}$ m. Daraus lassen sich die Netzverfeinerungsraten $r_{32} = 1,307$ und $r_{21} = 1,304$ berechnen, welche im zulässigen Bereich für eine Richardson-Extrapolation liegen.

Aufgrund der maßgeblich durch die Konvektion getriebenen Prozesse wird die Richardson-Extrapolation für die Strömungsgeschwindigkeiten entlang zweier Linien in Hauptströmungsrichtung von $x = -20$ bis 20 m detailliert ausgewertet. Gemäß dem Stand der Forschung sind die Linien innerhalb des Gebäudes platziert und $\pm\Delta y = 8$ m von der Symmetrieebene des Geheges positioniert. Nach FRANKE und BAKLANOV (2007) ist die Auswertung mindestens zwei Knoten vom Rand entfernt zu beginnen, weshalb sich die Linie auf einer Höhe von

$z = 2$ m befindet. Abbildung 4.13 zeigt die Positionierung der Linien und die Auswertung der Geschwindigkeitsbeträge für das grobe, mittlere und feine Rechenetz. Zusätzlich ist der nach Richardson extrapolierte Wert dargestellt, der mit einer entsprechenden Unsicherheit nach GCI umspannt ist.

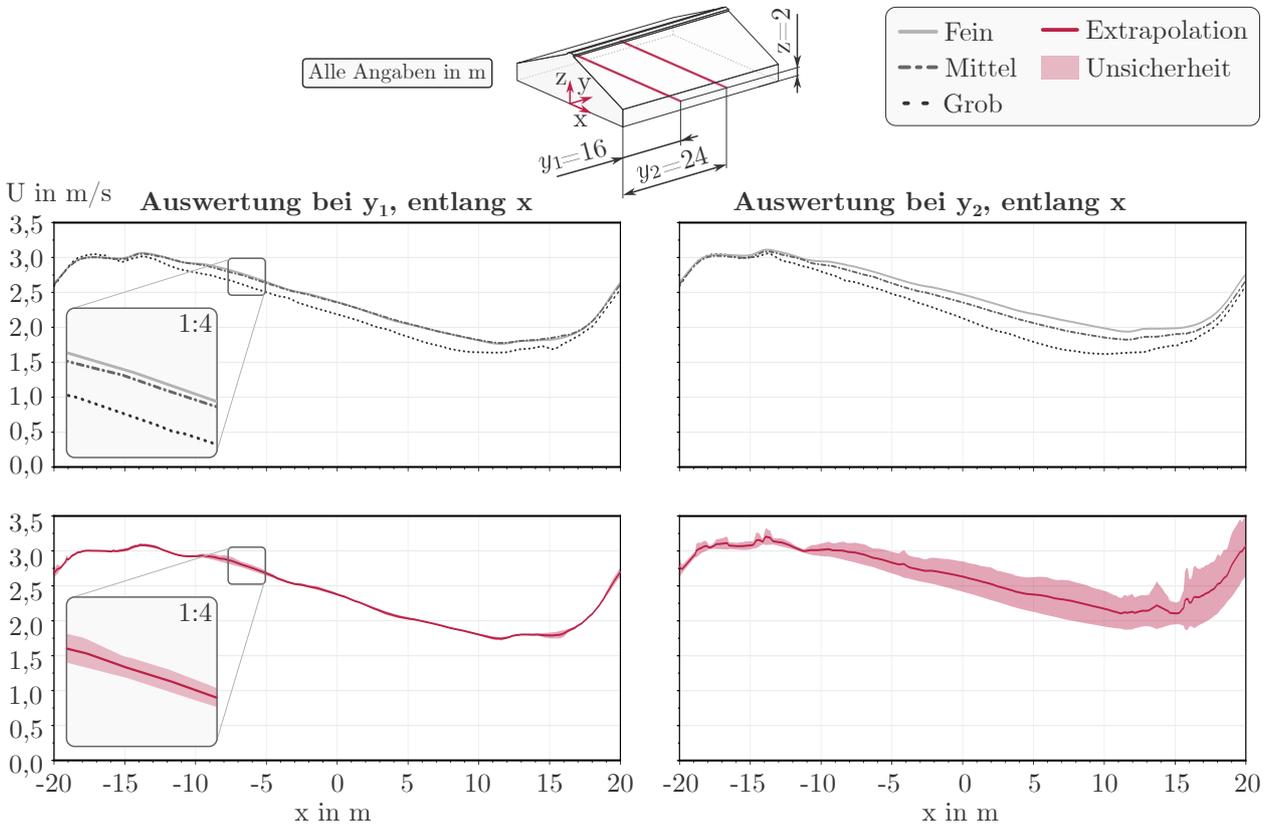


Abbildung 4.13: Ergebnisse der Auswertung der Strömungsgeschwindigkeiten in Hauptströmungsrichtung für die Parameterkonstellation des Verifizierungsmodells und Kennzeichnung der Position für die Auswertung im Stall (eigene Darstellung)

Die Auswertung der Richardson-Extrapolation entlang der horizontalen Linien mit einer Schrittweite von $\Delta x = 0,1$ m zeigt einen überwiegend monoton-konvergenten Modus und an einigen Stellen in der Nähe des Einlasses oszillierende Konvergenz bzw. Divergenz. Die gemittelte Konvergenzrate \bar{p}_{num} entlang der Linie y_1 beträgt 9,98, während sie entlang der Linie y_2 bei 2,60 liegt. Die gemittelte Konvergenzrate entlang y_1 übersteigt im Verifizierungsmodell deutlich den theoretisch maximal möglichen Wert, während \bar{p}_{num} entlang der Linie y_2 nur geringfügig über $p_{\text{num,max}} = 2,0$ liegt.

Daraus resultiert entlang y_1 ein mittlerer Fehler zwischen dem mittleren und feinen Netz von $\bar{\epsilon}_{21} = 0,45\%$ und vom feinen zum extrapolierten Ergebnis beträgt dieser Fehler $\bar{\epsilon}_{\text{extr}} = 0,14\%$. Die Unsicherheit nach GCI ist $\pm 0,17\%$. Entlang y_2 ist der Fehler zwischen den

Ergebnissen des mittleren und feinen Netzes $\bar{\epsilon}_{21} = 3,69 \%$. Zwischen dem feinen und abgeschätzten extrapoliertem Ergebnis beträgt der durchschnittliche Fehler $\bar{\epsilon}_{\text{extr}} = 4,84 \%$. Die zugehörige Unsicherheit nach \overline{GCI} ist $\pm 6,54 \%$.

Analog zur Strömungskenngröße können die Werte der mittleren Methankonzentration, Temperatur und Luftfeuchte entlang der Linien y_1 und y_2 ausgewertet werden. Diese Auswertungen sind in Tabelle 4.2 zusammengefasst und zeigen vergleichbare Ergebnisse hinsichtlich der Konvergenzrate und Unsicherheiten wie die Auswertung der Strömungsgeschwindigkeit.

Tabelle 4.2: Netzunabhängigkeitsstudie und Unsicherheitsanalyse der mittleren Temperatur, Methankonzentration und Luftfeuchte. Auswertung entlang der x -Achse von $x_{\min} = -20$ m bis $x_{\max} = +20$ m bei $y_1 = 16$ m und $y_2 = 24$ m

	\bar{T} in °C		\bar{c}_{CH_4} in $\frac{\text{mol}}{\text{m}^3}$		$\bar{\varphi}$	
	y_1	y_2	y_1	y_2	y_1	y_2
$\bar{\Phi}_3$	20,157	20,168	$1,62 \cdot 10^{-4}$	$1,71 \cdot 10^{-4}$	0,507	0,509
$\bar{\Phi}_2$	20,095	20,108	$1,31 \cdot 10^{-4}$	$1,39 \cdot 10^{-4}$	0,504	0,505
$\bar{\Phi}_1$	20,084	20,091	$1,24 \cdot 10^{-4}$	$1,28 \cdot 10^{-4}$	0,504	0,505
\bar{p}_{num}	6,692	5,594	7,433	5,748	8,780	6,977
$\bar{\Phi}_{\text{extr}}$	20,067	20,028	$1,21 \cdot 10^{-4}$	$1,22 \cdot 10^{-4}$	0,504	0,505
$\bar{\epsilon}_{21}$	0,059 %	0,082 %	5,616 %	8,134 %	0,070 %	0,136 %
$\bar{\epsilon}_{\text{extr}}$	0,016 %	0,034 %	2,129 %	4,641 %	0,027 %	0,047 %
\overline{GCI}	0,020 %	0,042 %	2,565 %	5,452 %	0,034 %	0,059 %
\bar{U}_{num}	$\pm 4,10 \cdot 10^{-3}$	$\pm 8,50 \cdot 10^{-3}$	$\pm 3,17 \cdot 10^{-6}$	$\pm 6,99 \cdot 10^{-6}$	$\pm 1,74 \cdot 10^{-4}$	$\pm 2,97 \cdot 10^{-4}$
\bar{u}_{num}	$\pm 3,57 \cdot 10^{-3}$	$\pm 7,39 \cdot 10^{-3}$	$\pm 2,76 \cdot 10^{-6}$	$\pm 6,08 \cdot 10^{-6}$	$\pm 1,51 \cdot 10^{-4}$	$\pm 2,58 \cdot 10^{-4}$

Aufgrund des zu erwartenden numerischen Aufwands der statistischen Versuchsplanung wird das mittlere Rechnetz für die weiteren Untersuchungen genutzt. Die in der Verifizierung dargelegten Fehler und Unsicherheiten werden hierfür als hinreichend gering bewertet. Zur Belegung der Validität der mittleren Diskretisierung schließt eine Bewertung an.

4.5.2 Validierung

Im Rahmen der Validierung werden die experimentellen Ergebnisse aus Kapitel 3 mit den numerischen Ergebnissen der Simulation abgeglichen. Da eine RANS-Simulation den stationären Zustand in der Strömungsdomäne berechnet und zeitliche Veränderungen nicht erfasst werden, ist es empfehlenswert, eine Messkampagne auszuwählen, bei der die Daten eine geringe zeitliche Variation aufweisen. Die Boxplots zur Auswertung der Geschwindigkeitsmessungen belegen dahingehend Unterschiede zwischen den Messkampagnen: Die Antennen der Messkampagne ① weisen für die Luv-Positionen die größte Spanne der Antennen und einen

dezentralen Median auf, wohingegen die Messkampagnen ② und ③ auf einem ähnlichen Niveau liegen (s. Kapitel 3.4.2). Daher wurde die Validierung der RANS-Simulation mit den vorherrschenden Bedingungen der Messkampagne ② durchgeführt und mit den Messwerten der Kampagne abgeglichen.

Da die Einstellung der Modellparameter und Randbedingungen mit Unsicherheiten behaftet sein kann, ordnet eine Sensitivitätsanalyse diese ein, um die optimale Konstellation der Simulation für die Validierung zu ermitteln. Bei den Randbedingungen gilt es zu beachten, dass die Auswertung der Wetterstationen am Versuchsstall und des DWD eine Differenz aufweisen, weshalb zum einen die Basisgeschwindigkeit für das Windprofil $v_m(10) \in \{2,12; 3,41; 4,69\} \frac{\text{m}}{\text{s}}$ und zum anderen die Windrichtung zwischen $\alpha \in \{186,6; 197,8; 209,0\}^\circ$ variiert werden kann. Diese Parameter stellen je die minimale, mittlere und maximale Einstellung der Messwerte dar. Beim Turbulenzmodell verwenden die in Kapitel 2.5.1 analysierten Arbeiten am häufigsten das k - ε - und *realizable*- k - ε -Modell, wobei einige Arbeiten ergänzend auf das k - ω -SST-Modell verweisen. Weiterhin können geometrische Objekte im Umfeld des untersuchten Gebäudes nach FRANKE und BAKLANOV (2007) im Simulationsmodell unterschiedlich berücksichtigt werden. Während Objekte in einem Radius von h_G detailliert zu modellieren sind, können weiter entfernte Objekte über vereinfachte Geometrien oder eine äquivalente Sandrauheit abgebildet werden. Die Sensitivitätsanalyse wird daher um die Einordnung der Genauigkeit der in Abbildung 4.14 dargestellten Varianten ergänzt.

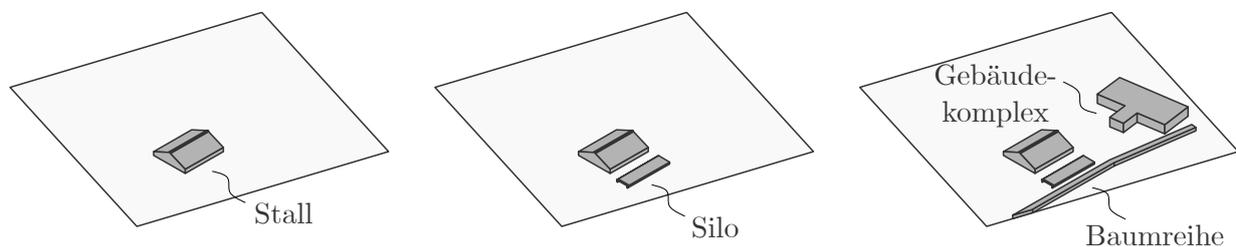


Abbildung 4.14: Geometrische Varianten des CAD-Modells für die Validierung mittels RANS-Simulation (eigene Darstellung)

Die resultierenden Geschwindigkeitskomponenten der verschiedenen Simulationsergebnisse, mit den o. g. variierten Parametern, können jeweils den Messergebnissen gegenübergestellt und die daraus resultierende Genauigkeit anhand der Validierungsmetriken ermittelt werden. Auf diese Weise ist der Parametereinfluss hinsichtlich der Abweichung vom Messwert zu identifizieren und bezüglich einer allgemeinen Übereinstimmung (VM), einem Bias (MG und FB), stochastischer Fehler (NMSE und VG) und großer Abweichungen (FAC2) einzuordnen (s. auch Kapitel 2.2.4). Da die aufwärts gerichtete Geschwindigkeitskomponente w geringe Absolutwerte mit Schwankungen im Bereich der Messgenauigkeit des Sensors aufweist, ist die Auswertung der Metriken im Folgenden auf die Auswertung der maßgeblich bestimmenden Strömungsgeschwindigkeiten u und v beschränkt.

Für die Sensitivitätsanalyse der Geometrien ergeben sich die in Abbildung 4.15 dargestellten Simulations- und Messwerte der Geschwindigkeiten u und v an den Sensorpositionen der Messkampagne ② sowie die zugehörigen Metriken. Die Analyse zeigt, dass eine Vereinfachung ohne umliegende Geometrien zu höheren negativen Geschwindigkeiten in Hauptströmungsrichtung u im Vergleich zu den Messwerten führt. Das Modell mit Berücksichtigung des Silos und das Modell mit ergänzter Baumreihe und Nachbargebäuden liegen dagegen auf einem vergleichbaren Niveau der allgemeinen Übereinstimmung (VM). Die Grenzwerte der systematischen Fehler werden durch die Simulationsmodelle mit externen Geometrien entweder durch den FB oder MG erfüllt, wohingegen das Modell ohne externe Geometrien die Grenzwerte beider Metriken überschreitet. Diese Beobachtung setzt sich für die stochastischen Fehler fort: Während die Metriken der geometrisch komplexeren Simulationsmodelle auf einem identischen Niveau liegen und die Grenzwerte erfüllen, überschreitet das vereinfachte Modell diese für den VG der Geschwindigkeitskomponente v . Der FAC2 weist für das Modell ohne externe Geometrien in Hauptströmungsrichtung u große Abweichungen aus.

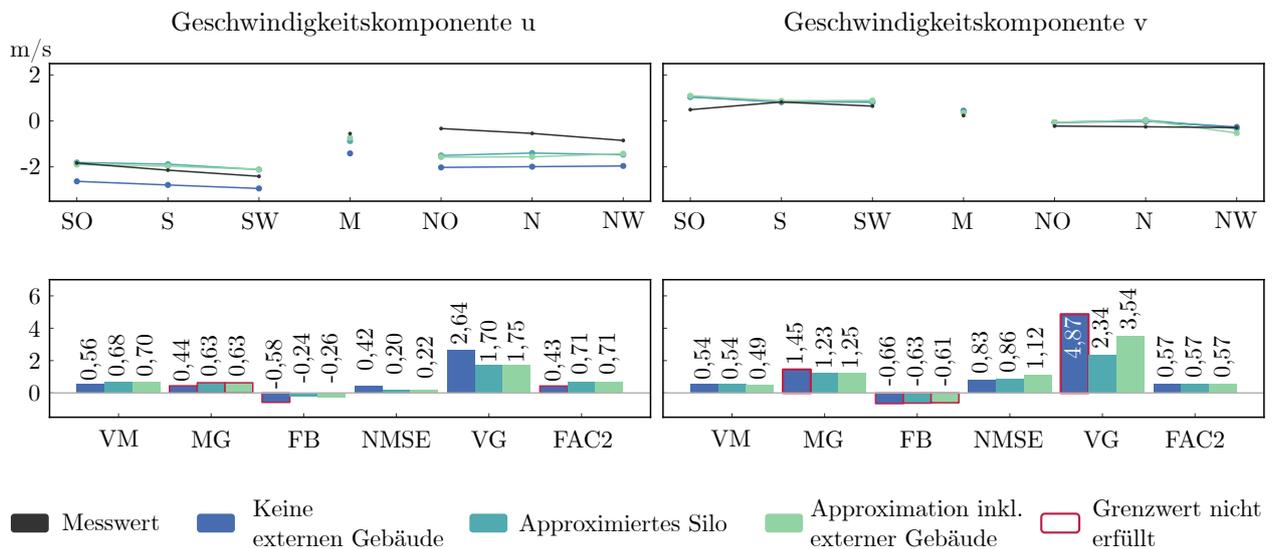


Abbildung 4.15: Sensitivitätsanalyse der geometrischen Varianten des CAD-Modells (eigene Darstellung)

In Abbildung 4.16 ist abzulesen, dass ein eingestelltes Windgeschwindigkeitsprofil nach DWD und der lokalen Station am Versuchsstall die Ergebnisse der Messung am Luv umspannen und somit die mittlere Einstellung die beste allgemeine Übereinstimmung mit den Messwerten aufweist (VM).

Am Lee hingegen zeigt der Abgleich der Hauptgeschwindigkeitskomponente u , dass die geringsten Fehler für die niedrigste Windgeschwindigkeit zu erwarten sind. Analog zur Beobachtung der Sensitivitätsanalyse der Geometrien weisen die Ergebnisse aller Modelle einen Bias auf, der teils die Grenzwerte überschreitet (MG und FB). Das Simulationsmodell mit der höchsten Basisgeschwindigkeit für das Windprofil zeigt außerdem nach FAC2 große Abweichungen.

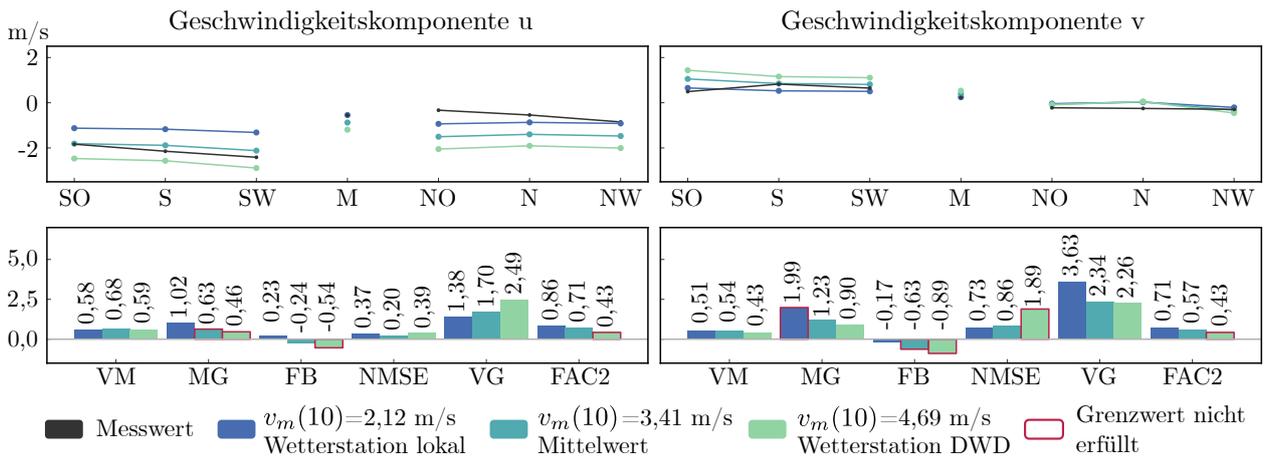


Abbildung 4.16: Sensitivitätsanalyse der Windgeschwindigkeit (eigene Darstellung)

Abbildung 4.17 visualisiert die Sensitivitätsanalyse bei variiertem Windrichtung. An der vergleichbaren allgemeinen Übereinstimmung nach VM für die Geschwindigkeitskomponente u , aber differenter Metrik für die Komponente v ist abzulesen, dass die unterschiedliche Einstellung der Windrichtung maßgeblich Einfluss auf die Geschwindigkeitskomponente v nimmt. Eine Einstellung der Windrichtung gemäß der Wetterstation am Versuchsstall ($\alpha = 186,6^\circ$) erfüllt die Metriken für den Bias und stochastische Fehler nicht und weist aufgrund der Absolutwerte um $0,0 \frac{m}{s}$ für v einen unverhältnismäßig hohen Wert für VG aus. Die gemittelte Windrichtung zwischen der Wetterstation des DWD und dem Versuchsstall weist die beste Validität des Modells nach.

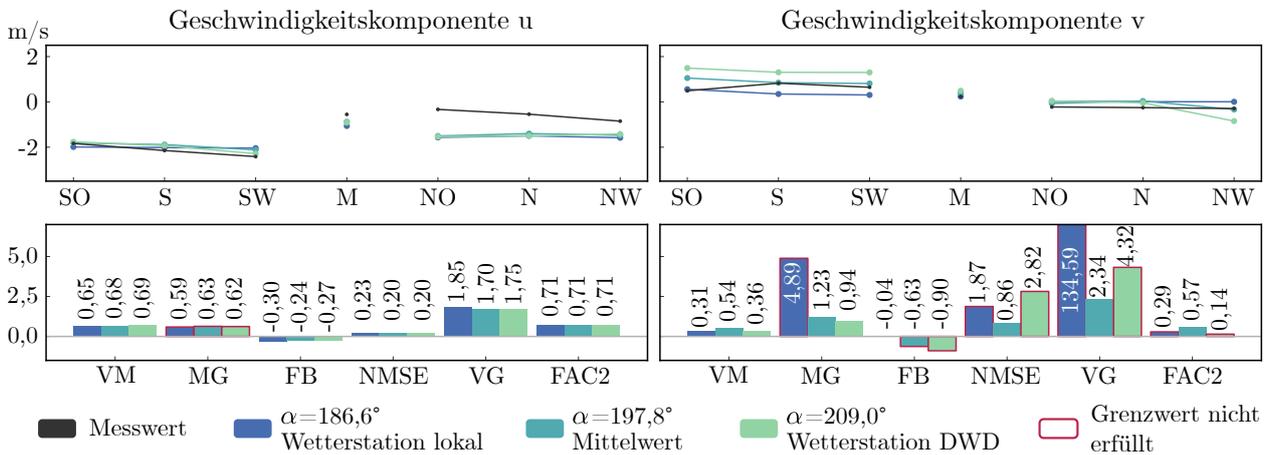


Abbildung 4.17: Sensitivitätsanalyse der Windrichtung α (eigene Darstellung)

Abschließend gilt es, den Einfluss des Turbulenzmodells zu analysieren. Die Sensitivitätsanalyse ist in Abbildung 4.18 für das *realizable-k- ϵ* -Modell, das *k- ϵ* -Modell und das *k- ω* -SST-Modell im Abgleich mit den Messungen dargestellt. Die Turbulenzmodelle weisen gegenüber den zuvor untersuchten Größen geringere Abweichungen voneinander auf. Über alle Geschwindigkeitskomponenten hinweg bieten das *k- ϵ* -Modell und das *realizable-k- ϵ* -Modell

vergleichbare Übereinstimmungen mit der Messung. Das k - ω -SST-Modell erfüllt keine der Metriken hinsichtlich systematischer Fehler und überschreitet zudem den Grenzwert der FAC2 Metrik bezüglich großer Abweichungen des Modells.

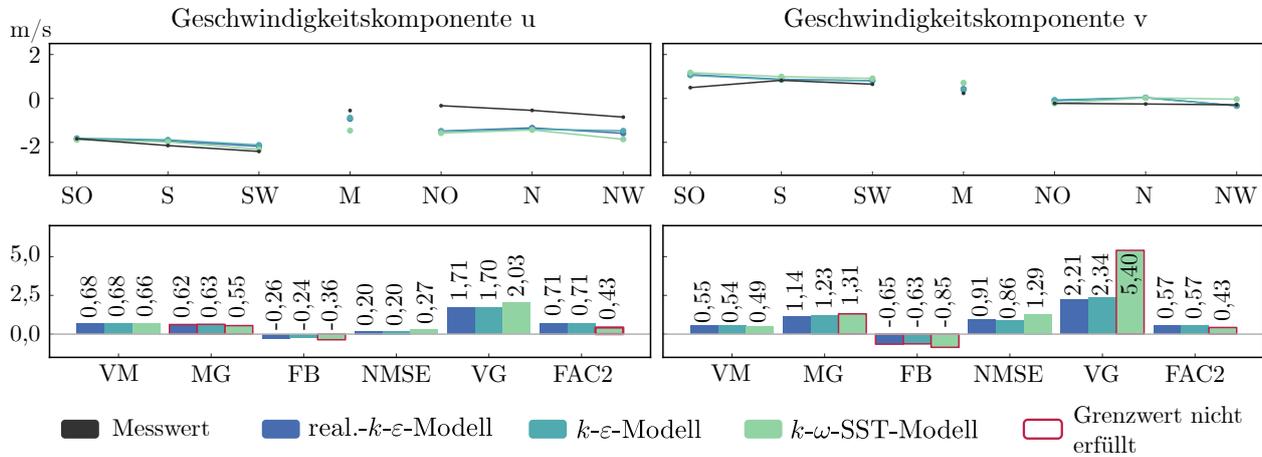


Abbildung 4.18: Sensitivitätsanalyse des Turbulenzmodells (eigene Darstellung)

Basierend auf den Ergebnissen der Sensitivitätsanalyse und den Messungen aus Kapitel 3, ist die folgende Kombination der Parameter der Simulation zur Validierung eingestellt und um die in Kapitel 4.3 dokumentierten Randbedingungen ergänzt:

Geometrie: Berücksichtigung des detailliert modellierten Silos

Windgeschwindigkeit: $v_m(10) = 3,41 \frac{\text{m}}{\text{s}}$

Windrichtung: $\alpha = 197,8^\circ$

Turbulenzmodell: k - ϵ -Modell

Lufttemperatur am Einströmrand: $T_\partial = 17,55 \text{ }^\circ\text{C}$

Luftfeuchte am Einströmrand: $\varphi_\partial = 78,63 \%$

Methangehalt am Einströmrand: $c_{\text{CH}_4} = 2,00 \text{ ppm}$

Tabelle 4.3 listet die Auswertung der Validierungsmetriken für die Simulationsergebnisse unter Verwendung der oben genannten Parameter im Vergleich zu den Messergebnissen aus Kapitel 3. Die Metriken werden gemäß der Einhaltung der im Kapitel 2.2.4 dokumentierten Grenzwerte nach PATRYL und GALERIU (2011) farblich kodiert. Eingehaltene Grenzwerte werden grün dargestellt, während überschrittene Metriken rot markiert sind. Gemäß PATRYL und GALERIU (2011) gilt das Simulationsmodell als valide: Die Grenzwerte beider Metriken zu stochastischen Fehlern (NMSE und VG) werden für die Geschwindigkeiten, Lufttemperatur, Methankonzentration und Luftfeuchte eingehalten; der Grenzwert des FAC2 wird ebenfalls über alle Größen hinweg eingehalten; beim Bias wird entweder der MG oder der FB erfüllt. Das Modell gilt trotz der Tendenz zum Bias als valide, da jeweils eine der beiden

Bias-Metriken erfüllt ist. Diese Tendenz lässt sich mit den Vereinfachungen der Geometrien im Inneren des Stalls begründen, woraus eine vom Simulationsmodell zu hoch vorhergesagte Querbelüftung resultiert. Dies ist auch in den Abbildungen 4.15 bis 4.18 an der Geschwindigkeitskomponente u der windabgewandten Seite abzulesen. Eine umfassende Diskussion der Abweichung zwischen den Mess- und Simulationswerten findet in Kapitel 7.2.2 statt.

Tabelle 4.3: Validierungsmetriken bei mittlerer Einstellung der Parameter

Quantität	VM	MG	FB	NMSE	VG	FAC2
u	0,68	0,63	-0,24	0,20	1,70	0,71
v	0,54	1,23	-0,63	0,86	2,34	0,57
T	0,98	1,00	0,00	0,00	1,00	1,00
c_{CH_4}	0,70	1,30	0,12	0,02	1,11	1,00
φ	0,93	1,07	0,07	0,01	1,01	1,00

Abbildung 4.19 stellt die Konturdiagramme der Validierungssimulation auf einer x - y -Ebene und der Schnitthöhe von $z = 1,7$ m dar. Die Darstellung zeigt einen Ausschnitt der Ergebnisse im und um den Versuchsstall.

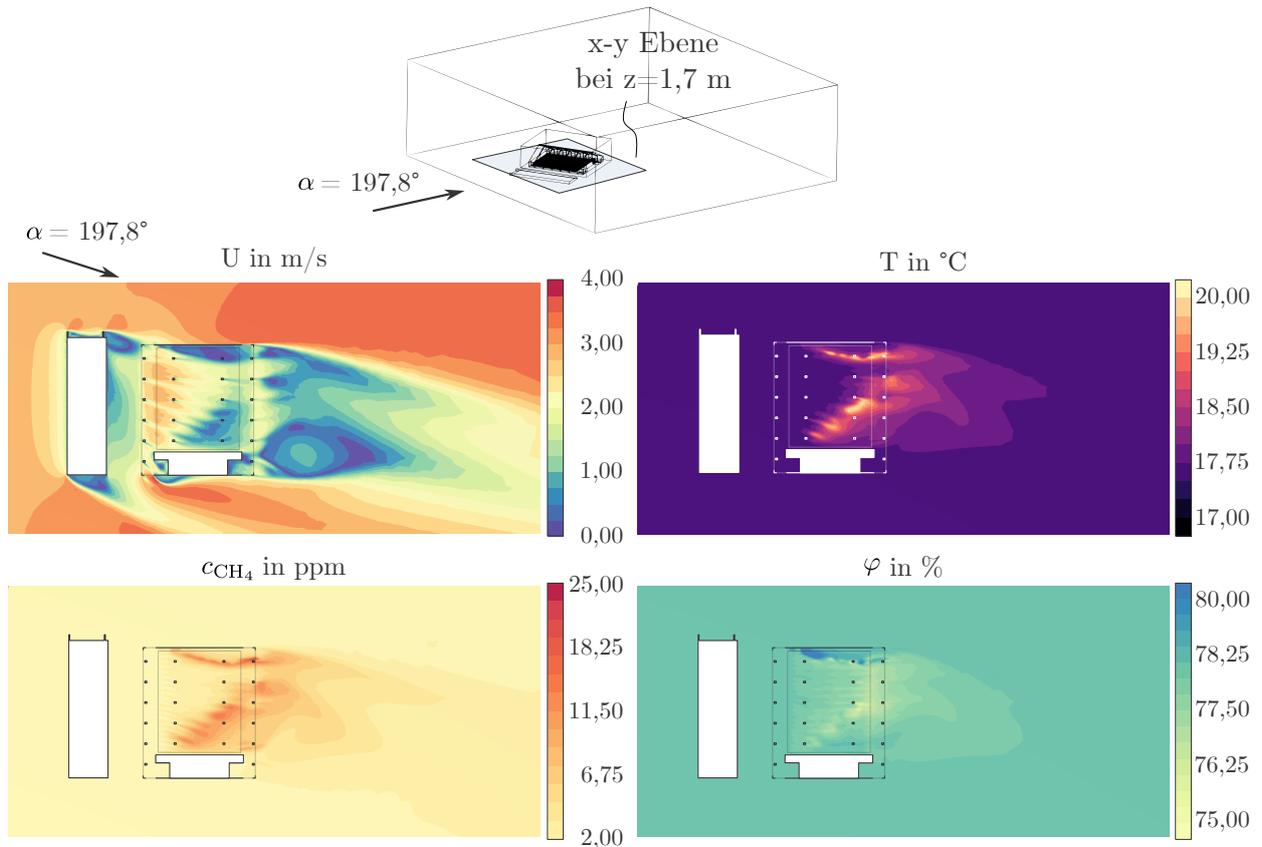


Abbildung 4.19: Ausschnitt der Ergebnisse des Geschwindigkeitsbetrages, der Lufttemperatur, Luftfeuchte und des Methangehaltes auf der Schnittebene x - y bei $z = 1,7$ m (eigene Darstellung)

In der Auswertung des Geschwindigkeitsbetrages ist der Einfluss des Silos, durch die abgesenkte Strömungsgeschwindigkeit am Luv der Wickellüftung, deutlich sichtbar. Ergänzend sind Eckenwirbel durch die Abrisskanten der Tore und um die Funktionsräume zu erkennen. Im Nachlauf des Stalls sind die gegenläufigen Wirbel der stationär berechneten Wirbelstraße sichtbar. Das Temperaturfeld ist maßgeblich durch die Konvektion geprägt und weist insbesondere in den Bereichen geringer Geschwindigkeiten hohe Temperaturen auf. Es ist ebenfalls zu erkennen, dass die Lufttemperatur der über die Wickellüftung abgeführten Luftmassen im direkten Nachlauf eine um $\Delta T \approx 0,25 \text{ °C}$ bis $2,00 \text{ °C}$ erhöhte Lufttemperatur aufweist. Diese erhöhte Lufttemperatur führt im Gegenzug zu lokal geringeren Werten der Luftfeuchte. Der Methangehalt zeigt sich im Nachlauf ebenfalls erhöht, mit etwa $c_{\text{CH}_4} \approx 11,5 \text{ ppm}$ bis $18,25 \text{ ppm}$.

Die Konturdiagramme der z - x -Schnittebene in Abbildung 4.20 bestätigen die auf der x - y -Schnittebene sichtbaren Effekte. Es sind die durch Konvektion getriebenen Transportprozesse der Lufttemperatur, Luftfeuchte und der Methanemissionen sichtbar. Die Querbelüftung nimmt etwa ab der Stallmitte sichtbar ab, was mit einer erhöhten Lufttemperatur im Gehege einhergeht. Erneut zeigt sich, dass durch die lokal erhöhten Lufttemperaturen die Luftfeuchte absinkt.

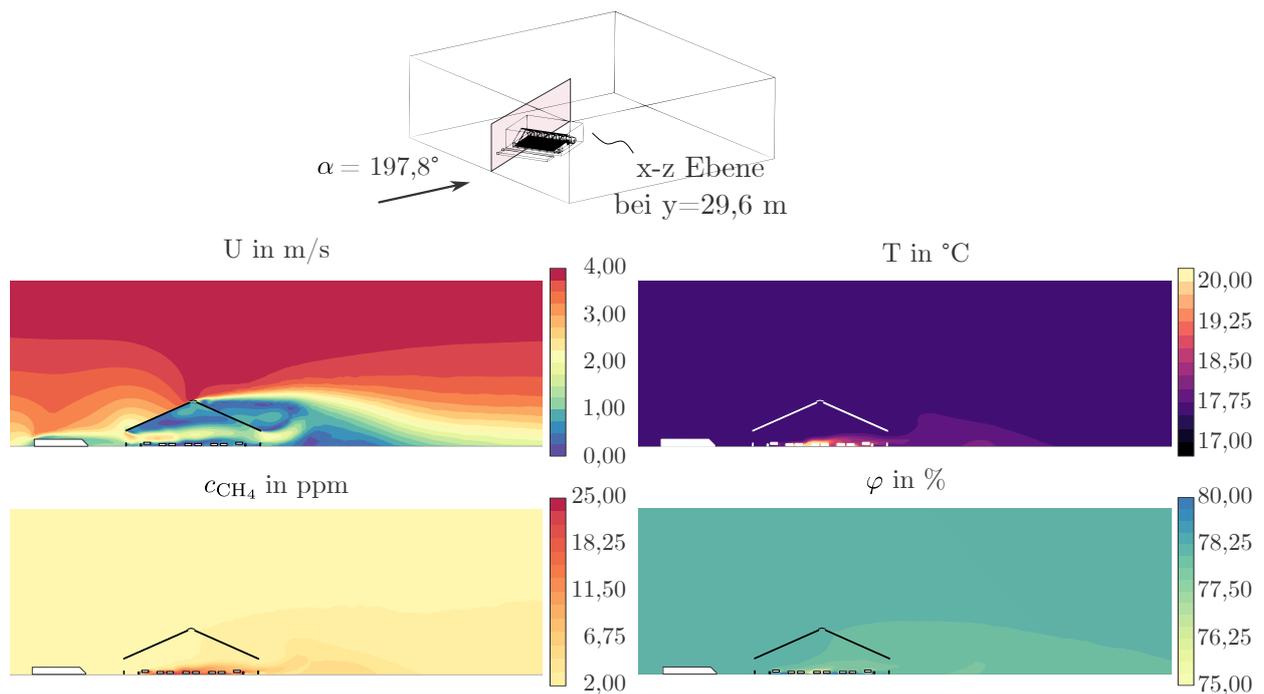


Abbildung 4.20: Ausschnitt der Ergebnisse des Geschwindigkeitsbetrages, der Lufttemperatur, Luftfeuchte und des Methangehaltes auf der Schnittebene z - x bei $y = 29,6 \text{ m}$ (eigene Darstellung)

Das in Abbildung 4.21 dargestellte Konturdiagramm zum y^+ -Wert zeigt auf, dass die nach FRANKE und BAKLANOV (2007) geforderten Werte im Stall und dem direkten Umfeld erreicht werden. In der etwa h_G entfernten Umgebung zeigt der y^+ -Wert Überschreitungen des

Grenzwertes. Dies lässt sich insbesondere mit der am Einlass und unterhalb von z_{min} konstant aufgeprägten Windgeschwindigkeit nach DIN1991-1-4/NA:2010-12 (2010) erklären. Ergänzend ist die Einstellung der Randschichten hinsichtlich des Stalls optimiert, woraus eine teils gröbere Randschicht mit erhöhten y^+ -Werten im weiteren Umfeld der Strömungsdomäne resultiert.

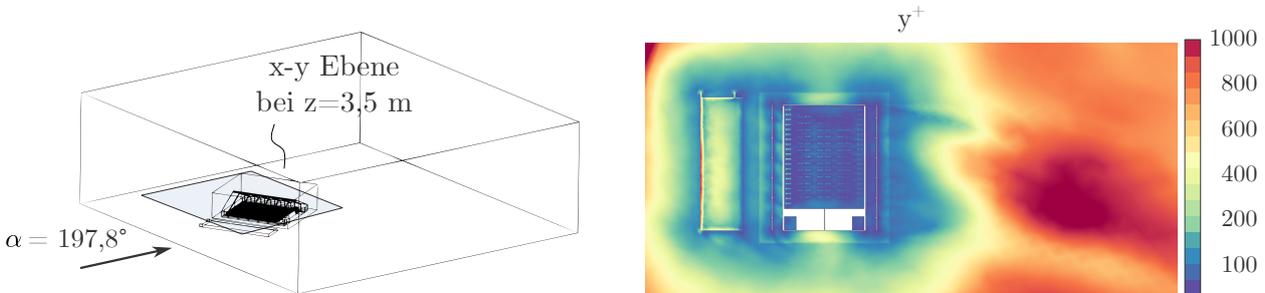


Abbildung 4.21: Darstellung des y^+ Wertes für die Validierungssimulation in der x - y -Ebene bei einer Schnittdarstellung auf einer Höhe von $z = 3,5$ m (eigene Darstellung)

4.6 Statistische Versuchsplanung

Die Verifizierung und Validierung hat aufgezeigt, dass das Simulationsmodell die Anforderungen objektiver Kriterien erfüllt und die relevanten Strömungscharakteristiken im Stall erfasst. Die ermittelten Einstellungen zur Diskretisierung, zum Löser und zu den Randbedingungen werden daher folgend für die numerischen Simulationsmodelle einer statistischen Versuchsplanung übernommen. Statistische Versuchsplanungen verfolgen das Ziel, den experimentellen Aufwand zur Untersuchung der Effektstärke zu reduzieren. Hierzu werden die Faktoren in verschiedenen Versuchsdurchläufen systematisch variiert. Die abschließende statistische Auswertung liefert eine Aussage zur statistischen Signifikanz der jeweiligen Faktoren. Diese Signifikanz wird über die normierte Effektstärke auf eine oder mehrere Zielgrößen bestimmt, indem diese gegen die Nullhypothese geprüft werden. Die Nullhypothese ist in diesem Fall so festgelegt, dass die normierte Effektstärke um den Wert null (kein Effekt) normal verteilt ist.

Zunächst werden daher in Kapitel 4.6.1 drei Zielgrößen definiert, die das Tierwohl, die Emissionen und Durchlüftung des Stalls bewerten. Die Menge der in Kapitel 4.1.3 definierten geometrischen Faktoren wird um die Randbedingungen der Windrichtung, Windgeschwindigkeit, Lufttemperatur und Luftfeuchte ergänzt, sodass insgesamt 14 Faktoren zu variieren sind. Die statistische Versuchsplanung wird konsekutiv aus zwei unterschiedlichen Methoden aufgebaut. Das in Kapitel 4.6.2 dokumentierte *Screening* ermittelt, welche der 14 Faktoren hinsichtlich der Zielgrößen statistisch signifikant sind. Da ein *Screening* mit einer geringen Auflösung einhergeht (s. Kapitel 2.3), folgt in Kapitel 4.6.3 eine detailliertere Auswertung ausgewählter Faktoren anhand eines Antwortflächen-Versuchsplans.

Im Anhang D.1 und D.2 sind die Konturdiagramme der einzelnen Simulationen aus dem *Screening*- und Antwortflächen-Versuchsplan hinterlegt.

4.6.1 Zielgrößen der statistischen Versuchsplanung

Für die Auswertung der einzelnen Versuchsdurchläufe der statistischen Versuchsplanung ergeben sich drei mögliche Fragestellungen. Dieses Kapitel wird eine Möglichkeit vorstellen, anhand derer die Quantifizierung der Fragestellungen über integrale oder räumlich gemittelte Zielgrößen möglich wird:

- (i) Wie ist die Effektstärke auf das Tierwohl zu bewerten?
- (ii) Kann die Emissionsbilanz durch Faktoreinstellungen beeinflusst werden?
- (iii) Können signifikante Unterschiede bei der Durchlüftung des Stalls identifiziert werden?

(i) Das Tierwohl kann über den Temperatur-Luftfeuchte-Index (engl. *Temperature Humidity Index* (THI)) objektiv bewertet werden. Der THI berechnet sich gemäß THOM (1959):

$$\text{THI} = 0,8 \cdot T + \left(\frac{\varphi}{100}\right) \cdot (T - 14,4) + 46,4 . \quad (4.4)$$

Ab einem $\text{THI} > 67$ gilt milder Hitzestress, ab $\text{THI} > 72$ mäßiger Hitzestress und mit $\text{THI} > 79$ setzt ein starker Hitzestress ein. Da Kühe homoiotherme Lebewesen sind, bedeutet dies eine zunehmende Lebensgefahr, mit möglichen Todesfällen ab einem THI von über 90. In den folgenden Auswertungen wird der THI über die gemittelten Werte der Lufttemperatur und Luftfeuchte in den Gehegen berechnet. Die Abmessungen der Grundfläche des Geheges wurden in Kapitel 4.1 bereits für den Versuchsstall dargelegt. Die Grundfläche der Versuchsdurchläufe ist je nach Einstellung der Faktoren über das parametrisierte Modell bestimmt, wohingegen die Höhe für alle Modelle auf 2,0 m beschränkt ist. (THOM, 1959)

(ii) Die Emissionen werden als Netto-Bilanz zwischen der in den Stall einströmenden und aus dem Stall ausströmenden Masse an Methan bestimmt. Auf diese Weise wird garantiert, dass lediglich die Anreicherung an Methan durch die enterische Fermentation und das Lagerungssystem ausgewertet wird. Aufgrund der unterschiedlichen Anzahl der Kühe in den Modellvarianten der Versuchsdurchläufe wird diese Bilanz je Kuh normiert.

(iii) Eine Möglichkeit, die Durchlüftung zu charakterisieren, ist die Bestimmung der Luftwechselrate. Die Luftwechselrate beschreibt, wie oft die Luft in einem Gebäude innerhalb einer Stunde ausgetauscht wird. Frei belüftete Kuhställe erreichen für unterschiedliche Randbedingungen üblicherweise Werte im Bereich von $10,0 \text{ h}^{-1}$ bis $126,0 \text{ h}^{-1}$ (SAHA, FIEDLER, AMMON et al., 2014).

4.6.2 Screening-Versuchsplan und Auswertung

Auf Kosten eindeutig zuzuweisender Wechselwirkungen und einer Linearisierung der Effekte lässt sich der Untersuchungsaufwand eines vollfaktoriellen Versuchsplans mit 14 Variablen von 2^{14} mittels Plackett-Burman-Versuchsplan auf minimal 15 Versuchsdurchläufe reduzieren. Zusätzliche freie Spalten im reduzierten Versuchsplan ermöglichen es, die kumulativen Wechselwirkungseffekte abzuschätzen (SIEBERTZ, BEBBER und HOCHKIRCHEN, 2017). Um den Einfluss der Effekte zu visualisieren, werden aus den Ergebnissen der Untersuchung Wahrscheinlichkeitsnetze abgeleitet.

Der Plackett-Burman-Versuchsplan generiert die zu erstellenden Varianten des numerischen Simulationsmodells, wobei + eine hohe und – eine niedrige Einstellung des Faktors kennzeichnet. Aus der Anzahl von 20 Durchläufen für 14 Faktoren ergibt sich die Auflösungsstufe III. Tabelle 4.4 listet die Faktoreinstellungen der Versuchsdurchläufe und führt die Auswertung der in Kapitel 4.6.1 festgelegten Zielgrößen auf.

Tabelle 4.4: Plackett-Burman-Versuchsplan für 14 Faktoren und die ausgewerteten Ergebnisse des THI, Emissionsstroms und der Luftwechselrate, erstellt mittels MINITAB (Version 19)

N	T	φ	α	$u_m(10)$	l_{NG}	V_A	V_D	V_S	h_G	β	l_G	b_G	W_o	W_u	THI	\dot{m}_{CH_4} in $\frac{kg}{h \cdot Kuh}$	R_V in $\frac{1}{h}$
1	+	-	+	+	-	-	-	-	+	-	+	-	+	+	71	0,01489	360,78
2	+	+	-	+	+	-	-	-	-	+	-	+	-	+	79	0,00887	13,02
3	-	+	+	-	+	+	-	-	-	-	+	-	+	-	41	0,01480	2,67
4	-	-	+	+	-	+	+	-	-	-	-	+	-	+	37	0,01520	31,91
5	+	-	-	+	+	-	+	+	-	-	-	-	+	-	77	0,02274	98,03
6	+	+	-	-	+	+	-	+	+	-	-	-	-	+	83	0,01978	7,56
7	+	+	+	-	-	+	+	-	+	+	-	-	-	-	83	0,01159	1,97
8	+	+	+	+	-	-	+	+	-	+	+	-	-	-	97	0,02570	2,13
9 ^a	-	+	+	+	+	-	-	+	+	-	+	+	-	-	72	0,15067	1,12
10	+	-	+	+	+	+	-	-	+	+	-	+	+	-	75	0,01442	4,11
11	-	+	-	+	+	+	+	-	-	+	+	-	+	+	34	0,01388	18,57
12	+	-	+	-	+	+	+	+	-	-	+	+	-	+	87	0,01859	0,95
13	-	+	-	+	-	+	+	+	+	-	-	+	+	-	35	0,01651	31,96
14	-	-	+	-	+	-	+	+	+	+	-	-	+	+	37	0,01483	6,57
15	-	-	-	+	-	+	-	+	+	+	+	-	-	+	39	0,01430	20,73
16 ^a	-	-	-	-	+	-	+	-	+	+	+	+	-	-	56	0,01577	0,31
17	+	-	-	-	-	+	-	+	-	-	+	+	+	-	83	0,02000	1,94
18	+	+	-	-	-	-	+	-	+	-	+	+	+	+	76	0,01335	0,57
19	-	+	+	-	-	-	-	+	-	+	-	+	+	+	44	0,01292	6,94
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	59	0,00407	3,37

^a Stationäre Zielgrößen, bei $\epsilon > 1 \cdot 10^{-4}$.

Die hohe und niedrige Einstellung der ergänzten Temperatur T , Luftfeuchte φ und Windgeschwindigkeit $v_m(10)$ erfolgte durch eine statistische Auswertung der Wetterstation des DWD (s. Kapitel 3). Da eine zu große Schrittweite der Faktoreinstellungen bei einer Linearisierung der Effekte nachteilig sein kann (SIEBERTZ, BEBBER und HOCHKIRCHEN, 2017), wurde der Mittelwert der Messungen von Januar 2019 bis Dezember 2022 berechnet und zwei Standardabweichungen addiert oder subtrahiert. Damit schließen die Einstellungen der hohen und niedrigen Faktoren ca. 95,4 % der Werte ein. Für die Windgeschwindigkeit folgt daraus $v_{m+}(10) = 6,5 \frac{m}{s}$ und $v_{m-}(10) = 0,5 \frac{m}{s}$. Die Temperatur beträgt hierbei $T_+ = 23,5 \text{ }^\circ\text{C}$ und $T_- = -0,5 \text{ }^\circ\text{C}$. Die Luftfeuchte auf hoher Einstellung entspricht $\varphi_+ = 100 \%$ und auf niedriger Einstellung $\varphi_- = 57 \%$. Die Windrichtung α wurde hingegen auf einen gerundeten Wert von 0° für eine vollständige Querbelüftung mit Nordwind und eine laterale Anströmung bei 270° mit Westwind festgelegt.

Abbildung 4.22 zeigt die statistische Auswertung der Effekte aller 14 Faktoren auf die Zielgröße THI in einem Wahrscheinlichkeitsnetz. Gemäß der Nullhypothese wird erwartet, dass die Auswertungen der Effekte der Faktoren aller Versuchsdurchläufe um null normalverteilt sind und somit der roten Linie im Wahrscheinlichkeitsnetz folgen. Zusätzlich wird ein Vertrauensniveau berücksichtigt, das die Irrtumswahrscheinlichkeit bezüglich der Nullhypothese abschätzt. Hieraus ergibt sich der p-Wert, der als Grenzwert für die statistische Signifikanz verwendet wird. Für die folgenden Auswertungen des Plackett-Burman-Versuchsplans wird ein Vertrauensniveau von 90 % angenommen. Daher werden Faktoren mit einem p-Wert von $\gamma \leq 0,10$ als statistisch signifikant eingestuft und im Wahrscheinlichkeitsnetz gesondert gekennzeichnet.

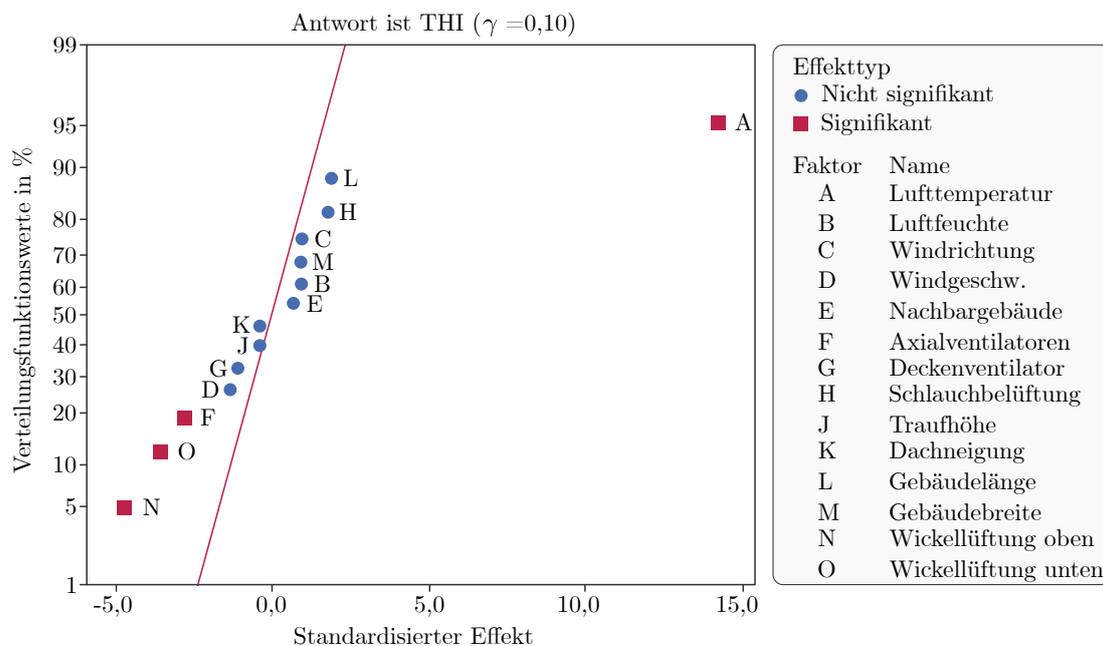


Abbildung 4.22: Wahrscheinlichkeitsnetz des Plackett-Burman-Versuchsplans hinsichtlich des THI ausgewertet in MINITAB (Version 19) (eigene Darstellung)

Die Auswertung des THI im Gehege zeigt für die Faktoren Lufttemperatur, Wickellüftung oben, Wickellüftung unten und Axialventilatoren eine statistische Signifikanz. Die Lufttemperatur weist den größten standardisierten Effekt von 14,2 auf. Eine hohe Lufttemperatur bedeutet also einen erwartungsgemäß hohen THI. Die obere Wickellüftung weist mit $-4,7$ den größten standardisierten Effekt zur Minderung des THI auf, was einer Optimierung des Tierwohls entspricht. Für die untere Wickellüftung ergibt sich mit $-3,6$ ebenfalls ein signifikanter standardisierter Effekt zur Verbesserung des Tierwohls. Die Axiallüfter weisen einen signifikanten Effekt von $-2,8$ auf. Damit auch Faktoreinstellungen berechnet werden können, die nicht im Versuchsplan enthalten sind, wird eine Regressionsgleichung benötigt. Die Regressionsgleichung enthält den Mittelwert des THI und je Faktor einen additiven oder subtraktiven Term. Die Werte der Faktoren sind für die Regressionsgleichung normiert, weshalb eine hohe Einstellung des Faktors mit $+1$ und eine niedrige Einstellung mit -1 angegeben wird. Zwischenwerte der Faktoren sind entsprechend umzurechnen, um sie in die Regressionsgleichung einzusetzen:

$$\begin{aligned}
 THI \approx & 63,25 + 17,85 \cdot T + 1,15 \cdot \varphi + 1,15 \cdot \alpha - 1,65 \cdot v_m(10) + 0,85 \cdot l_{NG} - 3,55 \cdot V_A \\
 & - 1,35 \cdot V_D + 2,15 \cdot V_S - 0,55 \cdot h_G - 0,55 \cdot \beta + 2,35 \cdot l_G + 1,15 \cdot b_G \\
 & - 5,95 \cdot W_o - 4,55 \cdot W_u .
 \end{aligned} \tag{4.5}$$

Abbildung 4.23 stellt das Wahrscheinlichkeitsnetz der Auswertung des *Screenings* für die Netto-Bilanz des emittierten Methans dar. Das Vertrauensniveau von 90 % wurde beibehalten.

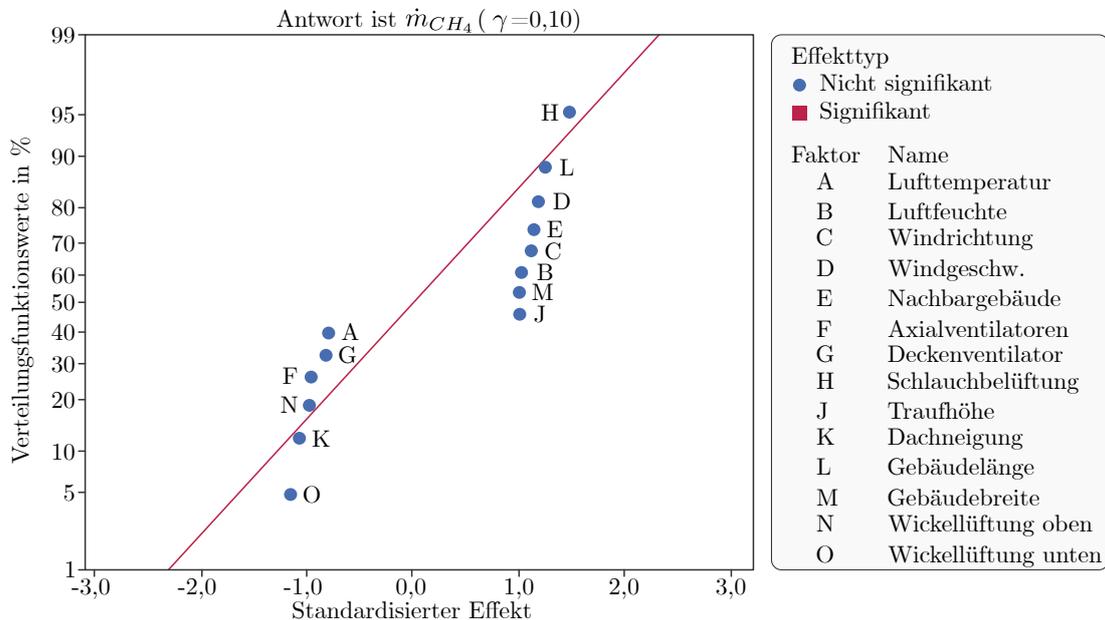


Abbildung 4.23: Wahrscheinlichkeitsnetz des *Screening*-Versuchsplans hinsichtlich des emittierten CH_4 ausgewertet in MINITAB (Version 19) (eigene Darstellung)

Die statistische Auswertung der standardisierten Effekte auf die Methanemission weist keine statistische Signifikanz für die untersuchten Faktoren auf. Eine besondere Beobachtung

stellt der hohe standardisierte Effekt des Dachneigungswinkels auf das emittierte Methan dar. Ein solcher Faktor ließe sich für jeden Neubau berücksichtigen, unabhängig von der (genehmigten) Grundfläche des Gebäudes.

Die Regressionsgleichung ermöglicht die Berechnung der Zwischenwerte des je Kuh normierten Massestroms an emittiertem Methan (in $\frac{\text{kg}}{\text{h} \cdot \text{Kuh}}$). Die Werte der Faktoren sind erneut zwischen -1 und $+1$ normiert und dahingehend umzurechnen:

$$\begin{aligned} \dot{m}_{\text{CH}_4} \approx & 2,21 \cdot 10^{-2} - 5,15 \cdot 10^{-3} \cdot T + 6,66 \cdot 10^{-3} \cdot \varphi + 7,22 \cdot 10^{-3} \cdot \alpha + 7,57 \cdot 10^{-3} \cdot v_m(10) \\ & + 7,29 \cdot 10^{-3} \cdot l_{NG} - 6,24 \cdot 10^{-3} \cdot V_A - 5,33 \cdot 10^{-3} \cdot V_D + 9,46 \cdot 10^{-3} \cdot V_S \\ & + 6,47 \cdot 10^{-3} \cdot h_G - 6,91 \cdot 10^{-3} \cdot \beta + 8,05 \cdot 10^{-3} \cdot l_G + 6,49 \cdot 10^{-3} \cdot b_G \\ & - 6,31 \cdot 10^{-3} \cdot W_o - 7,48 \cdot 10^{-3} \cdot W_u . \end{aligned} \quad (4.6)$$

In Abbildung 4.24 ist das Wahrscheinlichkeitsnetz für die Auswertung der Luftwechselrate visualisiert. Die Auswertung des statistischen Versuchsplans weist für keinen der untersuchten Parameter eine statistische Signifikanz der Luftwechselrate auf. Dennoch zeigt sich auch für die Luftwechselrate ein hoher standardisierter Effekt für den Dachneigungswinkel.

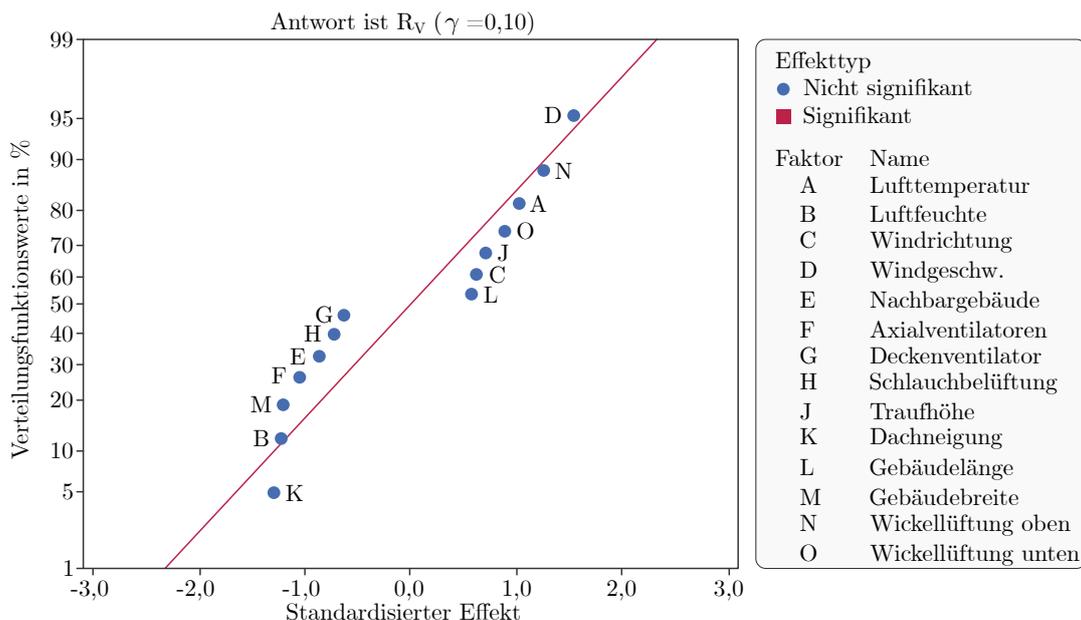


Abbildung 4.24: Wahrscheinlichkeitsnetz der Luftwechselrate R_V ausgewertet in MINITAB (Version 19) (eigene Darstellung)

Erneut lässt sich eine Regressionsgleichung bilden, um die Zielgröße für beliebige Faktoreinstellungen zu berechnen. Für die Luftwechselrate ergibt sich folgende Regressionsgleichung, mit den anzugebenden normierten Werten der Faktoren zwischen -1 und $+1$:

$$\begin{aligned} R_V \approx & 30,8 + 18,3 \cdot T - 22,1 \cdot \varphi + 11,2 \cdot \alpha + 27,5 \cdot v_m(10) - 15,5 \cdot l_{NG} - 18,5 \cdot V_A \\ & - 11,5 \cdot V_D - 13,0 \cdot V_S + 12,8 \cdot h_G - 23,1 \cdot \beta + 10,2 \cdot l_G - 21,5 \cdot b_G \\ & + 22,5 \cdot W_o + 16,0 \cdot W_u . \end{aligned} \quad (4.7)$$

4.6.3 Antwortflächen-Versuchsplan und Auswertung

Im Zuge des *Screenings* wurden 14 Faktoren auf eine statistische Signifikanz hinsichtlich der in Kapitel 4.6.1 definierten Zielgrößen untersucht. Es konnten erste Erkenntnisse gewonnen werden, die eine Einschränkung der Faktoren für den sich anschließenden Box-Behnken-Versuchsplan zulassen. Der Box-Behnken-Versuchsplan wird genutzt, um ein voll quadratisches Modell ausgewählter Faktoren zu erstellen und die Wechselwirkungen untereinander zu untersuchen. Ein Box-Behnken-Versuchsplan ist dabei effizienter als ein zentral zusammengesetzter Versuchsplan oder gar ein dreistufiger vollfaktorieller Versuchsplan. Die geringere Anzahl der erforderlichen Versuchsdurchläufe führt zu einem reduzierten numerischen Aufwand. Für die einzelnen Versuchsdurchläufe wird jeder Faktor auf einen von drei gleichmäßig verteilten Werten gesetzt, die mit $-$, o und $+$ kodiert sind (s. Tabelle 4.5).

Tabelle 4.5: Statistischer Box-Behnken-Versuchsplan und die ausgewerteten Ergebnisse für die Faktoren Wickellüftung W , den Dachneigungswinkel β und der Basisgeschwindigkeit $v_m(10)$, erstellt mittels MINITAB (Version 19)

N	W	β	$v_m(10)$	THI	\dot{m}_{CH_4} in $\frac{\text{kg}}{\text{h} \cdot \text{Küh}}$	R_V in $\frac{1}{\text{h}}$
1 ^a	-	-	o	54,40	0,01308	20,78
2	+	-	o	56,07	0,01494	69,32
3	-	+	o	56,72	0,01383	17,89
4	+	+	o	54,07	0,01527	57,41
5	-	o	-	54,83	0,01395	4,45
6	+	o	-	54,89	0,01539	16,24
7	-	o	+	54,61	0,01344	39,59
8	+	o	+	54,35	0,01449	111,91
9 ^a	o	-	-	54,53	0,01455	11,18
10	o	+	-	54,30	0,01506	9,21
11 ^a	o	-	+	54,11	0,01380	78,26
12	o	+	+	53,93	0,01452	64,55
13	o	o	o	54,06	0,01479	40,18

^a Stationäre Zielgrößen, bei $\epsilon > 1 \cdot 10^{-4}$.

Aufgrund der Erkenntnisse aus dem *Screening* ist die Wickellüftung als ein Faktor für die Untersuchung einbezogen. Im Vergleich mit den anderen Lüftungselementen bietet die Wickellüftung eine energieeffiziente Möglichkeit zur Optimierung, da die Energieaufnahme der Wickellüftung auf die Dauer des Verstellens beschränkt ist. Die Wickellüftungen oben und unten sind zu einem Faktor zusammengefasst, da diese eine ähnliche standardisierte Effektstärke mit dem gleichen Vorzeichen hinsichtlich der Auswirkung auf die Zielgröße zeigten. Die Wickellüftungen oben und unten sind symmetrisch mit $W \in \{30; 60; 90\}$ % variiert.

Für den Dachneigungswinkel wurde beobachtet, dass ein hoher standardisierter Effekt bezüglich des Emissionsausstoßes und der Luftwechselrate auftritt. Dieser Effekt ist von besonderer Relevanz, da der Dachneigungswinkel bereits während der Planung eines frei belüfteten Kuhstalls berücksichtigt werden kann, unabhängig von der genehmigten Grundfläche. Angesichts dessen ist dieser Faktor ebenfalls in die detaillierte Untersuchung einbezogen und auf $\beta \in \{15; 20; 25\}^\circ$ eingestellt. Da die Strömungscharakteristik im Stall maßgeblich durch den von außen aufgeprägten Wind bestimmt wird, ist die Basisgeschwindigkeit zehn Meter über dem Grund als dritter Faktor im Versuchsplan implementiert. Die Einstellungen des Faktors $v_m(10) \in \{0,5; 2,0; 3,5\} \frac{m}{s}$ berücksichtigen, dass niedrigere Windgeschwindigkeiten im Hinblick auf das Tierwohl und die Luftwechselrate kritischer einzuordnen sind.

Erneut wird der Versuchsplan für die drei in Kapitel 4.6.2 benannten Zielgrößen des THI, des normierten Emissionsstroms \dot{m}_{CH_4} und der Luftwechselrate R_V in einem Wahrscheinlichkeitsnetz ausgewertet. Aufgrund der geringeren Spannweite der Faktoreinstellungen ist das Vertrauensniveau für die Auswertung des Box-Behnken-Versuchsplans auf 95 % angehoben, woraus ein Signifikanzniveau von $\gamma \leq 0,05$ resultiert.

Auswertung zum THI

Bei der Auswertung des THI zeigt sich, dass sowohl die Wechselwirkung zwischen der Wickellüftung und dem Dachneigungswinkel, als auch der quadratische Term der Wickellüftung einen statistisch signifikanten Einfluss haben (s. Abbildung 4.25).

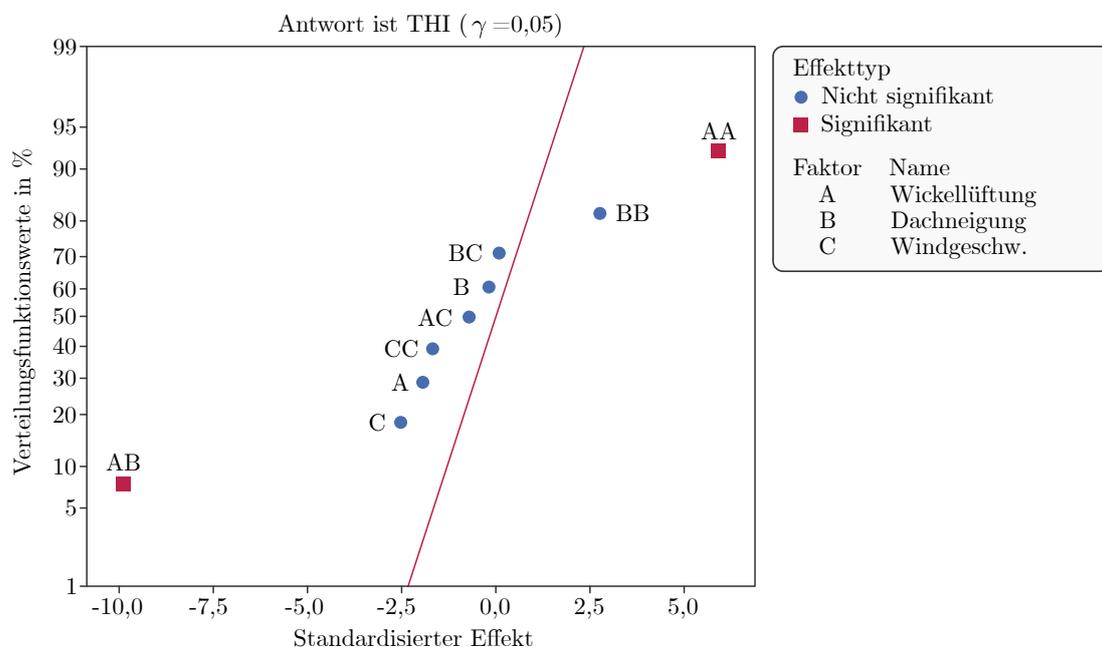


Abbildung 4.25: Wahrscheinlichkeitsnetz der Ergebnisse des Box-Behnken-Versuchsplans gemäß Auswertung des Temperatur-Luftfeuchte-Indexes in MINITAB (Version 19) (eigene Darstellung)

Auf Basis der Ergebnisse des Versuchsplans kann erneut eine Regressionsfunktion gebildet werden, mit welcher eine Abschätzung der Zielgröße für beliebige Kombinationen der drei Hauptfaktoren zu berechnen ist. Die Terme der Regressionsfunktion beinhalten quadratische Terme und Wechselwirkungsterme. Hierbei gilt es erneut zu beachten, dass z. B. die niedrige Einstellung des Faktors $W = 0,3$ mit einem Wert von -1 in die Regressionsfunktion eingeht, wohingegen eine Einstellung von $W = 0,6$ mit $0,0$ und $W = 0,9$ mit dem Wert $+1$ eingeht. Zwischenwerte sind entsprechend zu berechnen und einzusetzen:

$$\begin{aligned} THI \approx & 54,060 - 0,1475 \cdot W - 0,0113 \cdot \beta - 0,1937 \cdot v_m(10) + 0,854 \cdot W^2 + 0,401 \cdot \beta^2 \\ & - 0,244 \cdot v_m(10)^2 - 1,080 \cdot W \cdot \beta - 0,080 \cdot W \cdot v_m(10) + 0,013 \cdot \beta \cdot v_m(10) . \end{aligned} \quad (4.8)$$

In Abbildung 4.26 sind die Antwortflächen zur Zielgröße THI räumlich dargestellt. Hierbei sind jeweils zwei Faktoren variiert, während der dritte Faktor auf der mittleren Einstellung belassen wird. Die Wechselwirkung zwischen dem Dachneigungswinkel und der Wickellüftung ist in der linken Antwortfläche zu erkennen, bei der das Minimum quer im Raum verläuft. Die quadratische Natur der Funktion ist ebenfalls an der Form der Antwortfläche abzulesen.

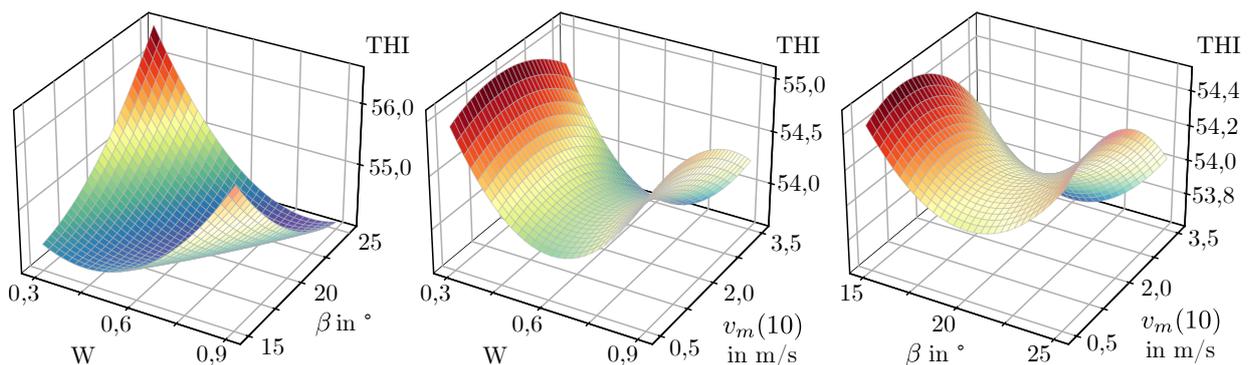


Abbildung 4.26: Antwortflächen der Faktoren des Box-Behnken-Versuchsplans gemäß Auswertung nach Tierwohl-Index THI (eigene Darstellung)

Auswertung zum Emissionsausstoß

Das Wahrscheinlichkeitsnetz für die Antwort des emittierten Methans weist eine statistische Signifikanz für die drei Hauptfaktoren der Untersuchung aus (s. Abbildung 4.27). Wechselwirkungseffekte oder quadratische Terme werden nicht als statistisch signifikant eingestuft.

Anhand der folgenden Regressionsfunktion können die Emissionen für beliebige Kombinationen der drei untersuchten Faktoren abgeschätzt werden:

$$\begin{aligned} \dot{m}_{CH_4} \approx & 1,48 \cdot 10^{-2} - 7,20 \cdot 10^{-4} \cdot W - 3,00 \cdot 10^{-4} \cdot \beta - 3,30 \cdot 10^{-4} \cdot v_m(10) \\ & + 3,30 \cdot 10^{-4} \cdot W^2 + 1,80 \cdot 10^{-4} \cdot \beta^2 - 1,20 \cdot 10^{-4} \cdot v_m(10)^2 \\ & - 9,00 \cdot 10^{-5} \cdot W \cdot \beta - 9,00 \cdot 10^{-5} \cdot W \cdot v_m(10) + 6,00 \cdot 10^{-5} \cdot \beta \cdot v_m(10) . \end{aligned} \quad (4.9)$$

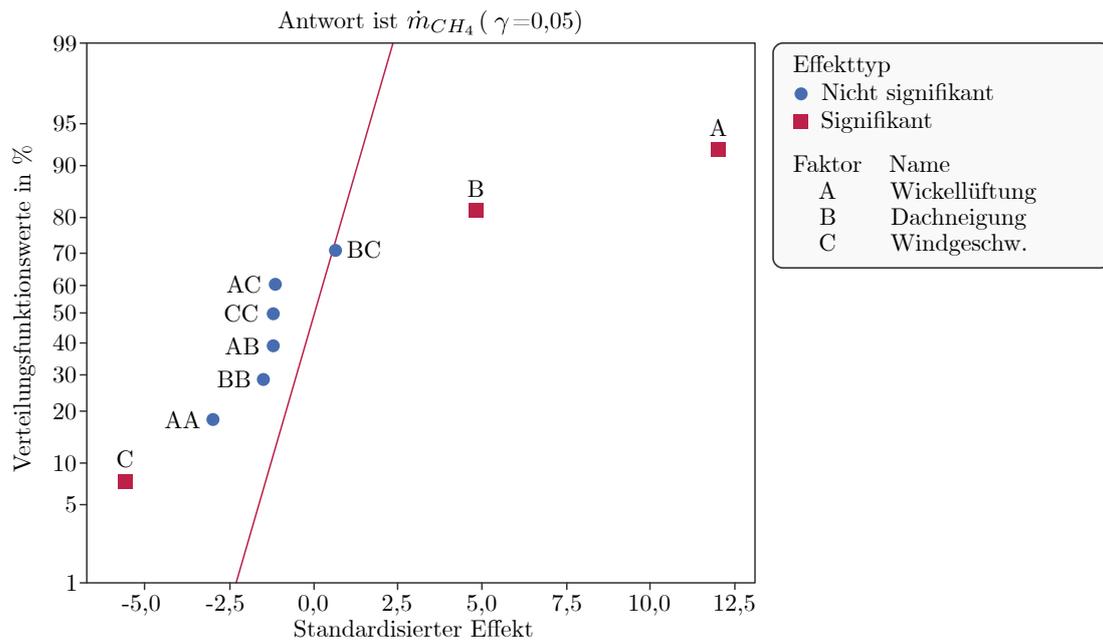


Abbildung 4.27: Wahrscheinlichkeitsnetz des Box-Behnken-Versuchsplans hinsichtlich des emittierten CH_4 ausgewertet in MINITAB (Version 19) (eigene Darstellung)

Die Antwortflächen der ausgewerteten Emissionsströme zeigen, dass eine hohe Einstellung der Wickellüftung zu einem korrespondierend höheren Emissionsausstoß führt (s. Abbildung 4.28). Dies ist bereits durch den hohen standardisierten Effekt im Wahrscheinlichkeitsnetz ersichtlich.

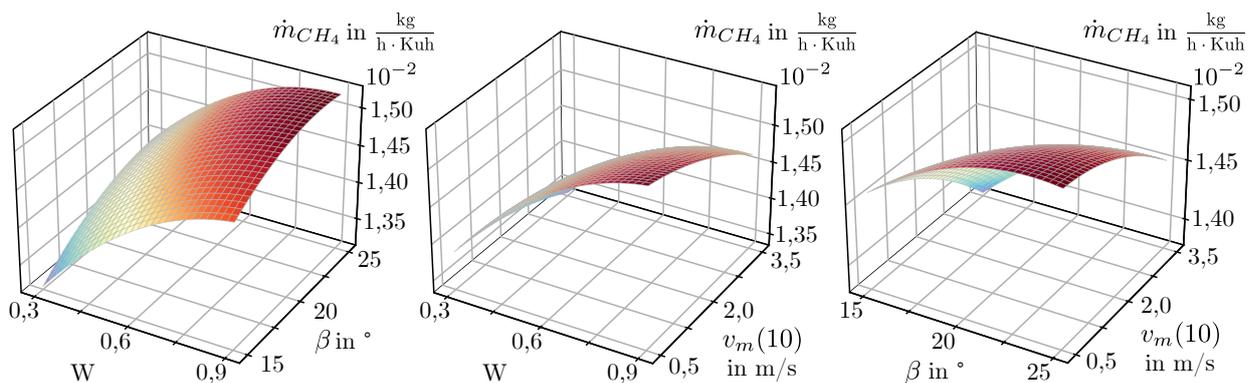


Abbildung 4.28: Antwortflächen der Faktoren des Box-Behnken-Versuchsplans gemäß Auswertung nach Emissionsfluss \dot{m}_{CH_4} (eigene Darstellung)

Auswertung zur Luftwechselrate

Das Wahrscheinlichkeitsnetz der Luftwechselrate zeigt neben den drei Hauptfaktoren auch für die Wechselwirkung zwischen Wickellüftung und Windgeschwindigkeit eine statistische Signifikanz (s. Abbildung 4.29).

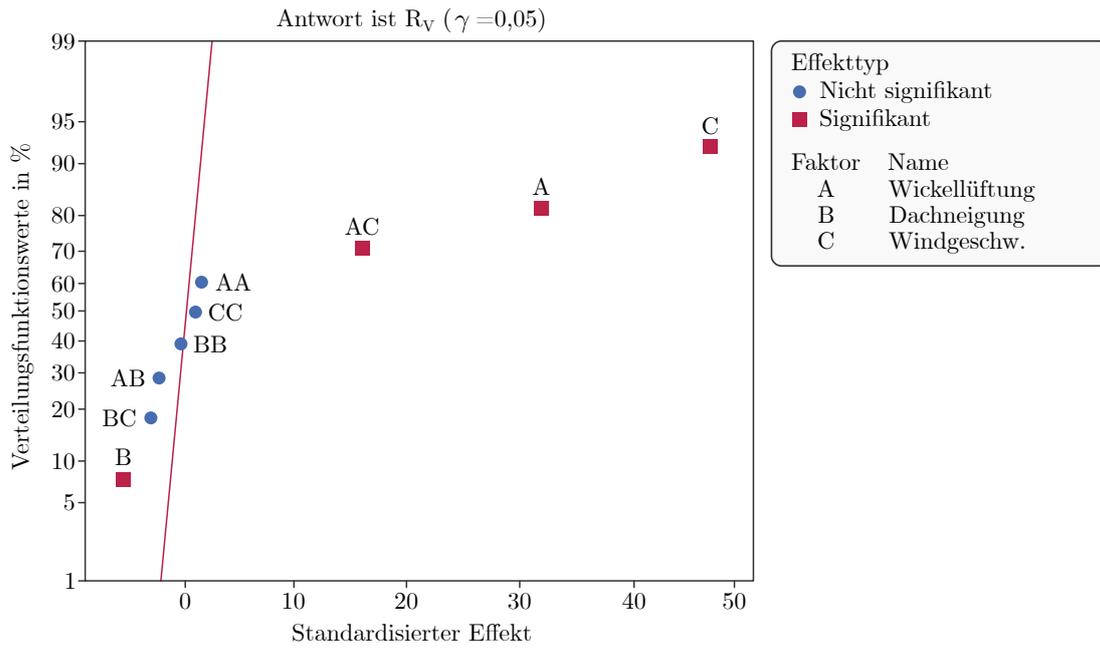


Abbildung 4.29: Wahrscheinlichkeitsnetz der Ergebnisse des Box-Behnken-Versuchsplans gemäß Auswertung der Luftwechselrate in MINITAB (Version 19) (eigene Darstellung)

Die räumlichen Antwortflächen in Abbildung 4.30 stellen die statistische Signifikanz der extern aufgeprägten Windgeschwindigkeit für die Luftwechselrate R_V deutlich heraus. Die schief im Raum verlaufende Neigung der Ebene der mittleren Antwortfläche illustriert außerdem die Wechselwirkung zwischen der Windgeschwindigkeit und Wickellüftung.

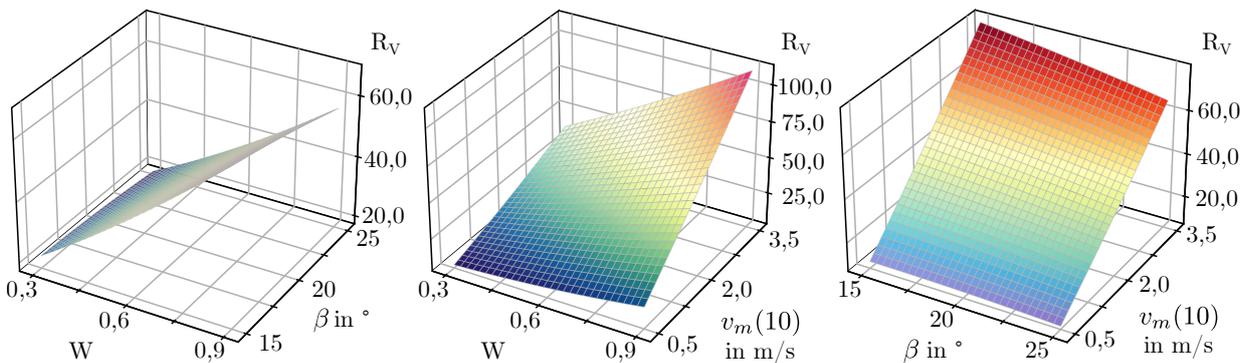


Abbildung 4.30: Antwortflächen der Faktoren des Box-Behnken-Versuchsplans gemäß Auswertung nach Luftwechselrate R_V (eigene Darstellung)

Mithilfe einer Regressionsfunktion können abermals beliebige Kombinationen der drei Faktoren angegeben und die korrespondierende Luftwechselrate abgeschätzt werden:

$$\begin{aligned}
 R_V \approx & 40,180 - 21,521 \cdot W - 3,810 \cdot \beta - 31,654 \cdot v_m(10) + 1,710 \cdot W^2 + 0,540 \cdot \beta^2 \\
 & - 1,160 \cdot v_m(10)^2 - 2,255 \cdot W \cdot \beta - 15,133 \cdot W \cdot v_m(10) + 2,935 \cdot \beta \cdot v_m(10) .
 \end{aligned} \tag{4.10}$$

Zwischenfazit

Die dem *Screening* angeschlossene statistische Versuchsplanung lieferte über die Anwendung eines Antwortflächen-Versuchsplans weitere Erkenntnisse über das Systemverhalten der in Kapitel 4.6.1 definierten Zielgrößen. Bei der Auswertung des THI wurde festgestellt, dass eine Wechselwirkung zwischen dem Dachneigungswinkel und der Wickellüftung besteht. Diese Wechselwirkung zeigt auf, dass eine vollständig geöffnete Wickellüftung nicht bei jedem Dachneigungswinkel auch das optimale Tierwohl gewährleistet. Für einen Dachneigungswinkel von $\beta = 15^\circ$ und Windgeschwindigkeit von $v_m(10) = 2,0 \frac{m}{s}$ ergibt sich, dass die Öffnung der Wickellüftung zu 49,5 % gegenüber der vollständigen Öffnung ein Optimierungspotenzial von ca. 4,4 % ausweist. Eine zunehmend geschlossene Wickellüftung führt außerdem zu einem geringeren Emissionsstrom aus dem Stall und bietet für das o. g. Beispiel ein Potenzial, die an die Umwelt abgeführten Emissionen ebenfalls um 6,9 % zu reduzieren. Bei einem größeren Dachneigungswinkel wird dieses Potenzial zunehmend geringer, da hierbei eine große Öffnung der Wickellüftung für einen niedrigen THI erforderlich wird. Dennoch legt die Auswertung nahe, dass eine in Echtzeit optimierte Luftführung ein Einsparpotenzial für viele Anwendungsfälle eröffnen kann.

Methodenentwicklung für echtzeitfähige Metamodelle von Strömungen

Die aus der statistischen Versuchsplanung resultierenden Regressionsgleichungen beschränken die Analyse der Um- und Durchströmung des Stalls auf eine zahlenwertige Zielgröße. Eine vielversprechende Alternative bieten Physik-informierte neuronale Netze, welche einen auf physikalischen Grundsätzen basierenden datensparsamen Ansatz bieten und räumlich aufgelöste Ergebnisse produzieren sowie Visualisierungen des gesamten Strömungsfeldes ermöglichen. Bislang wurden PINNs jedoch nur für niedrige Reynolds-Zahlen angewendet, während am Stall Reynolds-Zahlen bis zu $5,8 \cdot 10^6$ auftreten. Es bedarf daher weiterer Entwicklungen und Analysen, um PINNs für den Einsatz in ingenieurtechnischen Anwendungen mit komplexeren Geometrien und häufig eingesetzten RANS-Simulationen zu optimieren. Das vorliegende Kapitel befasst sich daher mit der Methodenentwicklung zur Verwendung von PINNs zur Vorhersage von Strömungsergebnissen in und um einen Stall, insbesondere im Hinblick auf variable Eingangsgrößen und die Turbulenzmodellierung. Zusätzlich wird eine Echtzeitfähigkeitsanforderung an das Modell gestellt und nach dem Trainingsprozess geprüft.

Die in diesen Kapiteln vorgestellten Ergebnisse basieren in Teilen auf den Arbeiten des Verfassers dieser Dissertation als Autor einer Veröffentlichung (PIOCH, HARMENING, MÜLLER et al., 2023) sowie als Co-Autor einer weiteren Veröffentlichung (HARMENING, PIOCH, FUHRIG et al., 2023).

5.1 Zusammengesetzte Verlustfunktion

Die Einhaltung bekannter Referenzergebnisse aus Messungen oder Simulationen kann über eine klassische Metrik vorgegeben werden. Für die vorliegende Arbeit wird der mittlere quadratische Fehler verwendet (s. Gleichung 2.46), woraus sich die folgende Formulierung ergibt:

$$\mathcal{L}_d = \frac{1}{N_d} \sum_{n=1}^{N_d} |u_{\text{ref}d}(x_n, y_n) - \tilde{u}_n|^2, \quad (5.1)$$

wobei N_d die Anzahl der Datensätze ist, $u_{\text{ref}d}(x_n, y_n)$ die Referenzdaten an der Koordinate (x_n, y_n) sind und \tilde{u}_n der durch das neuronale Netz approximierte Ausgabewert des n -ten Datensatzes ist.

Die strömungsdynamischen Randbedingungen können über eine Dirichlet-Randbedingung oder eine Neumann-Randbedingung implementiert werden. Die erzwungene Konvektion am Einlass der Strömungsdomäne wäre ein Beispiel für die Dirichlet-Randbedingung, ebenso das Setzen der reibungsbehafteten Wände, durch das Einstellen der Geschwindigkeitskomponenten auf $0,0 \frac{\text{m}}{\text{s}}$. Eine Neumann-Randbedingung bietet sich beispielsweise an, um ein normal zum Einlass verlaufendes Geschwindigkeitsprofil zu erzwingen. Allgemein formuliert führt dies zu:

$$\mathcal{L}_b = \frac{1}{N_b} \sum_{n=1}^{N_b} |u_{\text{ref}b}(x_n, y_n) - \tilde{u}_n|^2, \quad (5.2)$$

wobei N_b für die Anzahl der Randbedingungen, $u_{\text{ref}b}(x_n, y_n)$ für den Referenzwert an der Koordinate (x_n, y_n) und \tilde{u}_n für den vorhergesagten Wert des neuronalen Netzes steht.

Kern der PINNs sind die physikalischen Verlustterme, welche die zu Grunde liegenden Gleichungen inkorporieren. Im Falle strömungsdynamischer Betrachtungen gilt es, die Impulsgleichungen und Massenerhaltung zu implementieren. Für zweidimensionale strömungsdynamische Betrachtungen ergeben sich die Residuen der Massenerhaltungsgleichung und der beiden Impulsgleichungen, welche sich wie folgt zusammenfassen lassen:

$$\mathcal{L}_f = \sum_{k=1}^3 \frac{1}{N_f} \sum_{n=1}^{N_f} |\epsilon_k(x_n, y_n)|^2, \quad (5.3)$$

wobei die Residuen ϵ der Gleichungen k über die Anzahl der Datensätze N_f und die Koordinaten (x_n, y_n) summiert werden.

Das Residuum der Massenerhaltung inkompressibler newtonscher Fluide kann für eine zwei-dimensionale Betrachtung der RANS-Gleichungen wie folgt formuliert werden:

$$\epsilon_1 = \frac{\partial \bar{u}}{\partial x} + \frac{\partial \bar{v}}{\partial y}. \quad (5.4)$$

Für die Impulserhaltungen gilt:

$$\epsilon_2 = \bar{u} \frac{\partial \bar{u}}{\partial x} + \bar{v} \frac{\partial \bar{u}}{\partial y} + \overline{u' \frac{\partial u'}{\partial x}} + \overline{v' \frac{\partial u'}{\partial y}} + \frac{1}{\rho} \frac{\partial \bar{p}}{\partial x} - \nu \left(\frac{\partial^2 \bar{u}}{\partial x^2} + \frac{\partial^2 \bar{u}}{\partial y^2} \right). \quad (5.5)$$

$$\epsilon_3 = \bar{u} \frac{\partial \bar{v}}{\partial x} + \bar{v} \frac{\partial \bar{v}}{\partial y} + \overline{u' \frac{\partial v'}{\partial x}} + \overline{v' \frac{\partial v'}{\partial y}} + \frac{1}{\rho} \frac{\partial \bar{p}}{\partial y} - \nu \left(\frac{\partial^2 \bar{v}}{\partial x^2} + \frac{\partial^2 \bar{v}}{\partial y^2} \right). \quad (5.6)$$

Die zusammengesetzte Verlustfunktion ergibt sich aus der Summation aller Verlustterme der Datensätze, Randbedingungen und Residuen der PDEs. Über die Faktoren λ kann die Optimierung hinsichtlich einzelner Verlustterme gewichtet werden:

$$\mathcal{L} = \sum_{i=0}^N \lambda_i \mathcal{L}_{d,i} + \sum_{j=0}^N \lambda_j \mathcal{L}_{b,j} + \sum_{k=0}^N \lambda_k \mathcal{L}_{f,k}. \quad (5.7)$$

5.2 Trainingsalgorithmus

Der Ablauf des Trainings von PINNs unterscheidet sich gegenüber klassischen neuronalen Netzen in den folgend erläuterten Punkten. Im Falle der Navier-Stokes-Gleichungen ist für die Bildung der physikalischen Verlustterme eine Approximation der Gradienten erforderlich, die der Jacobi- und Hesse-Matrix zwischen Ein- und Ausgang des PINN entnommen werden können. Die in der Jacobi- und Hesse-Matrix enthaltenen Ableitungen erster und zweiter Ordnung sind im Falle der PINNs über die *backpropagation* approximiert. Die Gradienten werden anschließend in die Verlustfunktion eingesetzt und das Residuum gebildet, woraufhin die Gewichte und Biases angepasst werden. Diese Prozedur wird iterativ durchgeführt, bis das Limit der Iterationen, und im Falle des L-BFGS-Optimierers das Konvergenzkriterium erreicht wird. Abbildung 5.1 visualisiert den Ablauf des Trainingsprozesses für ein PINN, welches auf die zweidimensionalen Navier-Stokes-Gleichungen angewendet wird. Je nach Anwendungsfall und Turbulenzmodellierung kann sich die Netzarchitektur an der Eingangs- und Ausgangsschicht unterscheiden.

Die Initialisierung der Gewichte und Biases im PINN erfolgt nach randomisierten oder statistischen Grundsätzen. Wegen dieser Vorgehensweise und der Komplexität der Funktionslandschaften ist es notwendig, das Training mehrfach zu starten und die Modelle zu vergleichen, um die Erreichung eines (lokalen) Optimums sicherzustellen (LU et al., 2021).

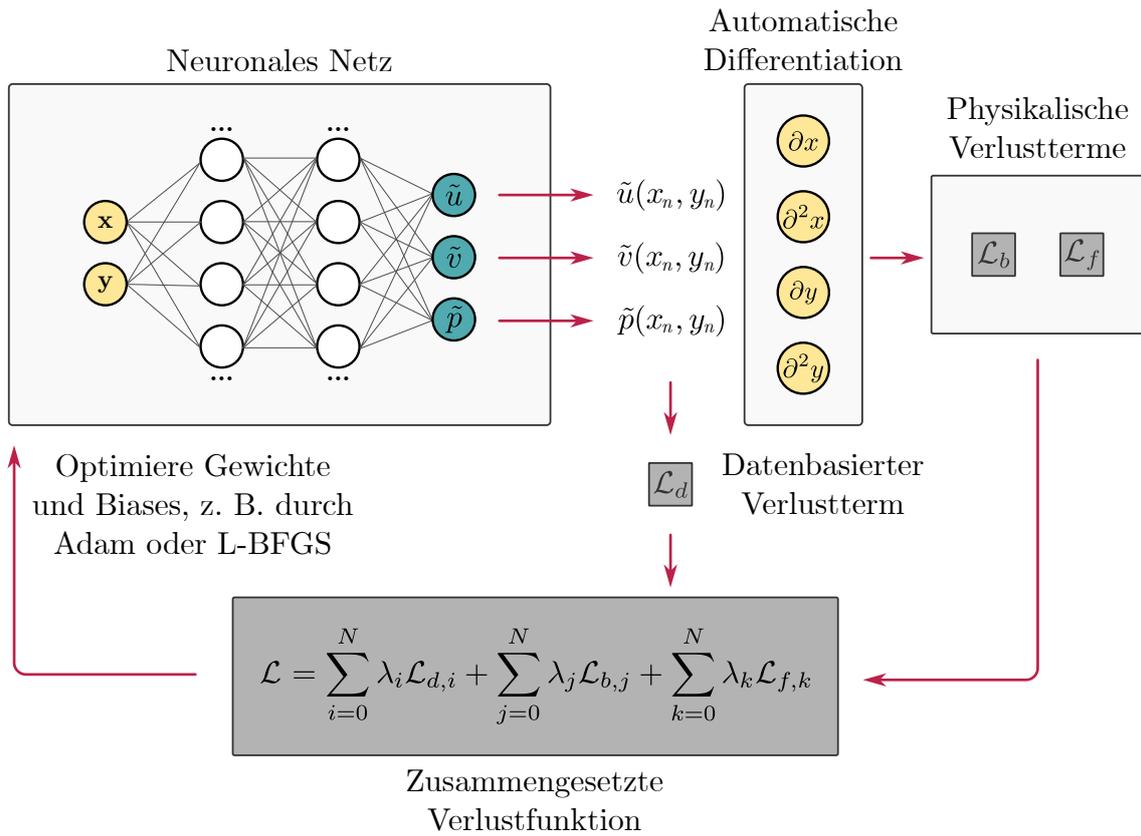


Abbildung 5.1: Darstellung des verallgemeinerten Trainingsalgorithmus für PINNs (eigene Darstellung)

5.3 Turbulenzmodellierung für Physik-informierte neuronale Netze

In der klassischen Strömungsdynamik wird das Schließungsproblem über Ersatzmodelle gelöst, die wiederum bestimmte Voraussetzungen im Lösungsprozess mit sich bringen. Das Mischlängenmodell benötigt zusätzlich den Wandabstand d als Eingangsgröße. Für die Zweigleichungsmodelle ist die Lösung der zusätzlichen Transportgleichungen und teils spezielle Behandlung der Randbedingungen für die transportierten Größen der turbulenten kinetischen Energie, Dissipationsrate oder spezifischen Dissipationsrate erforderlich. Fraglich ist, inwiefern sich die klassischen Ansätze der Turbulenzmodellierung für die Verwendung mittels PINN eignen und ob sich spezifisch für PINNs geeignete Turbulenzmodelle formulieren lassen. Folgend wird dies anhand einer rückwärtsgewandten Stufe untersucht, welche die in Abbildung 5.2 dargestellten Strömungsmerkmale beinhaltet (PIOCH, HARMENING, MÜLLER et al., 2023). Die Eingangsschicht des neuronalen Netzes der Untersuchung enthält die Eingangsgrößen x , y und d . Es sind fünf verdeckte Schichten mit je 128 Neuronen angelegt. Die Ausgangsschicht unterscheidet sich je Turbulenzmodell und wird daher in den folgenden

Kapiteln detailliert beschrieben. Als Aktivierungsfunktion ist der *Tangens hyperbolicus* festgelegt. Die Gewichte und Biases sind mittels *Glorot Uniform* initialisiert, um zu Beginn der Optimierung eine möglichst gleichmäßige Varianz über alle verdeckten Schichten zu gewährleisten und die Tendenz zur Divergenz der Gewichte zu vermeiden. Die ersten 10.000 Epochen sind mittels Adam bei einer Lernrate von $1 \cdot 10^{-3}$, weitere 20.000 Epochen via Adam mit einer Lernrate von $1 \cdot 10^{-4}$ und abschließend mit dem L-BFGS-Verfahren optimiert. Das neuronale Netz wurde je Untersuchung zehnmal neu initialisiert, um die stochastischen Effekte auf den Optimierungsprozess zu minimieren und ein lokales Optimum für die Untersuchung aller Turbulenzmodelle zu gewährleisten.

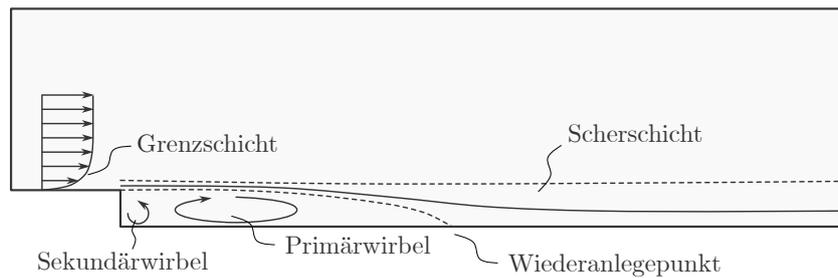


Abbildung 5.2: Strömungsmerkmale an einer rückwärtsgewandten Stufe, in Anlehnung an PIOCH, HARMENING, MÜLLER et al. (2023)

Für das Training des PINN auf die Strömung der beschriebenen rückwärtsgewandten Stufe sind Randbedingungen erforderlich, die den Einlass und die Wandbehandlung betreffen. Am Einlass ist die Strömungsgeschwindigkeit einer direkten numerischen Simulation (DNS) von LE, MOIN und J. KIM (1997), mittels Dirichlet-Randbedingung, bei $x/h = 0$ vorgegeben. Hierbei gilt es zu beachten, dass die Daten der Untersuchung von LE, MOIN und J. KIM (1997) in das hier verwendete, modifizierte Koordinatensystem transformiert wurden. Eine Neumann-Randbedingung erzwingt ein am Einlass normal verlaufendes Geschwindigkeitsprofil. Die unteren Begrenzungswände und die Stufe sind reibungsbehaftet mit einer Dirichlet-Randbedingung via $\bar{u} = 0$ und $\bar{v} = 0$ implementiert. Punktrandbedingungen mit einer Vorgabe der Simulationsdaten für \bar{u} , \bar{v} und \bar{p} sind im Folgenden für drei von fünf vorhandenen Liniendatensätzen vorgegeben, die bei $x/h \in \{7; 13; 22\}$ liegen. Die physikalischen Randbedingungen sind gemäß Kapitel 5.1 vorgegeben. Abbildung 5.3 visualisiert die Strömungsdomäne der DNS und des PINN sowie die benannten Randbedingungen.

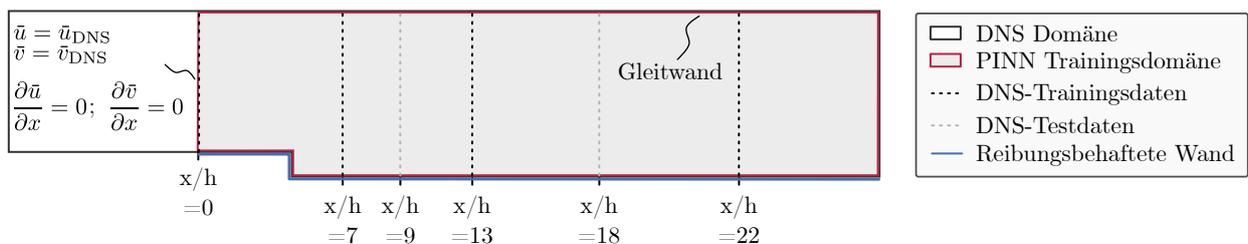


Abbildung 5.3: Randbedingungen für das Training eines PINN an einer rückwärtsgewandten Stufe, in Anlehnung an PIOCH, HARMENING, MÜLLER et al. (2023)

5.3.1 Physik-informierte neuronale Netze und das Mischlängenmodell

Das Null-Gleichungsmodell kann für Physik-informierte neuronale Netze gemäß HENNIGH et al. (2021) verwendet werden, indem die turbulente Viskosität gemäß Gl. 2.23 berechnet wird. Die Mischlänge berechnet sich gemäß HENNIGH et al. (2021) zu $l_m = \min(0,419 \cdot d; 0,09 \cdot d_{\max})$, wobei d_{\max} dem maximalen Wandabstand entspricht. Der Wandabstand kann an diskreten Koordinaten der Simulation entnommen werden, womit wiederum die Möglichkeiten des Trainings auf diese Punkte beschränkt sind. Um jedoch die Möglichkeit für ein Training des PINN an allen Punkten im Raum zu ermöglichen, wurde ein weiteres neuronales Netz zur Vorhersage des Wandabstandes in den Trainingsablauf implementiert. Die Genauigkeit der Vorhersage des zweiten neuronalen Netzes für den Wandabstand liegt bei der Auswertung mit einem Testdatensatz bei einem MSE von $1,34 \cdot 10^{-6}$. In Abbildung 5.4 ist das mittels PINN vorhergesagte Strömungsfeld als Konturdiagramm und das Liniendiagramm an der Stelle $x/h = 9,0$ dargestellt. Zu erkennen ist, dass das Mischlängenmodell eine Geschwindigkeitsüberhöhung im Feld über und im näheren Bereich hinter der Stufe zeigt. Der Primärwirbel ist zu erkennen, der Sekundärwirbel wird vom PINN nicht vorhergesagt. Es zeigt sich ein Fluss durch die Wand im Bereich der Stufe. Das Liniendiagramm weist einen absoluten Fehler von $\max. 8,02 \cdot 10^{-3} \frac{\text{m}}{\text{s}}$ aus.

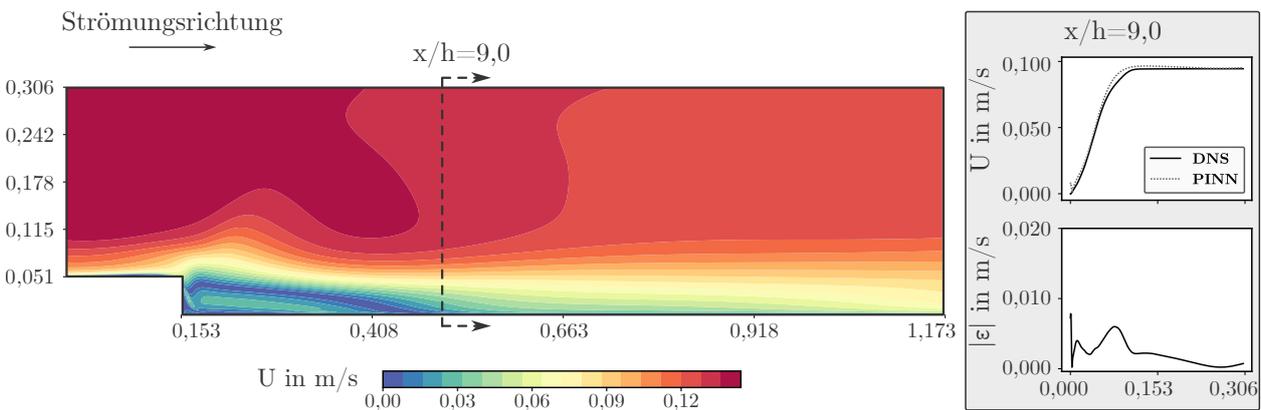


Abbildung 5.4: Vorhersage des PINN im Strömungsfeld für die rückwärts-gewandte Stufe mit einem Mischlängenmodell (links) und der Vergleich mit DNS-Daten (rechts) (eigene Darstellung)

5.3.2 Implementierung des k - ω -Modells für Physik-informierte neuronale Netze

Das k - ω -Modell ist in der CFD-Simulation weitverbreitet und bietet für viele Anwendungsfälle gute Übereinstimmungen mit Messergebnissen oder hochgenauen numerischen Lösungen. Für die Verwendung in PINNs muss die Ausgabeschicht um die Größen der turbulenten kinetischen Energie k und der spezifischen Dissipationsrate ω ergänzt werden. Über die automatische Differentiation können die erforderlichen Ableitungen der durch das neuronale

Netz approximierten Größen \tilde{k} und $\tilde{\omega}$ gebildet werden, um das Residuum der zwei zusätzlichen Transportgleichungen zu bestimmen. Die Residuen der Transportgleichungen ergeben sich gemäß Kapitel 2.2.3 für eine zweidimensionale Strömung zu

$$\begin{aligned} \epsilon_k = & \tau'_{xx} \frac{\partial \bar{u}}{\partial x} + \tau'_{xy} \frac{\partial \bar{u}}{\partial y} + \tau'_{yx} \frac{\partial \bar{v}}{\partial x} + \tau'_{yy} \frac{\partial \bar{v}}{\partial y} - C_{\beta^* k} \omega + \frac{\partial}{\partial x} \left((\nu + C_{\sigma^*} \nu_t) \frac{\partial k}{\partial x} \right) \\ & + \frac{\partial}{\partial y} \left((\nu + C_{\sigma^*} \nu_t) \frac{\partial k}{\partial y} \right) - \bar{u} \frac{\partial k}{\partial x} - \bar{v} \frac{\partial k}{\partial y} \end{aligned} \quad (5.8)$$

und

$$\begin{aligned} \epsilon_\omega = & C_\alpha \frac{k}{\omega} \left(\tau'_{xx} \frac{\partial \bar{u}}{\partial x} + \tau'_{xy} \frac{\partial \bar{u}}{\partial y} + \tau'_{yx} \frac{\partial \bar{v}}{\partial x} + \tau'_{yy} \frac{\partial \bar{v}}{\partial y} \right) - C_\beta \omega^2 \\ & + \frac{\partial}{\partial x} \left((\nu + C_\sigma \nu_t) \frac{\partial \omega}{\partial x} \right) + \frac{\partial}{\partial y} \left((\nu + C_\sigma \nu_t) \frac{\partial \omega}{\partial y} \right) - \bar{u} \frac{\partial \omega}{\partial x} - \bar{v} \frac{\partial \omega}{\partial y}, \end{aligned} \quad (5.9)$$

mit $C_{\beta^*} = 9/100$, $C_{\sigma^*} = C_\sigma = 0,5$, $C_\alpha = 5/9$ und $C_\beta = 3/40$, wie von WILCOX (1988) empfohlen. Bei Anwendung auf eine rückwärtsgewandte Stufe ergibt sich das in Abbildung 5.5 dargestellte Ergebnis. Auch das k - ω -Modell zeigt gegenüber den Referenzdaten eine Überhöhung der Geschwindigkeit außerhalb der Grenzschicht und oberhalb der Stufe. Der Primärwirbel ist zu erkennen, aber durch eine überhöhte Geschwindigkeit der wandnahen Rückströmung gekennzeichnet. Es ist ein Fluss durch die vertikale Wand der Stufe sichtbar, welcher die implementierten Randbedingungen verletzt. Der max. absolute Fehler im Liniendiagramm bei $x/h = 9,0$ beträgt $8,24 \cdot 10^{-3} \frac{\text{m}}{\text{s}}$.

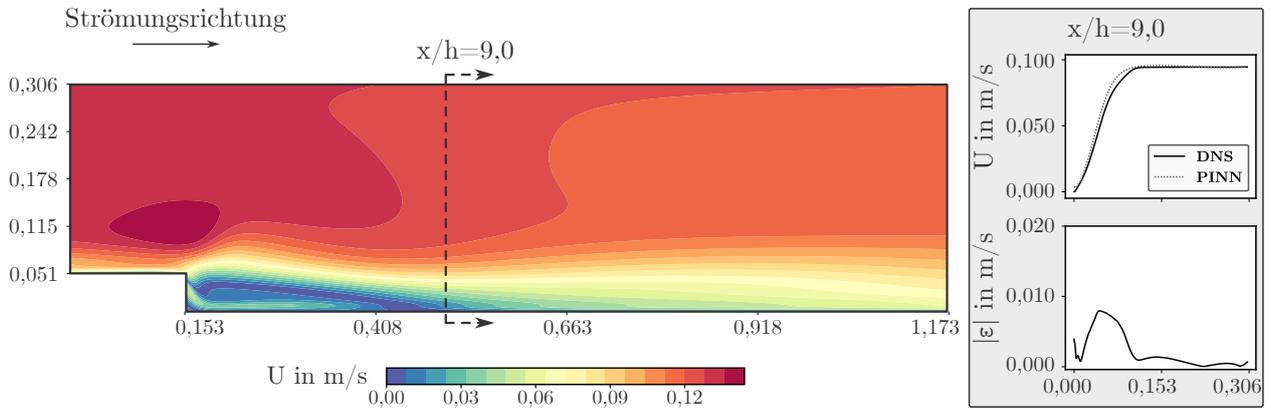


Abbildung 5.5: Vorhersage des PINN im Strömungsfeld für die rückwärtsgewandte Stufe mit einem k - ω -Modell (links) und der Vergleich mit DNS-Daten (rechts) (eigene Darstellung)

5.3.3 Turbulenzmodellierung mit dem ν_t -Modell

Ein Ansatz zur Turbulenzmodellierung für PINNs besteht darin, dass der Ausgabeschicht die turbulente Viskosität ν_t hinzugefügt wird, um diese mittels Optimierung der Gewichte

und Biases direkt durch das PINN bestimmen zu lassen (H. XU, W. ZHANG und Y. WANG, 2021). Die Methode kann als gleichungsfreier Ansatz bezeichnet werden, da die turbulente Viskosität in den RANS-Gleichungen direkt durch das PINN bestimmt wird, ohne eine ergänzende mathematische Umformung oder Berechnung. Die Ergebnisse dieses Ansatzes zeigen für die rückwärtsgewandte Stufe eine Überhöhung des Geschwindigkeitsfeldes oberhalb der Stufe. Der Primärwirbel ist zu erkennen, wobei abermals eine zu hohe wandnahe rückläufige Geschwindigkeit vorausgesagt wird. Der Sekundärwirbel wird durch das PINN nicht prädiziert. Abbildung 5.6 stellt dies in einem Konturdiagramm dar. Das Modell weist entlang der Testdaten bei $x/h = 9,0$ einen max. absoluten Fehler von $1,28 \cdot 10^{-2} \frac{\text{m}}{\text{s}}$ auf.

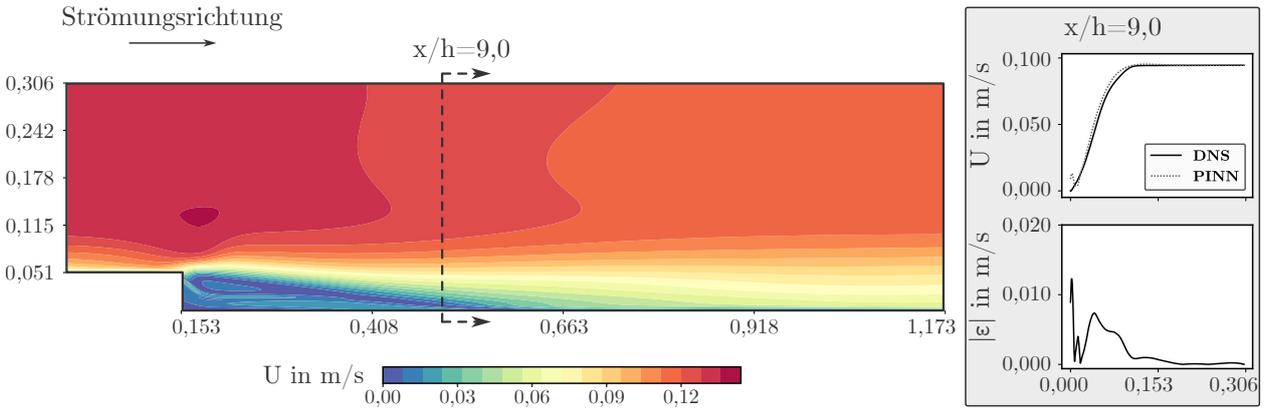


Abbildung 5.6: Vorhersage des PINN zur rückwärtsgewandten Stufe mit dem ν_t -Modell (links) und der Vergleich mit DNS-Daten (rechts) (eigene Darstellung)

5.3.4 Turbulenzmodellierung via pseudo-Reynolds-Spannungen

Aus der Herleitung der RANS-Gleichungen folgt der Ausdruck für den Reynolds-Spannungsterm (siehe Gleichung 2.20), dessen Komponenten für eine zweidimensionale Strömung mit den gemittelten Geschwindigkeitskomponenten \bar{u} und \bar{v} lauten:

$$\frac{\partial \tau'_{xx}}{\partial x} + \frac{\partial \tau'_{xy}}{\partial y} = \overline{u' \frac{\partial u'}{\partial x}} + \overline{v' \frac{\partial u'}{\partial y}} \quad (5.10)$$

und

$$\frac{\partial \tau'_{yx}}{\partial x} + \frac{\partial \tau'_{yy}}{\partial y} = \overline{u' \frac{\partial v'}{\partial x}} + \overline{v' \frac{\partial v'}{\partial y}}. \quad (5.11)$$

Aus der Hypothese, dass pseudo-turbulente zeitinvariante Größen der Geschwindigkeitskomponenten durch das PINN vorhergesagt und die o. g. Terme ersetzen können, folgen die Gleichungen (PIOCH, HARMENING, MÜLLER et al., 2023):

$$\frac{\partial \tau'_{xx}}{\partial x} + \frac{\partial \tau'_{xy}}{\partial y} = u'_p \frac{\partial u'_p}{\partial x} + v'_p \frac{\partial u'_p}{\partial y} \quad (5.12)$$

und

$$\frac{\partial \tau'_{yx}}{\partial x} + \frac{\partial \tau'_{yy}}{\partial y} = u'_p \frac{\partial v'_p}{\partial x} + v'_p \frac{\partial u'_p}{\partial y}. \quad (5.13)$$

Das Turbulenzmodell implementiert Terme, welche mittels automatisierter Differentiation berechnet werden und dem Äquivalent der Reynolds-Spannungen entsprechen, ohne dabei physikalischen Prinzipien zu folgen, da u'_p und v'_p zeitinvariante Größen sind, für die gilt:

$$u'_p \frac{\partial u'_p}{\partial x} + v'_p \frac{\partial u'_p}{\partial y} \neq \overline{u' \frac{\partial u'}{\partial x}} + \overline{v' \frac{\partial u'}{\partial y}}. \quad (5.14)$$

Das Turbulenzmodell wird daher folgend als pseudo-Reynolds-Spannungs-Turbulenzmodell benannt.

Für die Implementierung des Turbulenzmodells in ein PINN ist die Ergänzung der pseudo-turbulenten zeitinvarianten Größen u'_p und v'_p auf der Ausgabeschicht des neuronalen Netzes erforderlich. Die Terme, zum Ersatz der Reynolds-Spannungen in den Impulsgleichungen, werden über die automatische Differentiation durch die *backpropagation* des neuronalen Netzes berechnet. Einen ähnlichen Ansatz formulieren EIVAZI und VINUESA (2022), mit der Einschränkung des Gültigkeitsbereiches für inkompressible Fluide.

Die Ergebnisse des Turbulenzmodells mittels pseudo-Reynolds-Spannungen weisen ein im Gros übereinstimmendes Geschwindigkeitsfeld mit den DNS-Daten auf (s. Abbildung 5.7). Der Primärwirbel ist sichtbar abgegrenzt und zeigt ein realistisches, wandnahes, rückläufiges Geschwindigkeitsprofil. Im oberen Bereich der Stufe ist ein Fluss durch die Wand sichtbar, und der Sekundärwirbel wird nicht ausgebildet. Das Modell weist entlang der Testdaten bei $x/h = 9,0$ einen max. absoluten Fehler von $3,02 \cdot 10^{-3} \frac{\text{m}}{\text{s}}$ auf.

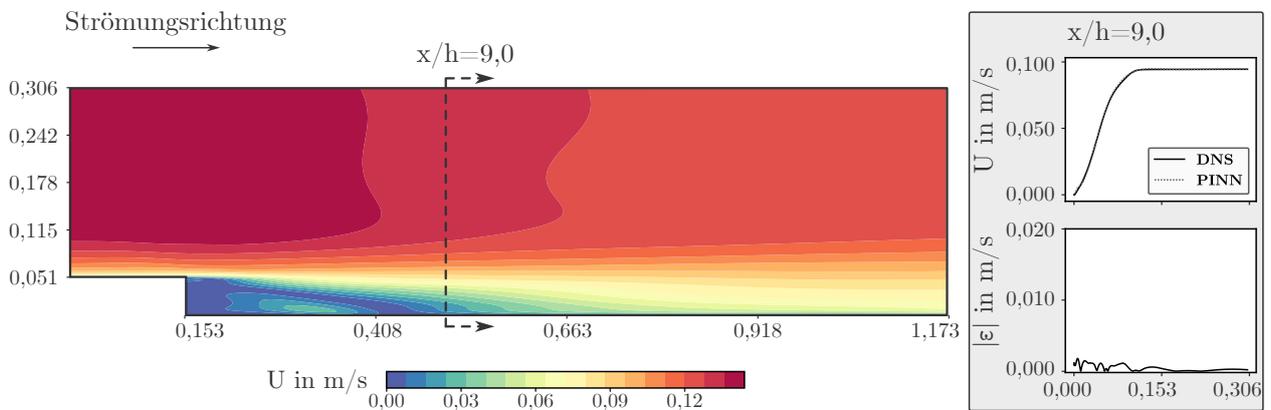


Abbildung 5.7: Vorhersage des PINN im Strömungsfeld für die rückwärts-gewandte Stufe mit einem pseudo-Reynolds-Spannungs-Turbulenzmodell (links) und der Vergleich mit DNS-Daten (rechts) (eigene Darstellung)

5.3.5 Auswertung der Turbulenzmodellierung

Die Ergebnisse der Untersuchungen zeigen, dass die Turbulenzmodellierung mittels pseudo-Reynolds-Spannungen für eine rückwärtsgewandte Stufe und einen unvollständigen Trainingsdatensatz die geringsten absoluten Fehler aufweist. Ferner zeigte die Untersuchung von PIOCH, HARMENING, MÜLLER et al. (2023), dass dieser Ansatz bei geringer Datenverfügbarkeit die geringsten Standardabweichungen der Fehler über die zehn Neu-Initialisierungen aufweist und damit als stabilster Ansatz eingeordnet werden kann. Dies bestätigte sich in einer Untersuchung des Ansatzes zur Approximation der Umströmung eines Tragflügelprofils bei variablen Anstellwinkeln (HARMENING, PIOCH, FUHRIG et al., 2023). Die Arbeit von HARMENING, PIOCH, FUHRIG et al. (2023) belegt eine Eignung des Turbulenzmodells für Reynolds-Zahlen bis $1 \cdot 10^6$.

5.4 Variable Eingangsgrößen

Um die Methode bei variablen Eingangsgrößen anwenden zu können, ist eine Erweiterung des betrachteten Raums auf ein n -dimensionales Gebilde erforderlich. Analog zu einem Quadrat oder Würfel kann die n -dimensionale Erweiterung als Hyperwürfel (engl. *Hypercube*) bezeichnet werden, welcher um beliebig viele Dimensionen erweitert werden kann. Im Falle der zweidimensionalen Darstellung eines umströmten Gebäudes mit variablem Dachneigungswinkel werden die räumlichen Dimensionen x und z um den Dachneigungswinkel β erweitert. Die Ergänzung dieser Dimension am Hyperwürfel ist in Abbildung 5.8 grafisch veranschaulicht.

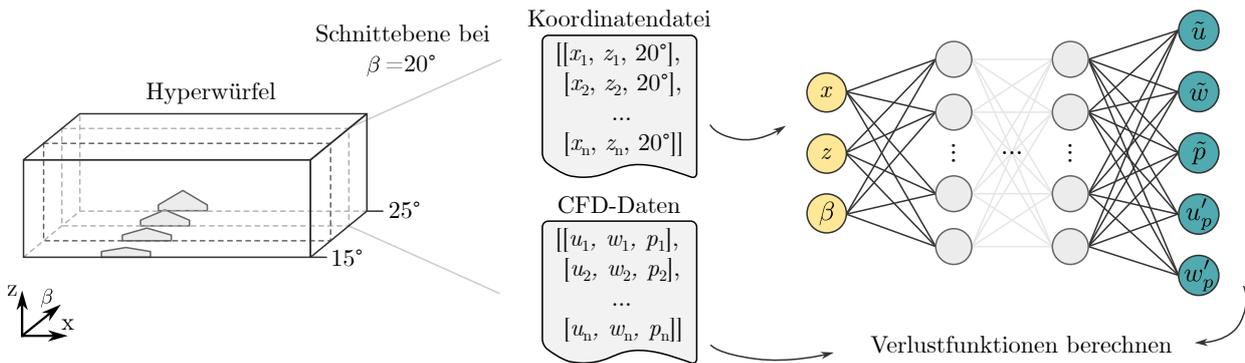


Abbildung 5.8: Grafische Darstellung eines Hyperwürfels bei der Umströmung eines Gebäudes mit variablem Dachneigungswinkel β (links), Koordinatendatei und Ergebnisdatei auf der Schnitt Ebene bei $\beta = 20^\circ$ (mittig) und das zugehörige neuronale Netz, bei Verwendung eines Turbulenzmodells mit pseudo-Reynolds-Spannungen (rechts) (eigene Darstellung)

Der Hyperwürfel ermöglicht es, das Strömungsphänomen unter Berücksichtigung der variablen Eingangsgrößen zu trainieren und auszuwerten. Die Strukturierung der Daten erfolgt in

einer $m \times n$ -dimensionalen Matrix, wobei n der Anzahl an Eingangsparametern am neuronalen Netz und m der Anzahl der Datensätze entspricht. Eine Erweiterung der Methode auf den dreidimensionalen Raum, den Einbezug der Zeitabhängigkeit sowie mehrere variable Eingangsgrößen sind ohne Weiteres möglich. Abbildung 5.8 stellt rechtsseitig die Netztopologie eines PINN dar, das bei der Anwendung dieser Methode auf die zweidimensionale stationäre Strömung mittels eines pseudo-Reynolds-Spannungs-Turbulenzmodells unter Einbezug des Dachneigungswinkels β verwendet wird.

5.5 Skalierung der Navier-Stokes-Gleichungen

Die Grundlagen neuronaler Netze dokumentierten, dass nach SUTTON (1986) eine unterschiedlich starke Krümmung der Flächen in den Dimensionen der Optimierungslandschaft zu einer instabilen Optimierung bei neuronalen Netzen führt. Dieser Umstand wird durch unterschiedlich hohe Absolutwerte der Verlustterme begünstigt, da diese zu einem Bias der Optimierung hinsichtlich des höchsten Absolutwertes führen. Deshalb werden die Daten für den Trainingsprozess klassischer neuronaler Netze normalisiert. Bei PINNs sind die Ausgabewerte zur Approximation der Navier-Stokes-Gleichungen in den Gleichungen implementiert und stehen in wechselseitigem Bezug zueinander, sodass eine unterschiedliche Skalierung der Geschwindigkeitskomponenten nicht möglich ist. Für den Druck gilt, dass lediglich der Gradient in die Gleichungen eingeht, weshalb hierbei eine unabhängige Skalierung der Daten erfolgen kann. Für die Vorgabe der Geometrie und Verwertung der vom PINN ausgegebenen Geschwindigkeitskomponenten ist die Skalierung komplexer. LAUBSCHER und ROUSSEAU (2021) wendeten zur Lösung dieser Problematik z. B. die nicht dimensionale Form der Navier-Stokes-Gleichungen an.

Im Falle der hier angenommenen inkompressiblen Medien ergibt sich aus dem von Reynolds aufgestellten Ähnlichkeitsgesetz, dass Strömungen bei übereinstimmender Reynolds-Zahl mechanisch ähnlich verlaufen. Die Reynolds-Zahl kann über die erzwungene Konvektion, Abmessungen der Geometrien und die Viskosität des Fluids eingestellt werden, woraus sich ein simpel zu implementierender Ansatz der Skalierung ergibt. Dieses Ähnlichkeitsgesetz ist ebenfalls die Grundlage der Argumentation für die Übertragung von maßstäblichen Ergebnissen aus Wassertank- oder Windkanalversuchen auf reale Objekte. Zur Anwendung für PINNs gilt es also, die Abmessungen der Geometrie, die Geschwindigkeitskomponenten und den Druck auf ein einheitliches Niveau zu bringen. Umsetzen lässt sich dies, indem die Strömungsdomäne bspw. auf Abmessungen in x - und z -Achsenrichtung $\in \{0; 1\}$ ausgelegt und ein künstliches Fluid mit passender Viskosität eingestellt wird, was zu einem Geschwindigkeitsfeld führt, dessen Absolutwerte den Abmessungen ähnlich sind. Der Druck wird unabhängig skaliert. Ergänzend ist bei den Eingangsgrößen am neuronalen Netz auf die Wahl der Einheit zu achten, so kann bspw. die Angabe eines Winkels in Radiant anstelle der Angabe in Grad erfolgen.

Im Rahmen einer Sensitivitätsanalyse wurde überprüft, inwiefern sich die Skalierung über die erzwungene Konvektion oder ein Fluid mit passgenauer Viskosität auf die Genauigkeit von PINNs auswirken. Die von HARMENING, PIOCH, FUHRIG et al. (2023) veröffentlichten Ergebnisse weisen deutliche Unterschiede zwischen den Methoden auf. In Abbildung 5.9 sind die Reynolds-Zahlen im Bereich von $Re \in \{1 \cdot 10^2; 1 \cdot 10^3; 1 \cdot 10^4; 1 \cdot 10^5; 1 \cdot 10^6\}$ zum einen über die Viskosität des Fluids eingestellt (links), und zum anderen über die variierte erzwungene Konvektion (rechts). Der logarithmische Boxplot ist nach Tukeys ursprünglicher Definition der Antennen dargestellt.

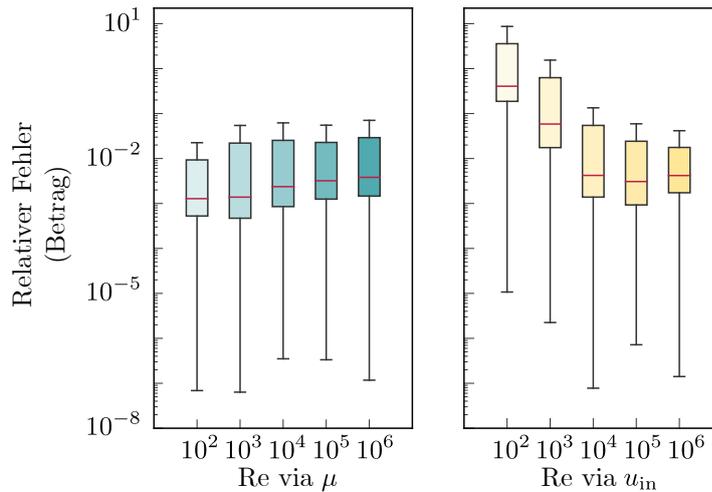


Abbildung 5.9: Boxplot der Betragswerte des relativen Fehlers von PINNs bei variiertem Reynolds-Zahl über die Viskosität (links) und erzwungener Konvektion (rechts), in Anlehnung an HARMENING, PIOCH, FUHRIG et al. (2023)

Die Ergebnisse weisen der Steuerung der Reynolds-Zahl über die Viskosität eine verhältnismäßig geringe Streuung der relativen Fehler nach. Der relative Fehler beträgt hierbei im Median über die Datensätze aller Testreihen 0,239 %, mit einem IQR von 2,073 %. Für die Steuerung über die erzwungene Konvektion dagegen beträgt der relative Fehler im Median 2,069 %, mit einem IQR von 25,124 %.

5.6 Sampling-Methode

Für das Sampling der Datensätze sind zwei Faktoren wesentlich: Zum einen ist fraglich, wie viele Datensätze für das Training benötigt werden, zum anderen stellt sich die Frage, ob die Verteilung der Datensätze im Raum einen Einfluss auf die Ergebnisse hat.

Zwar steigt mit zunehmender Anzahl der Trainingsdatensätze die Genauigkeit der Vorhersage eines PINN, allerdings geht dies mit einem erhöhten Berechnungsaufwand einher. Da das

Ziel der Methode ein Metamodell mit einer guten Genauigkeit der Vorhersage in der Interpolation und Extrapolation der Ebenen des Hyperwürfels sein soll, ist es erforderlich, die zu entnehmende Datenmenge der simulierten Schnittebenen des Hyperwürfels abzuschätzen, um so den Berechnungsaufwand minimal zu halten. HARMENING, PIOCH, FUHRIG et al. (2023) untersuchten dies für die Umströmung eines Tragflügelprofils bei variierten Anstellwinkeln und stellen fest, dass der Verlauf des relativen Fehlers ab 800 randomisiert entnommenen Trainingsdaten aus dem CFD-Datensatz stagniert (s. Abbildung 5.10). Die 800 Datensätze entsprechen einem Anteil von 1,29 % aller verfügbaren CFD-Datenpunkte des Rechengitters.

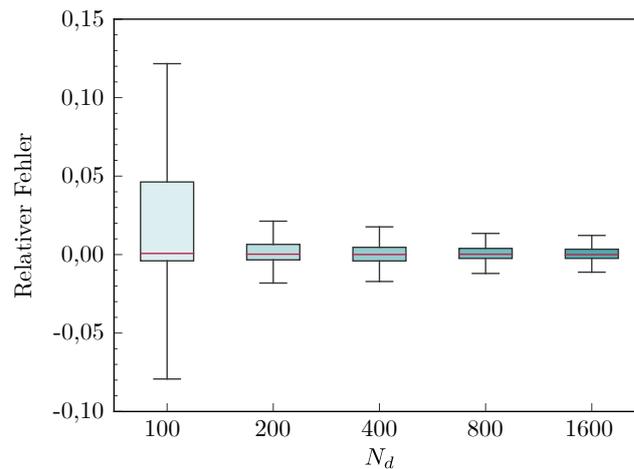


Abbildung 5.10: Boxplot der Auswertung des relativen Fehlers zwischen der CFD-Referenz und dem PINN für die Umströmung eines Tragflügelprofils bei variiertem Größe der Datensätze für das Training, in Anlehnung an HARMENING, PIOCH, FUHRIG et al. (2023)

Die ersten Arbeiten mit PINNs dokumentieren vorwiegend uniforme oder randomisiert verteilte Trainingsdaten auf äquidistanten Gittern. Aufgrund der in dieser Arbeit vorgeschlagenen Methode eines durch Daten assistierten PINN folgen weitere mögliche *Sampling*-Methoden: Zum einen dokumentieren EIVAZI und VINUESA (2022) oder HASANUZZAMAN et al. (2023) die Rekonstruktion eines hochaufgelösten Strömungsfeldes aus Messdaten, zum anderen weisen PIOCH, HARMENING, MÜLLER et al. (2023) oder R. WANG et al. (2020) die Eignung zur Rekonstruktion aus unvollständigen Simulationsdaten nach. Während Messungen häufig entlang von Linien entnommen werden, sind die CFD-Rechengitter durch verhältnismäßig kleine Elemente in der Nähe reibungsbehafteter Wände gekennzeichnet. Die Datensätze unterscheiden sich also je nach Art der Datengrundlage. Abbildung 5.11 stellt die je nach *Sampling*-Methode relativen Fehler der Vorhersage eines PINN für die Approximation eines Strömungsfeldes um ein Tragflügelprofil dar. Der logarithmische Boxplot zeigt die Box, den rot markierten Median und die Antennen. Die *Sampling*-Methoden sind rechts im Bild schematisch dargestellt und die Anzahl der Trainingsdatensätze N_d eingetragen. Von HARMENING, PIOCH, FUHRIG et al. (2023) wird für das *Sampling* mit CFD gestützten Datenpunkten die größte Genauigkeit nachgewiesen.

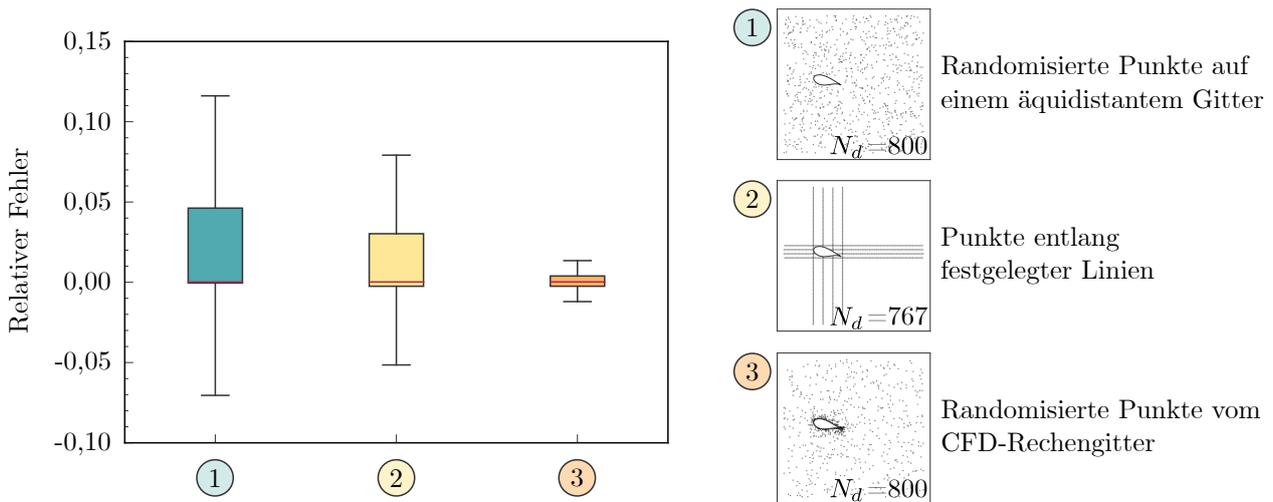


Abbildung 5.11: Boxplot der Auswertung des relativen Fehlers zwischen der CFD-Referenz und dem PINN für ein umströmtes Tragflügelprofil bei variiertes *Sampling*-Methode, in Anlehnung an HARMENING, PIOCH, FUHRIG et al. (2023)

5.7 Definition der Echtzeit für die Strömungsvorhersage in und um Stallungen

Die außen-klimatischen Windverhältnisse in der Atmosphäre bestimmen die Charakteristik der Durchströmung des frei belüfteten Stalls. Eine kurzzeitige Änderung der Windgeschwindigkeit wird als Böe bezeichnet und vom DWD als solche definiert, wenn die über zehn Minuten gemittelte Windgeschwindigkeit über eine Zeitspanne von $t \in \{3,0; \dots; 20,0\}$ s um mindestens $5,0 \frac{m}{s}$ überschritten wird. Die harte Echtzeit erfordert gemäß dieser Definition eine Lösung nach $\Delta t_{max} \leq 20$ s, um eine berechnete Lösung aus den Mittelwerten der vergangenen zehn Minuten von einer Böe unterscheiden zu können. Abbildung 5.12 visualisiert die Echtzeitanforderung für den PINN-Ansatz.

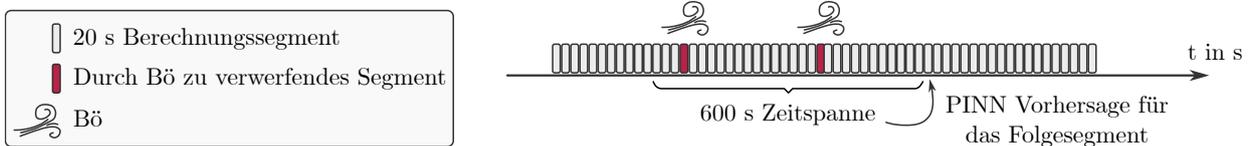


Abbildung 5.12: Echtzeitanforderung für die Vorhersage der vom Wind aufgeprägten Strömungen durch einen frei belüfteten Stall (eigene Darstellung)

Aufgrund des verwendeten Ansatzes der Approximation mittels neuronalem Netz ist ergänzend zu beachten, dass die Berechnung der Vorhersagen des Strömungsfeldes abhängig von der verwendeten Hardware ist. Es gilt daher nachzuweisen, dass die Ausgabe des Strömungsfeldes mittels leistungsarmer Hardware diese Anforderung weiterhin erfüllen kann.

5.8 Anwendung, Verifizierung und Validierung der Methode

Im Folgenden werden die ermittelten Einstellungen verwendet, um eine echtzeitfähige Strömungsvorhersage im und um einen Stall zu entwickeln.

Turbulenzmodell

Aufgrund der im Rahmen dieser Arbeit verwendeten inkompressiblen und mittels RANS simulierten Strömungen bei hohen Reynolds-Zahlen kann lediglich das in den Arbeiten von PIOCH, HARMENING, MÜLLER et al. (2023), HARMENING, PIOCH, FUHRIG et al. (2023) und EIVAZI, TAHANI et al. (2022) verwendete pseudo-Reynolds-Spannungs-Turbulenzmodell verwendet werden. Zum Zeitpunkt der Erstellung dieser Arbeit weist keine andere Arbeit die Eignung einer Methode für Reynolds-Zahlen bis $1 \cdot 10^6$ nach.

Hyperwürfel

Kapitel 4.6.3 dokumentierte die Auswahl der zu variierenden Faktoren für ein voll quadratisches Antwortflächenmodell. Teile dieser Ergebnisse der statistischen Untersuchung werden folgend für das Training des PINN verwendet, und die identischen Faktoren werden variabel gestaltet. Der Hyperwürfel wird daher, ergänzend zu den Raumrichtungen, auch die Dimension des Zustandes der Wickellüftung, Windgeschwindigkeit sowie Dachneigung beinhalten. Die Trainingskoordinaten für das PINN bestehen also aus einer $m \times n$ Matrix mit den Zeilen $\sum_{m=0}^N = [x_m, z_m, W_m, \beta_m, v(10)_m]$. Auf den Ebenen des Hyperwürfels unterscheiden sich die für das Training verwendeten kartesischen Koordinaten, nicht aber die ergänzenden Modellparameter. Abbildung 5.13 stellt dies am Beispiel des Versuchsdurchlaufes N8 des Box-Behnken-Versuchsplanes mit einem Dachneigungswinkel von $\beta = 20^\circ$ und der Wickellüftung $W = 90\%$ dar. Die Markierungen visualisieren die exportierten Datensätze aus der CFD-Simulation auf der Ebene des Hyperwürfels.

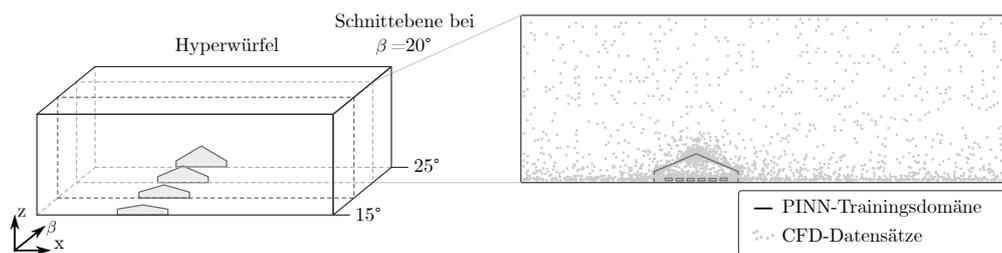


Abbildung 5.13: Hyperwürfel und Trainingskoordinaten des PINN auf einer Schnittebene bei $\beta = 20^\circ$ (eigene Darstellung)

Skalierung

Um einen Bias der Optimierung und die stark gekrümmten Raumrichtungen der Optimierungslandschaft zu vermeiden, werden die Parameter gemäß der dargelegten Methode der Ähnlichkeitsgesetze nach Reynolds skaliert. Die Abmessungen der räumlichen Dimensionen der Strömungsdomäne werden durch den Skalierungsfaktor 100 dividiert. Um die

Eingangsgrößen am neuronalen Netz vollständig zu skalieren, sind die ergänzenden Variablen ebenfalls normiert, indem $W \in \{0,3; 0,6; 0,9\}$, $\beta \in \{0,26; 0,35; 0,44\}$ rad und $v_m(10) \in \{0,14; 0,57; 1,00\} \frac{m}{s}$ gesetzt werden. Die mechanische Ähnlichkeit der Strömung wird letztlich über ein künstliches Fluid der Dichte $\rho = 1000 \frac{kg}{m^3}$ und der dynamischen Viskosität von $\mu = 7,29 \cdot 10^{-6} \frac{kg}{m \cdot s}$ eingestellt. Da beim Druck lediglich der Gradient in die Navier-Stokes-Gleichungen eingeht, kann dieser mit dem Faktor 100 unabhängig skaliert werden. Eine Reskalierung zur Visualisierung und Rückgabe von Werten in den Dimensionen des mit Luft umströmten Stalls in realer Größe ist gemäß der definierten Vorgehensweise und Faktoren möglich.

Sampling

Gemäß der Voruntersuchung bietet eine randomisierte Verteilung der Datenpunkte auf einem CFD-Rechengitter die höchste Genauigkeit der untersuchten Methoden. Daher sind die Datenpunkte aller CFD-Simulationen des Box-Behnken-Versuchsplans auf einer z - x -Ebene bei $y = 20,81$ m exportiert. Die x - z -Ebene bietet für die zweidimensionale Approximation des PINN eine etwa mittig des Geheges platzierte Ebene, unter Berücksichtigung der Geometrie stehender und liegender Kühler. Die Limitierung des Grafikspeichers begrenzt die Auswahl der Daten auf $N_d = 400$ Punkte je Ebene des Hyperwürfels, was insgesamt 5.200 Punkten oder 3,0 % des gesamten Datensatzes aus den CFD-Simulationen entspricht. Abbildung 5.14 stellt das Konturdiagramm der CFD-Simulation des Box-Behnken-Versuches $N8$, die Verteilung der verfügbaren Datenpunkte auf der Schnittebene und eine randomisierte Auswahl der 400 Datenpunkte für das Training dar.

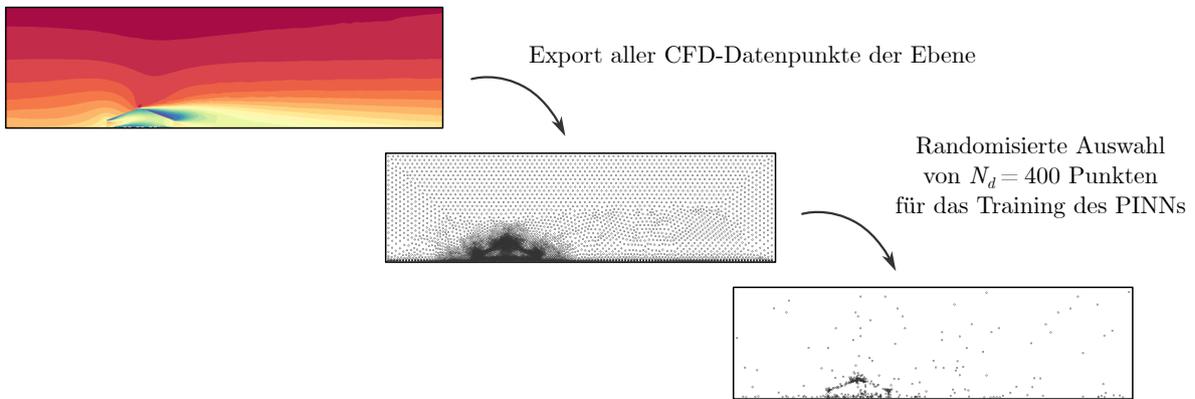


Abbildung 5.14: *Sampling* der Datenpunkte für das Training des PINN am Beispiel des Box-Behnken-Versuchsdurchlaufes $N8$ (eigene Darstellung)

Hyperparameter

Das Training des PINN wird mit den in Tabelle 5.1 aufgeführten Parametern durchgeführt. Die Untersuchung der Netztopologie bei der Anwendung des pseudo-Reynolds-Spannungsturbulenzmodells wurde bereits von HARMENING, PIOCH und SCHRAMM (2022) analysiert. Die mehrfache Neuinitialisierung ist gemäß LU et al. (2021) zehnmals vorgesehen. Eine Voruntersuchung der Aktivierungsfunktionen wies für den *Tangens hyperbolicus* die besten Er-

gebnisse nach. Zur Vermeidung der Tendenz zu lokalen Optima ist die Optimierung zunächst nach Adam mit hoher Lernrate, darauf folgend nach Adam mit geringer Lernrate und abschließend mit dem L-BFGS-Verfahren vorgesehen. Die Anzahl der verwendeten Epochen beruht auf den Erkenntnissen der untersuchten Beispiele in PIOCH, HARMENING, MÜLLER et al. (2023) sowie HARMENING, PIOCH, FUHRIG et al. (2023) und HARMENING, PIOCH und SCHRAMM (2022). Der Konvergenzverlauf wird zur Beurteilung für diesen Anwendungsfall über die zehn Durchläufe gemittelt und die Unsicherheit in Form einer Standardabweichung dokumentiert.

Tabelle 5.1: Hyperparameter des PINN

Parameter	Wert
Netztopologie	128 Neuronen auf fünf verdeckten Schichten
Initialisierung	<i>Glorot uniform</i>
Aktivierungsfunktion	<i>Tangens hyperbolicus</i>
Optimierer	1. Adam, 2. Adam und 3. L-BFGS
Lernrate	1. $1 \cdot 10^{-3}$ und 2. $1 \cdot 10^{-4}$
Epochen und Konvergenzkriterium	1. 10.000, 2. 10.000 und 3. $\epsilon = 1 \cdot 10^{-5}$

Programmablauf

Der Trainingsablauf beinhaltet die in Abbildung 5.15 dargestellten Prozesse, Subprozesse und zuvor dargelegten Verlustfunktionen und Randbedingungen. Zudem sind die Ex- und Importe erforderlicher Daten visualisiert. Im Anhang E.2 ist das verwendete PYTHON-Programm dokumentiert.

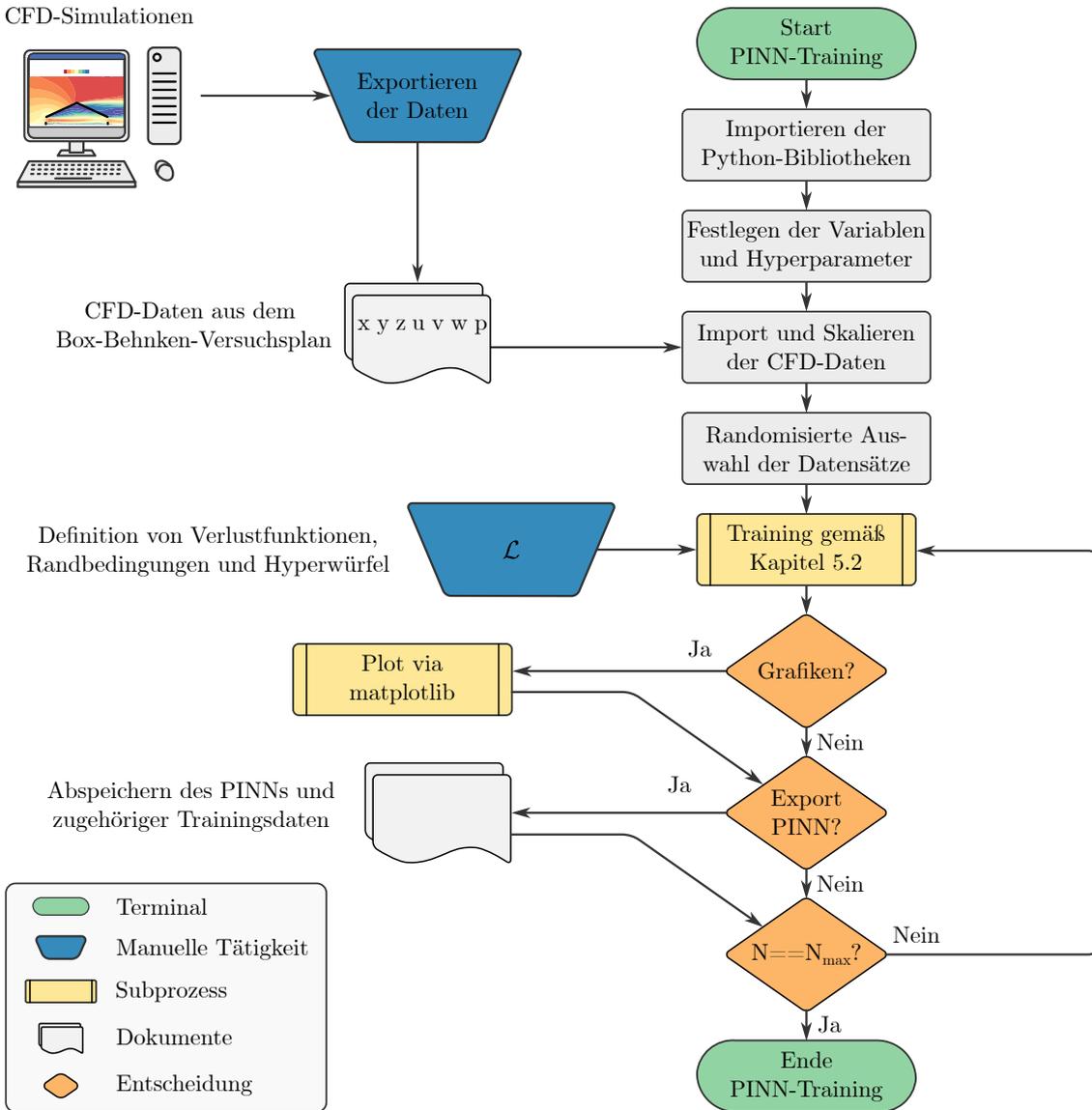


Abbildung 5.15: Ablaufdiagramm für das PINN, vom Datenexport über die Angabe der Gleichungen, bis hin zur Visualisierung und Speicherung der Daten des Trainingsprozesses (eigene Darstellung)

Numerischer Aufwand und Echtzeit

Das Training der Durchläufe des PINN-Ansatzes betrug über die zehn Neuinitialisierungen im Mittel 5.784 s, mit einer Standardabweichung von ± 156 s, wohingegen die Berechnungsdauer einer 3D-CFD-Simulation für eine Konstellation aus dem Box-Behnken-Versuchsplan im Mittel 43.644 s, mit einer Standardabweichung von ± 16.653 s betrug. Sowohl das Training des PINN, als auch die CFD-Simulation wurden auf einem Computer mit einem INTEL CORE i7-9700K, einer NVIDIA GEFORCE RTX 2700 und 64 Gigabyte DDR4 Arbeitsspeicher durchgeführt. Das 64-Bit-Betriebssystem des Computers war eine WINDOWS 10 EDUCATION Version mit dem Betriebssystembuild 19045.2965.

Die Echtzeitfähigkeit wurde für die Vorhersage unterschiedlich großer Datensätze und zweier

Hardwarekonfigurationen überprüft. Die am Eingang des PINN aufgegebenen Koordinaten führen zur Vorhersage der gleichen Anzahl diskreter Werte für die Geschwindigkeitskomponenten, den Druck und die pseudo-turbulenten Größen des PINN spezifischen Turbulenzmodells. Tabelle 5.2 listet die Ergebnisse einer Untersuchung zu den Laufzeiten bei einer Ausführung zur Vorhersage von $1 \cdot 10^2$, $1 \cdot 10^4$ und $1 \cdot 10^6$ Werten mittels Desktop-Prozessor und einer Workstation-Grafikkarte.

Tabelle 5.2: Rechenzeit und Standardabweichung des PINN für unterschiedlich große Datensätze und zwei Hardwarekonfigurationen

Hardware	Datensätze	\bar{t} in s	σ in s
Desktop-CPU INTEL CORE I5-12400F	$1 \cdot 10^2$	$0,4 \cdot 10^{-3}$	$\pm 0,1 \cdot 10^{-6}$
	$1 \cdot 10^4$	$6,5 \cdot 10^{-3}$	$\pm 6,5 \cdot 10^{-6}$
	$1 \cdot 10^6$	$556,0 \cdot 10^{-3}$	$\pm 9,1 \cdot 10^{-3}$
Workstation-GPU NVIDIA QUADRO RTX 5000	$1 \cdot 10^2$	$1,3 \cdot 10^{-3}$	$\pm 1,4 \cdot 10^{-6}$
	$1 \cdot 10^4$	$1,4 \cdot 10^{-3}$	$\pm 0,1 \cdot 10^{-6}$
	$1 \cdot 10^6$	$62,9 \cdot 10^{-3}$	$\pm 1,1 \cdot 10^{-3}$

Sowohl bei der Berechnung mittels Desktop-Prozessor (INTEL CORE I5-12400F), als auch bei der Workstation-Grafikkarte (NVIDIA RTX 5000) ist ein deutlicher Unterschied zwischen einem Datensatz mit $1 \cdot 10^4$ und $1 \cdot 10^6$ Werten erkennbar, woraus eine um den Faktor 85,5 oder 45,0 längere Laufzeit resultiert. Beim größten Datensatz wird außerdem der Unterschied zwischen der Hardware sichtbar, welcher in einer etwa neunmal höheren Laufzeit für den Desktop-Prozessor mündet. Dennoch erfüllen auch die längeren Laufzeiten für $1 \cdot 10^6$ Werte die formulierte Echtzeitanforderung bei Weitem.

Verifizierung und Validierung

Zur Optimierung der Gewichte und Biases werden die in Kapitel 5.1 dokumentierten Verlustmetriken während des Trainings des neuronalen Netzes berechnet und alle 1.000 Iterationen gespeichert. In Abbildung 5.16 ist der Verlauf des über alle Verlustmetriken und Neuinitialisierungen gemittelten MSE der Iterationen des Trainings dargestellt. Die Standardabweichung der Neuinitialisierungen ist für den gemittelten MSE ebenfalls alle 1.000 Iterationen berechnet und visualisiert. Die logarithmische Darstellung unterstreicht die anfänglich deutlich erzielte Optimierung, welche im Laufe der weiteren Iterationen abnimmt. Durch die Umstellung auf das L-BFGS-Verfahren zeigt sich eine weitere sichtbare Reduktion der Verlustmetriken, bevor nach etwa 45.000 Iterationen das interne Konvergenzkriterium des L-BFGS-Verfahrens erreicht wird und die Optimierung abbricht. Die mittlere Verlustmetrik am Ende der Optimierung beträgt $\overline{\text{MSE}} = 1,46 \cdot 10^{-2}$, bei einer Standardabweichung von $\sigma \overline{\text{MSE}} = 1,28 \cdot 10^{-3}$. Wobei die mittlere Verlustmetrik der physikalischen Terme $\overline{\text{MSE}} = 3,57 \cdot 10^{-3}$, der strömungstechnischen Randbedingungen $\overline{\text{MSE}} = 1,77 \cdot 10^{-2}$ und hinsichtlich der Daten mit $\lambda = 100$ gewichtete $\overline{\text{MSE}} = 2,24 \cdot 10^{-2}$ beträgt.

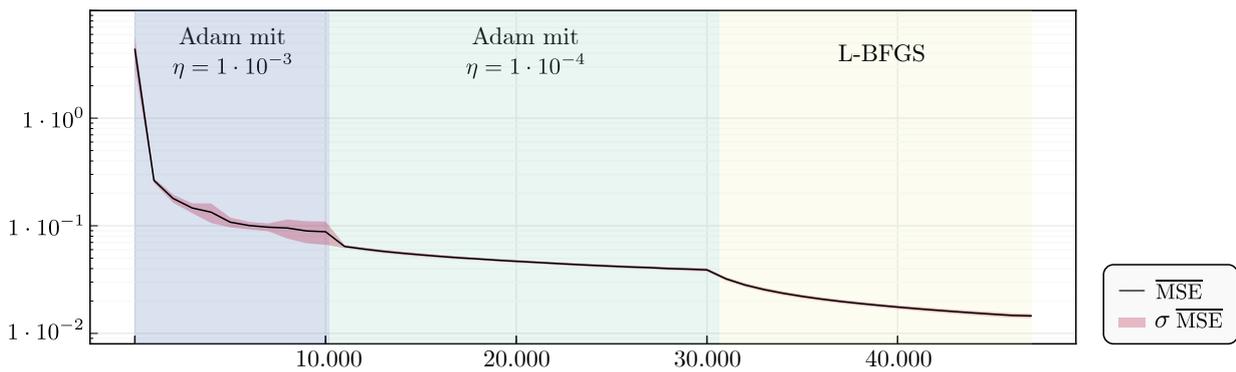


Abbildung 5.16: Gemittelte Verlustmetrik und Standardabweichung des Trainingsprozesses für ein PINN zur Vorhersage der Strömungen am Stall (eigene Darstellung)

Die globale Verlustmetrik kann über den Abgleich von Datensätzen der CFD-Simulationen mit der Vorhersage des PINN um eine lokale Einordnung ergänzt werden. Vergleicht man die Vorhersage des PINN auf einer Ebene des Hyperwürfels, welche Teil des Trainings war, ergeben sich die in Abbildung 5.17 dargestellten Konturdiagramme mit den angegebenen lokalen absoluten Fehlern.

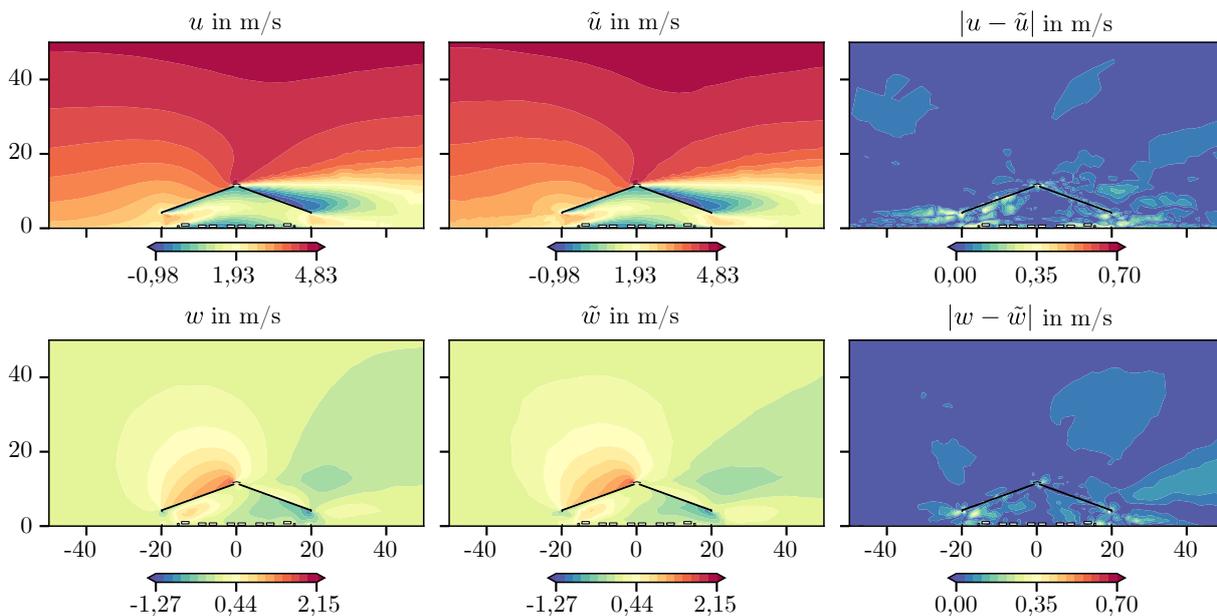


Abbildung 5.17: Konturdiagramme der CFD-Simulation, (reskalierten) PINN-Vorhersage und dem sich ergebenden absoluten Fehler für $W = 60\%$, $\beta = 25^\circ$ und $v_m(10) = 3,5 \frac{m}{s}$ (eigene Darstellung)

Im Abgleich mit 1.000 CFD-Datensätzen, welche nicht im Trainingsdatensatz enthalten waren, ergeben sich bei der Kombination nach Versuchsdurchlauf zwölf ($W = 60\%$, $\beta = 25^\circ$ und $v_m(10) = 3,5 \frac{m}{s}$) Abweichungen in den Bereichen hoher Gradienten, wie an den Stauflächen der versperreten Wickellüftung. Die Darstellung des absoluten Fehlers, mit einer Limitierung

auf 20 % der Referenzgeschwindigkeit $v_m(10)$, belegt eine im Gros gute Übereinstimmung für beide Geschwindigkeitskomponenten, wobei die Geschwindigkeitskomponente w gegenüber der Hauptströmungsrichtung u größere relative Abweichungen aufweist. Die Auswertung der Validierungsmetriken zeigt eine Validität des PINN-Ansatzes im Vergleich zu den CFD-Daten einer im Training enthaltenen Ebene des Hyperwürfels, wie die in Tabelle 5.3 aufgeführten Werte belegen.

Tabelle 5.3: Validierungsmetriken für den Abgleich des PINN mit den Daten der CFD-Simulation für den Versuchsdurchlauf zwölf des Box-Behnken-Versuchsplans

Quantität	VM	MG	FB	NMSE	VG	FAC2
u	0,85	0,97	0,01	0,01	1,48	0,91
w	0,56	0,71	0,03	0,21	9,26	0,58

Bei der Auswertung zur Interpolation im Hyperwürfel sind die Parameter aller drei zusätzlichen Dimensionen interpoliert. Die Wickellüftung ist zu $W = 75\%$ geöffnet, der Dachneigungswinkel beträgt $\beta = 22,5^\circ$ und die Basisgeschwindigkeit $v_m(10) = 2,75 \frac{m}{s}$. Abbildung 5.18 stellt die Ergebnisse der CFD-Simulation der Vorhersage des neuronalen Netzes gegenüber und visualisiert den absoluten Fehler im Feld.

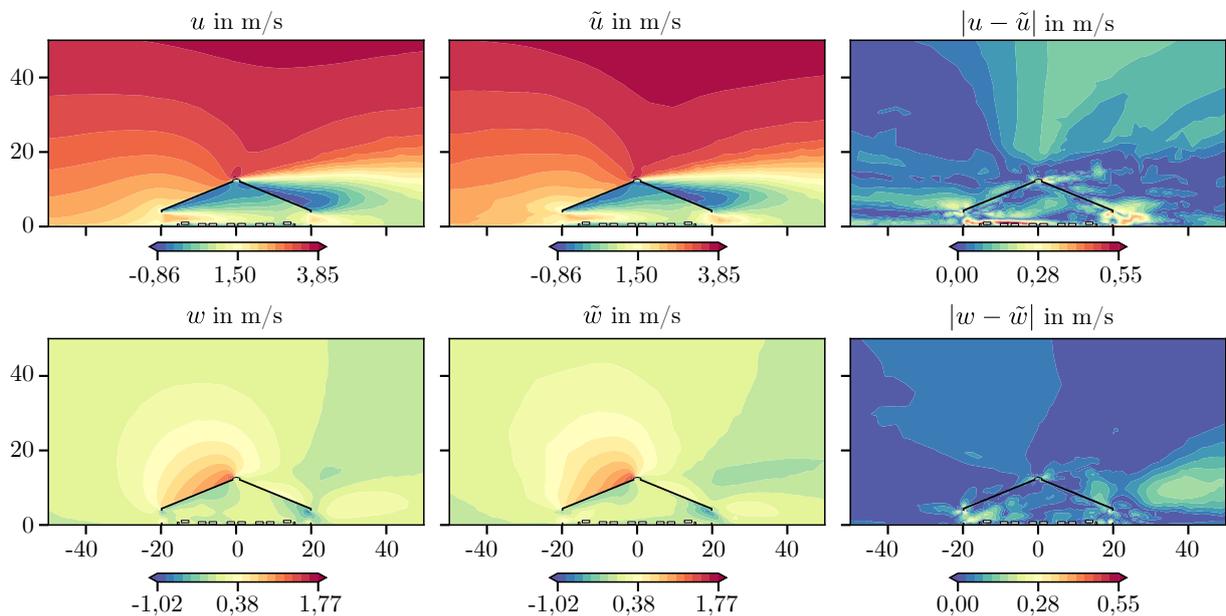


Abbildung 5.18: Konturdiagramme der CFD-Simulation, (reskalierten) PINN-Vorhersage und dem sich ergebenden absoluten Fehler für $W = 75\%$, $\beta = 22,5^\circ$ und $v_m(10) = 2,75 \frac{m}{s}$ (eigene Darstellung)

Bezogen auf die Basisgeschwindigkeit $v_m(10)$ am Einströmrand zeigen sich auf der Interpolationsebene des Hyperwürfels Abweichungen von etwa 5,0 % im Umfeld des Stalls und bis

zu 15,0 % in den Bereichen hoher Geschwindigkeitsgradienten. Die Auswertung der Validierungsmetriken für einen Datensatz von 1.000 randomisiert ausgewählten Punkten belegt die Validität des Modells für eine Interpolation im Hyperwürfel. Der FAC2 ist jedoch für die Geschwindigkeitskomponente w nur grenzwertig erfüllt. Dies kann anhand der Absolutwerte der betrachteten Geschwindigkeitskomponente w erklärt werden, die häufig bei ca. $0,0 \frac{\text{m}}{\text{s}}$ liegt und für die der FAC2 bereits bei geringfügigen Abweichungen nicht erfüllt ist.

Tabelle 5.4: Validierungsmetriken für den Abgleich des PINN mit den Daten der CFD-Simulation bei einer Interpolation im Hyperwürfel

Quantität	VM	MG	FB	NMSE	VG	FAC2
u	0,79	0,97	-0,01	0,02	1,81	0,88
w	0,51	0,57	0,00	0,27	17,75	0,51

Die Extrapolation zeigt für die Anwendung auf die vorgestellte Problemstellung und unter Verwendung der zusätzlichen Dimensionen des Hyperwürfels eine unzureichende Approximation des Geschwindigkeitsfeldes (s. Abbildung 5.19). Die Kombination der Extrapolation ist, analog zur Interpolation, in allen ergänzten Dimensionen, mit $W = 15 \%$, $\beta = 27,5^\circ$ und $v_m(10) = 4,25 \frac{\text{m}}{\text{s}}$, angelegt.

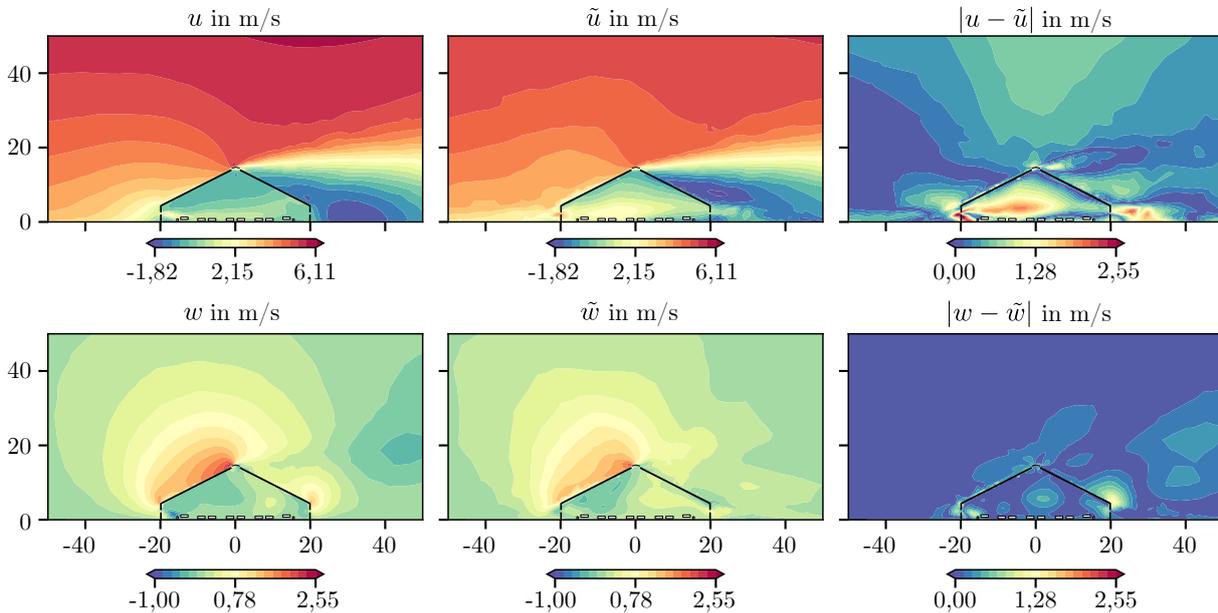


Abbildung 5.19: Konturdiagramme der CFD-Simulation, (reskalierten) PINN-Vorhersage und dem sich ergebenden absoluten Fehler für $W = 15 \%$, $\beta = 27,5^\circ$ und $v_m(10) = 4,25 \frac{\text{m}}{\text{s}}$ (eigene Darstellung)

Bei der Extrapolation ist sichtbar, dass der qualitative Verlauf der Hauptgeschwindigkeitskomponente u im Stall durch das PINN nicht vorhergesagt wird und der Fehler bis zu 60 %

der Referenzgeschwindigkeit $v_m(10)$ beträgt. Für die Geschwindigkeitskomponente w zeigen sich ebenfalls große Abweichungen von bis zu 30 %. Die qualitative Auswertung des Geschwindigkeitsfeldes der Extrapolation bestätigt sich in den Validierungsmetriken, die das Modell als nicht valide bewerten. In Tabelle 5.5 ist zu erkennen, dass sowohl die allgemeine Übereinstimmung nach VM, als auch die großen Abweichungen nach FAC2 gegenüber der Referenz und Interpolation deutlich abfallen.

Tabelle 5.5: Validierungsmetriken für den Abgleich des PINN mit den Daten der CFD-Simulation für eine Extrapolation aus dem Hyperwürfel

Quantität	VM	MG	FB	NMSE	VG	FAC2
u	0,60	0,63	0,02	0,18	7,96	0,64
w	0,41	0,42	-0,05	0,88	141,92	0,40

Zwischenfazit

Die Berechnung der Geschwindigkeitskomponenten u und w auf der x - z -Ebene bei $y = 20,8$ m zeigte eine nach Metriken valide Approximation unter Verwendung der vorgestellten Methode für die Referenzebene und Interpolation im Hyperwürfel. Die Extrapolation zeigte dagegen keine validen Ergebnisse. Das Modell kann daher für die Referenzebenen und die Interpolation eingesetzt werden und eine vollständige Visualisierung der Geschwindigkeitskomponenten liefern. Dies ermöglicht eine echtzeitfähige Analyse des qualitativen Strömungsverlaufes und den in und um den Stall vorherrschenden Geschwindigkeiten. Können diese Daten für landwirtschaftlich Beschäftigte oder das Regelungssystem der Lüftungselemente verfügbar gemacht werden, wäre es möglich, die Aktoren im Stall auf die aktuellen oder künftigen Bedingungen einzustellen und so optimale Strömungsbedingungen im Stall herzustellen und zu gewährleisten.

Entwicklung des digitalen Zwillings für Stallungen

Der Einsatz digitaler Zwillinge in der Landwirtschaft kann dazu beitragen, die Produktivität zu steigern, den Ressourcenverbrauch zu verringern und die Nachhaltigkeit zu erhöhen. Um digitale Zwillinge erfolgreich zu implementieren und zu betreiben, ist es jedoch entscheidend, das Kommunikationsschema, die Kommunikationsprotokolle, die Aspekte der IT-Sicherheit und den Zugriff hinsichtlich der Anforderungen der Landwirtschaft zu entwickeln. Diese Entwicklungen gewährleisten den nahtlosen Datenaustausch, die Benutzungsfreundlichkeit und den größtmöglichen Schutz vor Cyber-Bedrohungen, was letztlich zur erfolgreichen Implementierung von digitalen Zwillingen in der Landwirtschaft und zur Realisierung ihrer Vorteile führt.

Dieser Abschnitt der Arbeit baut auf dem Beitrag von PIOCH, HARMENING und SCHRAMM (2021) auf. Der Beitrag wird in diesem Kapitel ergänzt, weiterentwickelt und überarbeitet.

6.1 Kommunikationsteilnehmer und -protokolle

Die Kommunikationsteilnehmer bestimmen maßgeblich den Aufbau der Kommunikationsarchitektur und spielen daher eine entscheidende Rolle bei der Implementierung digitaler Zwillinge. In diesem Kapitel werden die Kommunikationsteilnehmer ermittelt und darauf basierend geeignete Kommunikationsprotokolle ausgewählt.

6.1.1 Kommunikationsteilnehmer

Für den landwirtschaftlichen digitalen Zwilling eines frei belüfteten Kuhstalls gibt es vielfältige Kommunikationsteilnehmer unterschiedlicher Komplexität, die an dem System beteiligt sind und unterschiedlichen Kommunikationsebenen zugeordnet werden können. Im Folgenden werden die Komponenten ermittelt, erläutert und anschließend in einer hierarchischen Struktur geordnet.

Sensoren sind für die Erfassung von Daten zu verschiedenen umwelt- und tierbezogenen Parametern im Stall zuständig, z. B. Lufttemperatur, Luftfeuchtigkeit, Emissionen oder Windgeschwindigkeit. Sensoren unterschiedlicher Hersteller können über drahtgebundene oder drahtlose Verbindungen verfügen und sind oftmals mit differenten Kommunikationsmöglichkeiten ausgestattet.

Aktoren steuern verschiedene Stallelemente wie Ventilatoren, Wickellüftungen oder die Beleuchtung und sollen auf die von den Sensoren erfassten Daten reagieren. Eine vorausschauende Regelung dieser Aktoren kann vom digitalen Zwilling auf der Grundlage vordefinierter Regeln und Algorithmen oder als Reaktion auf bestimmte Ereignisse oder Bedingungen gesteuert werden.

Das **Regelungssystem** ist ggf. als separate Einheit für die Verarbeitung der von den Sensoren gesammelten Daten oder Prädiktionen des digitalen Zwillings zuständig, um die Regelung der Aktoren im Stall zu übernehmen. Erst durch die Nutzung der Daten für eine vorausschauende Regelung der Aktuatoren können die Vorteile des digitalen Zwillings nutzbar gemacht werden.

Abbildung 6.1 ordnet die Kommunikationsteilnehmer ihrer Hierarchieebene, in Anlehnung an DIN EN 62264-1:2014-07 (2014), zu.

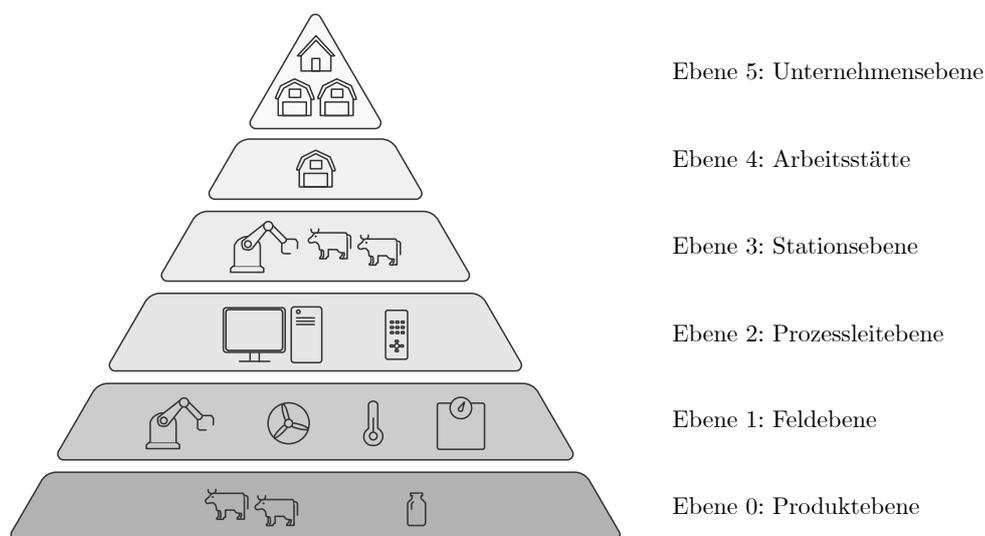


Abbildung 6.1: Kommunikationshierarchie in Anlehnung an DIN EN 62264-1:2014-07 (2014) (eigene Darstellung)

Auf unterster Ebene befindet sich das Produkt, welches im Falle des frei belüfteten Kuhstalls die Milch oder bei der Zucht die Kuh selbst ist. Auf der Feldebene sind Sensoren und Aktoren verortet, wozu u. a. die Lüftungselemente, Temperatursensoren und Sensorik des Melkroboters zählen. Die Steuerung oder Regelung dieser Elemente findet auf der darüber liegenden Prozessleitebene statt. Im Falle der Melkroboter lassen sich die darunter liegenden Elemente zu einer Station zusammenfassen. Der gesamte frei belüftete Kuhstall kann als übergeordnetes Element angesehen werden, welches die darunter liegenden Elemente in einer Arbeitsstätte zusammenfasst. Auf der Unternehmensebene ist davon auszugehen, dass landwirtschaftlich Beschäftigte mehrere Arbeitsstätten verwalten, die sich mit der Zeit verändern können. Das Konzept des digitalen Zwillinges muss diese Flexibilität implementieren, indem sowohl eine durchgängige Kommunikation zwischen den unterschiedlichen Ebenen als auch die erforderlichen Funktionen für das Erstellen, Manipulieren und Löschen von Elementen oder Daten bereitgestellt werden.

Die **Internet der Dinge (IoT-)Bridge** ist dafür verantwortlich, Komponenten ohne native Unterstützung der erforderlichen Kommunikationsprotokolle mit dem digitalen Zwilling zu verbinden. Erst durch die *IoT-Bridge* können Sensoren, Aktoren oder Kontrolleinheiten aller Art in das Kommunikationsschema eingebettet und die Daten zuverlässig übertragen werden. Die *IoT-Bridge* kann also durch die Ergänzung der vorhandenen Software oder eine Weiterverarbeitung und Übersetzung nativer Formate durch einen Mikrocontroller umgesetzt werden.

Das **Datenbanksystem** ist für die Speicherung und Verarbeitung der vom digitalen Zwilling gesammelten Daten zuständig. Die größtmögliche Flexibilität wird erst durch ein Konzept ermöglicht, das sowohl online- als auch offline-fähig ist. Aus einer solchen Vorgabe resultiert, dass keine Anpassung des Programms erforderlich ist, und die Entscheidungsgewalt des Konzepts weiterhin bei den anwendenden Personen liegt. Das Datenbanksystem muss in der Lage sein, die heterogenen großen Datenmengen der verschiedenen Sensoren und Aktoren zu verarbeiten.

Eine **Programmierschnittstelle** ermöglicht einen Zugriff für die Feld- und Prozessleitebene, sodass die Daten gesendet, überwacht und ggf. die relevanten Aktoren des Stalls geregelt werden können. Ergänzend ist es für die Mensch-Maschine-Interaktion erforderlich, die Daten über Desktop- oder Mobilgeräte verfügbar zu machen, um den landwirtschaftlich Beschäftigten zu ermöglichen, den Status des Stalls jederzeit im Blick zu behalten und ggf. auch manuell eingreifen zu können.

6.1.2 Analyse zum Kommunikationsprotokoll

Die Erprobung verschiedener Kommunikationsprotokolle für die Anwendung auf digitale Zwillinge in der Landwirtschaft ist essentiell, um die Leistungsfähigkeit, Interoperabilität

und Sicherheit zu bewerten. Nach HEIDEL et al. (2017) empfiehlt sich für industrielle Applikationen die OPC UA Schnittstelle auf der Feldebene. Ergänzend ist der Einsatz des *Representational State Transfer* (REST) für netzwerkübergreifende Kommunikation in Betracht zu ziehen.

OPC UA ist als offener und interoperabler Standard für die industrielle Kommunikation konzipiert. Das bedeutet, dass dieser Standard mit einer Vielzahl von Systemen und Geräten verschiedener Hersteller kompatibel ist, was die Integration in bestehende Systeme und Technologien für den landwirtschaftlichen digitalen Zwilling erleichtert. OPC UA bietet eine Reihe von Sicherheitsfunktionen, einschließlich Verschlüsselung, Authentifizierung und Zugriffskontrolle. Diese Sicherheitsmechanismen können dazu beitragen, dass das System des digitalen Zwillings sicher und vor unbefugtem Zugriff oder Datenverletzungen geschützt bleibt. OPC UA beinhaltet einen standardisierten Datenmodellierungsansatz, der eine konsistente Darstellung von Daten über verschiedene Systeme und Geräte hinweg ermöglicht. Dies kann besonders nützlich für digitale Zwillinge in der Landwirtschaft sein, bei denen oft Daten aus verschiedenen Quellen integriert werden müssen. Das Protokoll ist so konzipiert, dass es in der Lage ist, große Datenmengen mit geringen Latenzzeiten zu verarbeiten. Dies ist besonders wichtig für Prozesse, bei denen es um die Echtzeitüberwachung und -steuerung mit geringen Zykluszeiten geht. Abbildung 6.2 stellt die Kommunikation zwischen Client und Server via OPC UA dar.

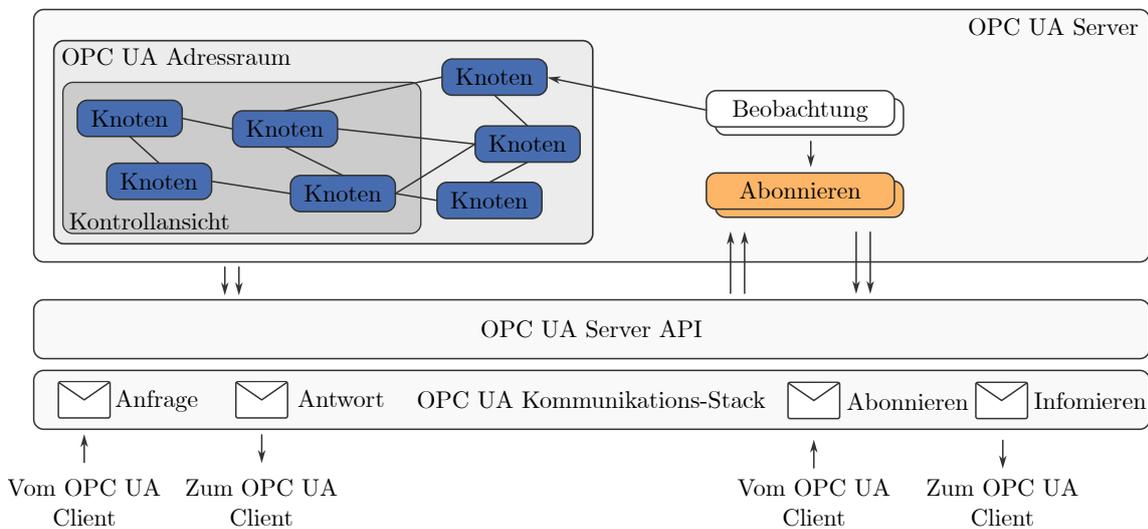


Abbildung 6.2: OPC UA Kommunikationsstruktur, in Anlehnung an CAVALIERI und CHIACCHIO (2013) (eigene Darstellung)

REST-basierte Protokolle hingegen können leicht skaliert werden, um eine große Anzahl von Anfragen und anwendenden Personen zu unterstützen. Sie sind außerdem so konzipiert, dass sie zustandslos sind. Das heißt, jede Anfrage enthält alle Informationen, die für die Ausführung der Anfrage erforderlich sind, was die Verteilung von Anfragen auf mehrere Server erleichtert. REST-basierte Protokolle sind weit verbreitet und werden von vielen Kom-

ponenten unterstützt, was die Integration mit anderen Systemen und Diensten erleichtert. REST-basierte Protokolle können mit einer Vielzahl von Datenformaten verwendet werden, darunter *JavaScript Object Notation* (JSON) oder *Extensible Markup Language* (XML). Diese Flexibilität kann besonders für digitale Zwillinge in der Landwirtschaft nützlich sein, die oft mit unterschiedlichen Datentypen der verschiedenen Sensoren oder externen Quellen arbeiten. REST-basierte Protokolle können mit Standard-Web-Sicherheitsmechanismen wie *Secure Sockets Layer* (SSL), *Transport Layer Security* (TLS) Verschlüsselung, Authentifizierung und Zugriffskontrolle gesichert werden. Dies kann dazu beitragen, dass das System des digitalen Zwillings sicher und vor unbefugtem Zugriff oder Datenverletzungen geschützt bleibt. Abbildung 6.3 visualisiert eine REST-basierte Kommunikation zwischen Client und Server.

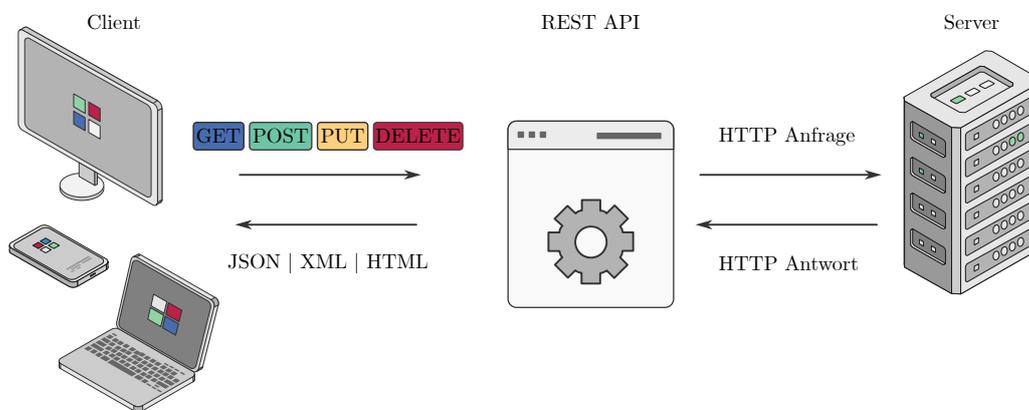


Abbildung 6.3: REST Kommunikationsstruktur (eigene Darstellung)

Der Aufbau und die Erläuterungen zu den vorgestellten Kommunikationsprotokollen stellen heraus, dass die REST-basierte Kommunikation flexibler gestaltet werden kann, wohingegen OPC UA die Standardisierung und Echtzeitfähigkeit fokussiert.

Hinsichtlich der Latenz wurde in Kapitel 5.7 eine Echtzeitanforderung für das System von $\Delta t \leq 20$ s definiert. Bezüglich der Datenformate wurde in Kapitel 3 bereits dargestellt, dass die Sensoren oft von unterschiedlichen Herstellern bereitgestellt oder gar kombiniert werden. Insbesondere die oft unterschiedlichen Datenformate begründen die Verwendung der REST-basierten Kommunikation für den digitalen Zwilling in dieser Arbeit. Für die auf der Feldebene stattfindende Kommunikation ist eine ergänzende OPC UA Integration sinnvoll, welche zwischen den Kommunikationsprotokollen übersetzt. Im Anhang E.5 ist eine beispielhafte Übersetzung zwischen OPC UA und REST dokumentiert. Diese Kombination realisiert auf der Feldebene eine echtzeitfähige Kommunikation für geringe Zykluszeiten und implementiert mittels REST die Flexibilität und Skalierbarkeit des digitalen Zwillings für weniger zeitkritische Prozesse. Das daraus resultierende Kommunikationsschema wird in den folgenden Kapiteln ergänzt, in Kapitel 6.4.1 umgesetzt sowie in Abbildung 6.4 dargestellt.

6.2 Datenbanksystem

Im Bereich der digitalen Landwirtschaft ist die Verwaltung einer großen Menge von Daten erforderlich. Datenbanken spielen daher eine Schlüsselrolle bei der Speicherung und Organisation dieser Daten, damit sie für Analysen und Entscheidungen leicht zugänglich sind.

In Kapitel 6.1 wurde beschrieben, dass bzgl. der Komponenten des Systems mit unterschiedlichen Datenformaten zu rechnen ist. Die gesammelten Daten sind in hoher Frequenz und über einen langen Zeitraum sicher zu archivieren. Es ist einzuplanen, dass ein online basiertes Konzept für eine Vielzahl von anwendenden Personen zugänglich gemacht werden muss. Weiterhin ist die Anzahl der Anfragen und zu verwaltenden Daten idealerweise auf verschiedene Server aufzuteilen.

Grundsätzlich lassen sich Datenbanken nach relationalen und nicht-relationalen Typen unterscheiden. Relationale Datenbanken implementieren die *Structured Query Language* (SQL) zur Definition der Datenstrukturen in Form von Tabellen, die untereinander verknüpft sind. Nicht-relationale Datenbanken werden hingegen als NoSQL-Datenbanken bezeichnet und benötigen kein festgelegtes tabellarisches Schema miteinander verknüpfter Daten und skalieren daher horizontal. Insbesondere NoSQL-Datenbanken liefern mit der Möglichkeit zur Verteilung der Daten und Anfragen auf mehrere Server sowie mit dem *In-Memory-Caching*, als Zwischenspeicher hoch frequentierter Abfragen zwischen Datenbank und Applikation, eine hohe Leistung bei großen Datensätzen. Die Möglichkeit der Distribution und Replikation von Daten und Anfragen ist auch bei Hardware- oder Netzwerkausfällen vorteilhaft, um eine hohe Verfügbarkeit des Systems zu gewährleisten. Insgesamt können NoSQL-Datenbanken eine leistungsstarke und flexible Plattform für die Verwaltung großer und komplexer Datensätze bieten, die für digitale Zwillinge in der Landwirtschaft erforderlich sind.

6.3 IT-Sicherheitskonzept und Kontenverwaltung

Die vorigen Kapitel weisen darauf hin, dass der digitale Zwilling unternehmensbezogene Daten verwaltet, dessen Zugang es abzusichern gilt. Die Sensibilität der Daten belegt die Relevanz der IT-Sicherheit als entscheidenden Aspekt bei der Einführung von digitalen Zwillingen in der Landwirtschaft. Digitale Zwillinge beruhen auf dem Austausch dieser sensiblen und vertraulicher Daten zwischen verschiedenen Komponenten des Systems. Ein IT-Sicherheitskonzept ist daher unerlässlich, um digitale Zwillinge in der Landwirtschaft vor Angriffen zu schützen. In diesem Kapitel werden die verschiedenen IT-Sicherheitsmaßnahmen erörtert, die für die Gewährleistung der größtmöglichen Sicherheit des digitalen Zwillings eingesetzt werden.

Mittels Passwort-Hashing kann das Passwort über eine kryptografische Einweg-Funktion verschlüsselt werden. Aufgrund der Einweg-Funktion bietet auch ein Abgreifen des Passwortes im Klartext aus der Datenbank dem Angreifer keinen Zugang zum System, da eine direkte

Rekonstruktion des gehashten Passwortes nicht möglich ist. Allerdings sind die Hashing-Algorithmen, wie z. B. MD5, SHA-* oder Bcrypt, bekannt, weshalb die Möglichkeit besteht, dass Klartextkennwörter rekonstruiert werden können. Über einen *Rainbow Table* Angriff ist die effiziente Rekonstruktion eines Klartextkennwortes über den Abgleich des aus der Datenbank abgegriffenen Hashwertes mit bekannten Wertepaaren von Kennwörtern und deren Hashwerten möglich. Eine Möglichkeit dies zu verhindern besteht darin, dass das eingegebene Passwort am Server um ein sogenanntes *Salt* oder *Pepper* ergänzt wird. Beim „Würzen“ der Passwörter fügt der Server dem Passwort eine ergänzende Bitfolge hinzu, bevor der Hashing-Algorithmus angewendet und der Abgleich oder das Abspeichern des gehashten Passwortes durchgeführt wird. Für einen erfolgreichen Angriff wäre es dann erforderlich, die *Rainbow Table* für alle Kombinationen von *Salts* und Klartextkennwörtern zu erstellen, was zu langen Rechenzeiten und hohem Speicherplatzbedarf führen würde.

Für das Absichern der Kommunikation zwischen dem Webbrowser (Client) und Server wird i. d. R. eine Kombination aus symmetrischer und asymmetrischer Verschlüsselung eingesetzt. Bei der symmetrischen Verschlüsselung erfolgt die Codierung und Decodierung mit dem gleichen (geheimen) Schlüssel, wohingegen es für die asymmetrische Verschlüsselung einen öffentlichen Schlüssel (codieren) für den Sender und einen privaten Schlüssel (decodieren) für den Empfänger gibt (MENEZES, VANSTONE und OORSCHOT, 1996). Abgegriffene Informationen zwischen den Kommunikationspartnern können somit nur mit bekannten Schlüsseln decodiert werden. Die SSL und TLS Mechanismen kombinieren diese Verfahren für eine sichere und effiziente Kommunikation.

Die Verwaltung der Konten mit den o. g. Zugangsdaten ist für einen digitalen Zwilling in der Landwirtschaft wichtig, um die Sicherheit und rollenbasierte Zugangskontrolle zu gewährleisten. Der Sicherheitsaspekt ist relevant, um das System generell vor dem unbefugten Zugriff zu schützen, wohingegen die Zugangskontrolle absichert, dass nur befugtes Personal Zugang zu sensiblen Daten und Systemfunktionen hat. Ergänzend kann die Rechenschaftspflicht bezüglich der Aktivitäten innerhalb des digitalen Zwillings realisiert werden, indem etwaige Sicherheitsvorfälle oder illegitime Aktionen erkannt und dokumentiert werden. Neben dem sicheren Speichern der Zugangsdaten und Berechtigungen ist es also erforderlich, jeder anwendenden Person eine eindeutige Identifikationsnummer zuzuweisen.

Der Zugriff externer Systeme kann über eine Programmierschnittstelle, engl. *Application Programming Interface* (API), realisiert werden, die es ebenfalls abzusichern gilt. Hierfür können z. B. die nach BRAY (2014) genormten JSON *Webtoken* (JWT) eingesetzt werden. Unter Verwendung eines Ablaufdatums und der eindeutigen Identifikationsnummer kann mittels Verschlüsselung ein zeitlich limitiertes und eindeutig zuzuordnendes JWT erstellt werden, welches der REST-Abfrage am Server hinzugefügt wird. Der Abgleich von Token und der Identifikationsnummer der anwendenden Person in der Datenbank regelt die Zugriffsrechte für das *Token* und übermittelt die Antwort an den Sender der Anfrage.

6.4 Kommunikationsarchitektur und Aufbau der Webapplikation

Eine skalierbare, interoperable und sichere Kommunikationsarchitektur erleichtert den Datenaustausch zwischen den verschiedenen Hardware- und Softwarekomponenten des Systems. Der Aspekt der Mensch-Maschine-Interaktion bei digitalen Zwillingen in der Landwirtschaft ist ebenfalls von entscheidender Bedeutung, da dieser über die Benutzungsfreundlichkeit des Systems entscheidet. In diesem Kapitel wird daher zunächst eine Kommunikationsarchitektur für das Gesamtsystem dokumentiert und darauf aufbauend ein Ansatz zur Mensch-Maschine-Interaktion mit grafischer Bedienoberfläche erörtert, welcher anstelle einer approximierten zahlenwertigen Zielgröße die Visualisierung des Strömungsfeldes über ein PINN inkludiert. Der Zugang zum digitalen Zwilling wird darüber hinaus flexibel hinsichtlich des Endgerätes ausgelegt, sodass der Zugriff über den Computer, das *Tablet* oder Mobiltelefon erfolgen kann. In Anhang E.3 ist der vollständige Programmcode für die Webapplikation und in Anhang E.4 beispielhafte Programmbausteine zur Nutzung der implementierten API hinterlegt.

6.4.1 Kommunikationsschema

Das Kommunikationsschema umreißt den Gesamtentwurf des Kommunikationssystems, einschließlich der Hardware- und Softwarekomponenten und deren Interaktion miteinander. Dazu gehört die Auswahl der geeigneten Softwarekomponenten und die Gestaltung des Kommunikationsflusses zwischen ihnen, die Konfiguration der Kommunikationsprotokolle und das Testen des Systems auf Funktionalität und Leistung. Anhand der zuvor dokumentierten Analyse der Komponenten und Kommunikationsprotokolle kann das in Abbildung 6.4 visualisierte Gesamtbild der Kommunikationsarchitektur für den digitalen Zwilling erstellt werden. Das zentrale Element ist die Logik zur Verarbeitung der Anfragen aus dem *Frontend* bzw. der API im *Backend* und die zugehörige Absicherung der Zugriffsberechtigungen und abschließende Antwort der Abfragen.

Unter Verwendung der horizontalen Skalierung der NoSQL-Server ist die Implementierung eines Servers je frei belüftetem Kuhstall möglich, welcher durch eine Eigentumskennzeichnung in den Datensätzen dem Unternehmen zugeordnet werden kann. Ein zentraler Server verwaltet die Zugangsdaten, welche mithilfe der genannten Verfahren verschlüsselt werden. Der Zugriff auf die grafische Bedienoberfläche wird über eine in *Hypertext Markup Language* (HTML) programmierte (Web-)Seite bereitgestellt, die von allen Geräten abgerufen werden kann, die über einen Webbrowser verfügen. Diese Seite kann von Servern sowohl lokal (offline) als auch online bereitgestellt werden. Der Abruf unterscheidet sich lediglich durch die Angabe der lokalen Netzwerkadresse (z. B. 198.168.0.1:5000) oder die Angabe des Domain-Namens (z. B. <https://digitaltwin.pythonanywhere.com/>).

Die durchgängige Kombination zwischen der Feld- und Prozessleitebene wird über die mit-

tels JWT abgesicherte Programmierschnittstelle realisiert, indem bestimmte Adressen des Servers für zuvor definierte Funktionen bereitgestellt werden.

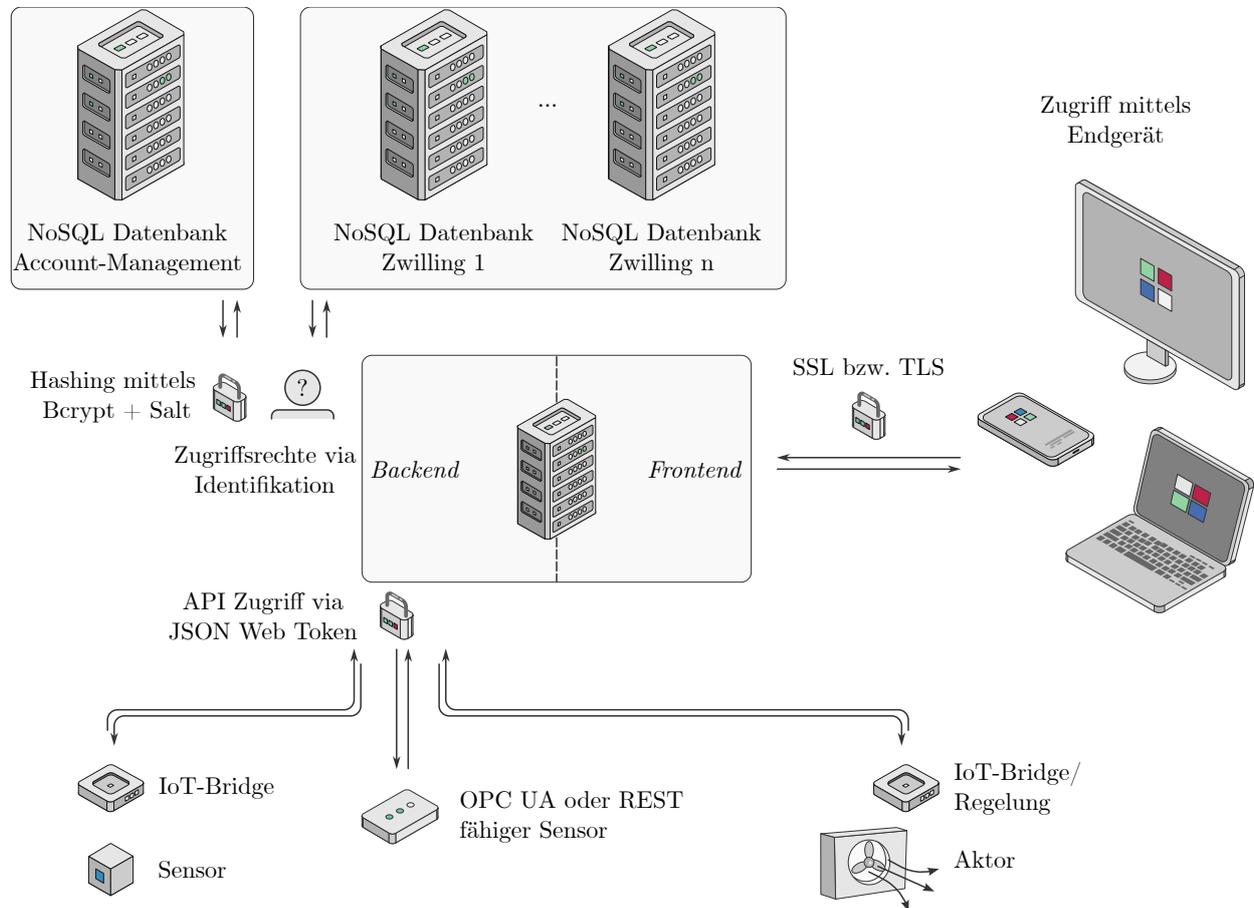


Abbildung 6.4: Visualisierung des entwickelten Kommunikationsschemas für digitale Zwillinge (eigene Darstellung)

6.4.2 Webapplikation

Dateien

Die Dateistruktur der Webapplikation ist in Abbildung 6.5 dargestellt. In höchster Dateiebene ist das Hauptprogramm („app.py“), die Softwareanforderungen („requirements.txt“) sowie die Unterordner gelistet. Der Ordner „api“ beinhaltet ein Programm mit Funktionen zur Rückgabe der Werte aus den in Kapitel 4.6 dokumentierten Regressionsfunktionen („prediction.py“) und zum Laden des PINNs und der zugehörigen Hilfsfunktionen („pinn.py“). Der Ordner „static“ stellt in weiteren Unterordnern Bilder, Logos und das PINN bereit. In „templates“ sind die das *Frontend* beschreibenden HTML-Dateien sowie ein Unterordner für die modifizierte *Login*-Seite abgelegt. Die virtuelle PYTHON-Umgebung ist im Ordner „venv“ organisiert.

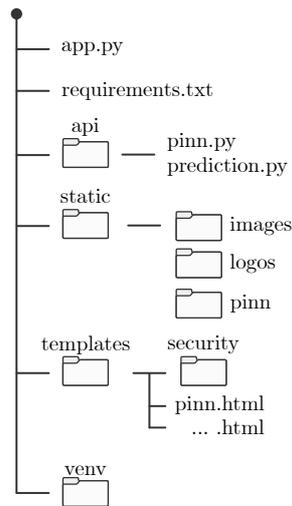


Abbildung 6.5: Dateistruktur der Webapplikation (eigene Darstellung)

Zugriff über Endgeräte

Der Zugang über die grafische Bedienoberfläche erfolgt über die *Login*-Seite (s. Abbildung 6.6). Die Angaben werden im *Backend* geprüft und bei gültigen Login-Daten der Zugang gewährt.

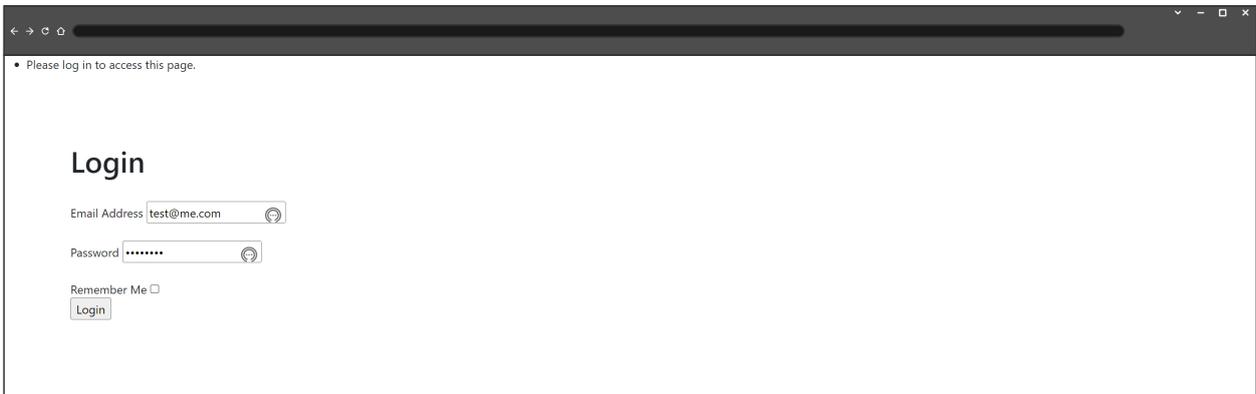


Abbildung 6.6: *Login*-Seite der entwickelten Webapplikation (eigene Darstellung)

Die Startseite wurde im Zuge der Dissertation mit einer Projektbeschreibung und den Logos der Projektbeteiligten versehen. Am oberen Bildrand in Abbildung 6.7 ist die Menüleiste zur Navigation zu erkennen, welche zu den verschiedenen Seiten der Webapplikation mit den entsprechenden Funktionalitäten führt. Über den Menüpunkt PINN-Simulation kann eine Seite aufgerufen werden, die das in Kapitel 5.8 entwickelte Modell des Stalls mit den variablen Eingangsparametern der Windgeschwindigkeit $v_m(10)$, Wickellüftung W und Dachneigungswinkel β bereitstellt und nach Einstellung der Schieberegler und Bestätigung der Taste „Berechnen“ die Vorhersage des PINNs mittels asynchroner Kommunikation zwischen *Front*- und *Backend* als Konturdiagramm auf der Seite darstellt. Dieses Konturdiagramm erlaubt es den landwirtschaftlich Beschäftigten, z. B. die vorherrschenden äußeren Umweltbedingungen

einzustellen und die qualitative und quantitative Darstellung der Strömungsgeschwindigkeiten auf einer zweidimensionalen Ebene des Stalls zu betrachten. Darüber hinaus können die Parameter variiert werden, um eine manuelle optimale Einstellung zu ermitteln. Für das Konturdiagramm wird ein Datengitter mit 90.000 Einzelwerten erstellt.

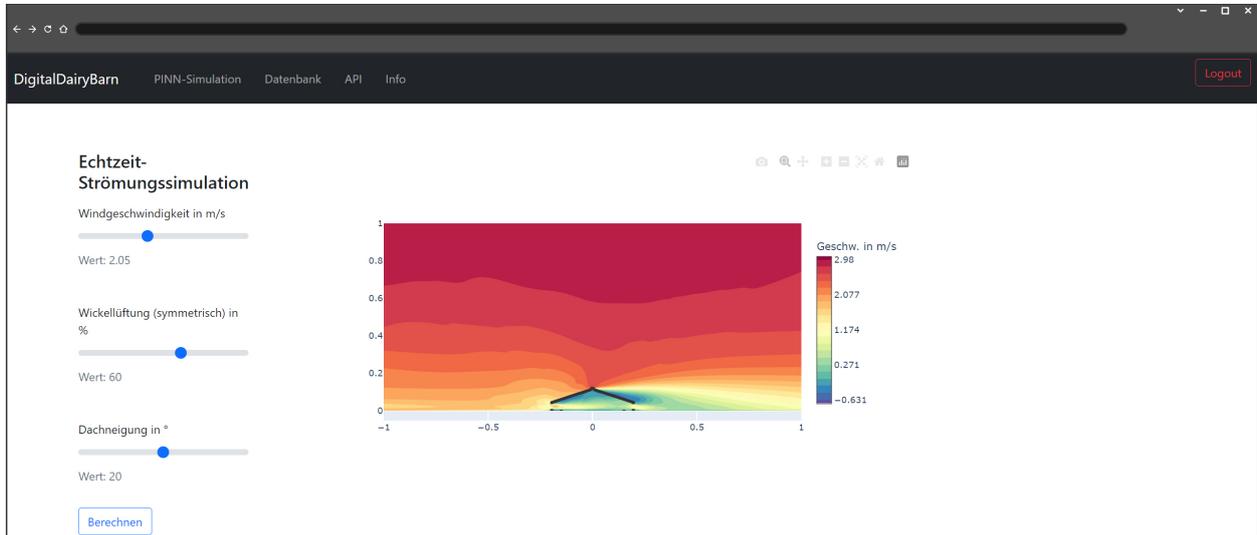


Abbildung 6.7: Darstellung der Seite der Webapplikation für die PINN-Simulation beim Zugriff über einen Computer (eigene Darstellung)

Die Datenbankseite ermöglicht den abgesicherten Zugang zu den in der Datenbank hinterlegten Sensordaten der eingeloggtten Person (s. Abbildung 6.8).

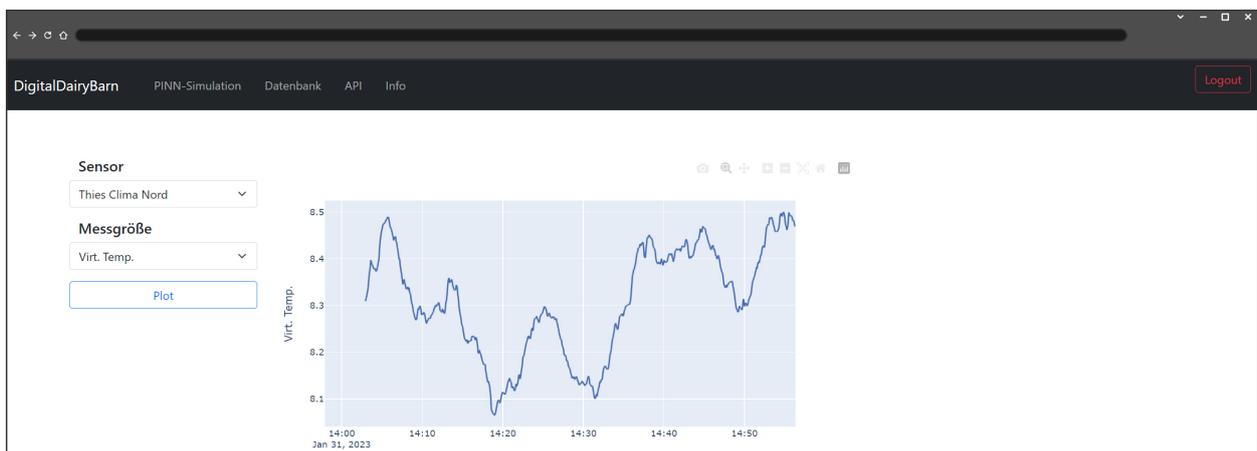


Abbildung 6.8: Datenbankseite der entwickelten Webapplikation (eigene Darstellung)

Über *Dropdown*-Listen kann zunächst der Sensor und weiter die zu visualisierende Quantität ausgewählt werden. Über die Bestätigung der Taste „Plot“ wird das Liniendiagramm mittels asynchroner Kommunikation zwischen *Front*- und *Backend* dargestellt. Analog zum Konturdiagramm des PINN kann über den Mauszeiger ein Bereich der Grafik ausgewählt

6.4.3 Programmierschnittstelle

Die Programmierschnittstelle bietet einen Zugang zu der Verarbeitungslogik und den Funktionen im *Backend* der Kommunikationsarchitektur. Der Zugang ist über REST realisiert und durch das JWT abgesichert, welches die anwendende Person über die Webapplikation erzeugen kann. Das JWT ist bei der Nutzung der REST-Anfragen am Server anzufügen, woraufhin der Server dieses prüft und entweder die Anfrage mit dem Rückgabewert der Anfrage oder dem Fehlercode 403 und einer entsprechenden Nachricht beantwortet. Für die Dokumentation der API wird die Adresse des Servers beim Bereitstellen der Webapplikation auf dem lokalen Endgerät angenommen. Wird die Webapplikation hingegen online oder im gesamten Netzwerk zur Verfügung gestellt, ist der Adressteil „http://127.0.0.1:5000/“ durch die IP-Adresse des Servers oder den Domain-Namen auszutauschen. Die Adresse für eine REST-Anfrage am Server besteht aus der IP-Adresse, der Pfadangabe und dem JWT, gemäß der Struktur „http://127.0.0.1:5000/API/get/{Parameter}?token={JWT}“.

Im Rahmen dieser Arbeit wurden Abfragen programmiert, die eine Rückgabe der Regressionsfunktionen (s. Kapitel 4.6) oder der am Server hinterlegten Sensordaten ermöglichen. Die Funktionalitäten im Programmcode (s. Anhang E.3) können jederzeit modifiziert oder erweitert werden. Für alle folgend erläuterten und implementierten Anfragen ist in Anhang E.4 eine beispielhafte Umsetzung in der Programmiersprache PYTHON dokumentiert, wobei diese in allen Programmiersprachen möglich ist, die eine REST-basierte Kommunikation unterstützen.

Rückgabe des Regressionswertes nach Plackett-Burman

Um die Regressionsfunktion aus dem Plackett-Burman-Versuchsplan zu berechnen, ist es erforderlich, die Einstellung der 14 Parameter in Form einer JSON-Datei der Anfrage anzufügen. Algorithmus 6.1 stellt die erforderlichen Schritte und die aufzurufende Adresse dar.

Algorithmus 6.1 API-Anfrage der Regressionsfunktionen nach Plackett-Burman

Erfordert: JWT, JSON-Datei

- 1: Erstellen einer JSON-Datei, die die erforderlichen Parameter enthält
`{'T': Wert T, ..., 'W_u': Wert W_u}`
 - 2: Anfügen des JWT an die Adresse
`'http://127.0.0.1:5000/API/get/regression_pb?token={JWT}'`
 - 3: Senden der Anfrage an den Server
 - 4: Prüfen der Antwort vom Server im JSON-Format
-

Rückgabe des Regressionswertes nach Box-Behnken

Analog zur Abfrage der Regressionsfunktion nach Plackett-Burman erfolgt die Anfrage der Regressionswerte aus dem Box-Behnken-Versuchsplan. Es ist ebenfalls eine JSON-Datei mit der Einstellung der drei Parameter anzufügen, woraus sich der in Algorithmus 6.2 aufgezeigte Ablauf der Anfrage ergibt.

Algorithmus 6.2 API-Anfrage der Regressionsfunktionen nach Box-Behnken

Erfordert: JWT, JSON-Datei

- 1: Erstellen einer JSON-Datei, die die erforderlichen Parameter enthält
`{'W': Wert W, 'beta': Wert beta, 'v_m': Wert v_m}`
 - 2: Anfügen des JWT an die Adresse
`'http://127.0.0.1:5000/API/get/regression_bb?token={JWT}'`
 - 3: Senden der Anfrage an den Server
 - 4: Prüfen der Antwort vom Server im JSON-Format
-

Abrufen von Sensordaten

Ergänzend zu den Regressionsfunktionen können Sensorwerte am Server abgefragt werden, indem eine entsprechende REST-Anfrage mit einer JSON-Datei gestellt wird, welche die Identifikationsnummer des Sensors und die Angabe des Zeitraumes beinhaltet. Algorithmus 6.3 erläutert die allgemeine Vorgehensweise.

Algorithmus 6.3 API-Anfrage von Sensorwerten am Server

Erfordert: JWT, JSON-Datei

- 1: Erstellen einer JSON-Datei, die die erforderlichen Parameter enthält
`{'_id': _id, 'daterange': {'datetime': [Start, Ende]}}`
 - 2: Anfügen des JWT an die Adresse
`'http://127.0.0.1:5000/API/get/sensordata?token={JWT}'`
 - 3: Senden der Anfrage an den Server
 - 4: Prüfen der Antwort vom Server im JSON-Format
-

Erstellen eines neuen Sensors

Das Senden, Bearbeiten oder Löschen von Daten erfolgt auf die gleiche Art und Weise, wie zuvor für die Anfrage von Rückgabewerten dokumentiert. Die Adressen der REST-Anfragen sind an die Befehle angelehnt, mit denen Daten gesendet, verändert und gelöscht werden können (*post*, *patch* und *delete*). Eine anwendende Person kann einen neuen Sensor und ggf. bestehende Daten mittels des in Algorithmus 6.4 dargestellten Ablaufs in der Datenbank anlegen. Bei erfolgreicher Bearbeitung enthält die Rückgabe die für den erstellten Sensor vergebene Identifikationsnummer.

Algorithmus 6.4 API-Anfrage zum Erstellen eines neuen Sensors

Erfordert: JWT, JSON-Datei

- 1: Erstellen einer JSON-Datei, welche die erforderlichen Parameter enthält
`{'sensor': Name, 'data': {'datetime': [Datum 1, ..., Datum n], 'Messgröße 1': [Wert 1, ..., Wert n], ..., 'Messgröße n': [Wert 1, ..., Wert n]}}`
 - 2: Anfügen des JWT an die Adresse `'http://127.0.0.1:5000/API/post?token={JWT}'`
 - 3: Senden der Anfrage an den Server
 - 4: Prüfen der Antwort vom Server im JSON-Format
-

Ergänzen von Daten eines bestehenden Sensors

Um einem erstellten Sensor Daten anzufügen, werden lediglich die Identifikationsnummer des Sensors und der Datensatz im JSON-Format benötigt. Der in Algorithmus 6.5 dargestellte Ablauf gibt bei einer Anfrage entweder den Fehlercode oder einen Code 200 (OK) zurück. Die Logik des Servers bearbeitet sowohl einzelne Datensätze als auch die Anfrage mit mehreren Messdaten.

Algorithmus 6.5 Ergänzen von Sensordaten mittels API

Erfordert: JWT, JSON-Datei

- 1: Erstellen einer JSON-Datei, welche die erforderlichen Parameter enthält
`{'_id': _id, 'data': {'datetime': [Datum 1, ..., Datum n], 'Messgröße 1': [Wert 1, ..., Wert n], ..., 'Messgröße n': [Wert 1, ..., Wert n]}}`
 - 2: Anfügen des JWT an die Adresse `'http://127.0.0.1:5000/API/patch?token={JWT}'`
 - 3: Senden der Anfrage an den Server
 - 4: Prüfen der Antwort vom Server im JSON-Format
-

Löschen eines Sensors aus der Datenbank

Die API stellt außerdem eine Funktion bereit, mit welcher der Sensor und die zugehörigen Daten aus der Datenbank entfernt werden können. Anhand der Identifikationsnummer und dem JWT wird geprüft, ob der Sensor der besitzhabenden Person des JWT zugeordnet ist. Die erfolgreiche Bearbeitung der Anfrage wird mit einer Nachricht im JSON-Format beantwortet.

Algorithmus 6.6 Löschen eines Sensors aus der Datenbank mittels API

Erfordert: JWT, JSON-Datei

- 1: Erstellen einer JSON-Datei mit den erforderlichen Parameter
`{'_id': _id}`
 - 2: Anfügen des JWT an die Adresse `'http://127.0.0.1:5000/API/delete?token={JWT}'`
 - 3: Senden der Anfrage an den Server
 - 4: Prüfen der Antwort vom Server im JSON-Format
-

Mithilfe dieser API-Implementierungen wird die grafische Bedienoberfläche um Funktionalitäten für die Feld- und Prozessleitebene ergänzt. Auf diese Weise können Sensordaten gesendet oder Kontrollsysteme und Aktoren mit den erforderlichen Informationen versorgt und der digitale Zwilling in Gänze umgesetzt werden.

Diskussion

Das Ziel dieser Arbeit bestand in der Entwicklung eines digitalen Zwillings zur echtzeitfähigen Vorhersage der Strömungen in und um einen frei belüfteten Kuhstall. Der wissenschaftliche Beitrag dieser Arbeit wurde durch die Formulierung von drei Forschungsfragen in Kapitel 2.6 definiert. Das Kapitel 7.1 reflektiert und ordnet die Ergebnisse der Arbeit in Bezug auf diese Forschungsfragen ein. Im darauf folgenden Kapitel 7.2 werden die Grenzen der entwickelten Methoden und Ergebnisse der Arbeit diskutiert und mögliche Lösungsansätze erörtert.

7.1 Analyse zum wissenschaftlichen Beitrag der Arbeit

In diesem Kapitel wird analysiert, inwiefern die Ergebnisse dieser Arbeit die eingangs formulierten Forschungsfragen beantworten können. Ergänzend wird reflektiert, ob und inwiefern die Ergebnisse auch im größeren wissenschaftlichen Kontext, für andere Arbeiten oder in anderen Forschungsgebieten, eingesetzt werden können.

Forschungsfrage 1: *Lassen sich die multiphysikalischen Effekte von Fluid-, Stoff- und Wärmetransport in und um einen frei belüfteten Kuhstall mittels CFD valide simulieren und der Parametereinfluss auf das Tierwohl, den Emissionsausstoß und die Luftwechselrate über eine statistische Methode systematisch bestimmen?*

Um diese Frage zu beantworten, wurden zunächst Messdaten erhoben, anhand derer die Validierung des Simulationsmodells möglich wurde. Die experimentellen Ergebnisse dieser Arbeit liefern Messdaten zu drei saisonalen Kampagnen. Innerhalb des Stalls wurden an diskreten

Messpunkten die Windgeschwindigkeit, Windrichtung, Lufttemperatur, Luftfeuchte und Methankonzentration erfasst. Zusätzlich ergänzten Daten einer lokalen Wetterstation und einer Wetterstation des DWD den Datensatz um die außerhalb des Stalls herrschende Windgeschwindigkeit, Windrichtung, Lufttemperatur und Luftfeuchte. Der Umfang unterschiedlich erhobener Quantitäten eröffnete die Möglichkeit, den Stall hinsichtlich multiphysikalischer Effekte zu analysieren und geht damit über den Stand bekannter Arbeiten hinaus (BJERG, G. ZHANG et al., 2011; FIEDLER et al., 2013; JANKE, WILLINK et al., 2020). Die exponierte Lage des Versuchstalls eliminierte den Einfluss von Nachbargebäuden und reduzierte den Einfluss von Störgeometrien auf die Messergebnisse. Die Messung der Methankonzentrationen konnte über kosteneffiziente MOX-Sensoren umgesetzt werden.

Die experimentellen Ergebnisse ermöglichen eine umfassende Einordnung zum Zustand des frei belüfteten Kuhstalls und können als Grundlage für weitere Untersuchungen und Validierungen dienen.

Die Messdaten bildeten die Grundlage der anschließenden Validierung des numerischen Simulationsmodells. Die Validierung zeigt eine hinreichende Übereinstimmung zwischen dem Modell und der Realität, für diese Bewertung wurden sowohl systematische als auch stochastische Fehler gemäß CHANG und HANNA (2004) bewertet. Auch die Iterations- und Diskretisierungsfehler erfüllen die Grenzwerte der relevanten Richtlinien (TOMINAGA et al., 2008; FRANKE und BAKLANOV, 2007). Damit belegt diese Arbeit, dass es möglich ist, die multiphysikalischen Effekte von Fluid-, Stoff- und Wärmetransport in und um einen frei belüfteten Kuhstall valide zu simulieren. Dies ergänzt den Stand der Forschung insofern, als bisher entweder einzelne Segmente (BJERG und ANDERSEN, 2010; RONG, NIELSEN und G. ZHANG, 2010; DREWRY, CHOI et al., 2018) oder ausschließlich das Innere des Stalls betrachtet wurden (DREWRY, MONDACA et al., 2018).

Das valide numerische Simulationsmodell wurde eingesetzt, um den Einfluss unterschiedlicher Faktoren systematisch zu untersuchen. Das Kapitel 4.6 definiert dafür die Zielgrößen für das Tierwohl, den Emissionsaustoss und die Durchlüftung. In einem *Screening* wurden 14 Faktoren hinsichtlich ihrer statistischen Signifikanz und Effektstärke auf die definierten Zielgrößen untersucht. Es stellte sich heraus, dass mehrere Faktoren eine statistische Signifikanz bezüglich des Tierwohls aufweisen. Nach der Lufttemperatur weist die Wickellüftung die zweitgrößte Effektstärke auf. Diese Erkenntnis ist für den energieeffizienten Betrieb frei belüfteter Kuhställe von besonderer Bedeutung, da die Wickellüftung einmalig aktuiert und anschließend abgeschaltet werden kann, bis eine Anpassung an sich ändernde Gegebenheiten erforderlich wird. Es zeigte sich außerdem, dass es möglich ist, den Stall bereits in der Planungsphase hinsichtlich des Tierwohls optimal zu gestalten. Der Dachneigungswinkel weist unter den Faktoren der Gebäudegeometrie eine hohe Effektstärke auf und ist unabhängig von der genehmigten Grundfläche eines Stalls auf alle Neubauten anzuwenden. Aufgrund dieser Anhaltspunkte des *Screenings* folgte die detaillierte Analyse in einem Antwortflächen-Versuchsplan nach Box-Behnken, welcher ergänzend zur Wickellüftung und dem Dachneigungswinkel auch die Windgeschwindigkeit als umweltbezogenen Faktor betrachtete, da diese die Transportmechanismen maßgeblich beeinflusst. Es zeigt sich, dass zwischen der Wi-

ckellüftung und dem Dachneigungswinkel ein statistisch signifikanter Wechselwirkungseffekt besteht, der bezüglich des Tierwohls statistisch signifikant ist. Hieraus leitet sich ab, dass es für unterschiedliche Dachneigungswinkel auch individuelle Regelalgorithmen für die Einstellung der Wickellüftung benötigt. Die dargelegte Methode stellt die Effektivität einer konsekutiven statistischen Versuchsplanung deutlich heraus, indem zunächst die Relevanz einer großen Anzahl unterschiedlicher Faktoren in einem *Screening* eingeordnet und im Anschluss spezifischer analysiert wurde.

Aus den Ergebnissen des vorgestellten Formalismus geht hervor, dass individualisierte Regelalgorithmen das Tierwohl in frei belüfteten Kuhställen um bis zu 4,4 % bei gleichbleibendem Energieaufwand optimieren können. Bei Ställen mit geringeren Dachneigungswinkeln geht diese Optimierung mit einer Reduktion der Methanemissionen bis zu 6,9 % einher.

Forschungsfrage 2: *Kann die Methode der PINNs dahingehend erweitert werden, dass echtzeitfähige Strömungssimulationen für komplexe Geometrien und variable Eingangsparameter mittels der für technische Anwendungen oft genutzten RANS-Gleichungen, insbesondere für hohe Reynolds-Zahlen, implementiert werden können?*

Um die Ergebnisse der RANS-Simulationen mittels PINNs zu approximieren, stellt die Arbeit eine Untersuchung geeigneter Turbulenzmodelle vor. Aus den Überlegungen zur Turbulenzmodellierung für PINNs resultiert ein PINN spezifischer Ansatz, der die Reynolds-Spannungen über pseudo-turbulente zeitinvariante Größen annähert. Das pseudo-Reynolds-Spannungs-Turbulenzmodell zeigt in der Untersuchung die geringste Varianz über die Neuinitialisierungen und bietet die größte Genauigkeit bei limitierter Datenverfügbarkeit (PIOCH, HARMENING, MÜLLER et al., 2023). Die maximal erreichte Reynolds-Zahl von $3,1 \cdot 10^6$ übertrifft die Arbeiten von EIVAZI, TAHANI et al. (2022) ($Re = 2 \cdot 10^5$) und HENNIGH et al. (2021) ($Re = 1,3 \cdot 10^4$) um mehr als eine Größenordnung. Die Variabilität für unterschiedliche Geometrien oder fluidmechanische Eingangsparameter wurde über das Konzept des Hyperwürfels realisiert. Hierbei wurden die Eingangsparameter als zusätzliche Neuronen am Netz ergänzt und dementsprechend die Datensätze der Koordinaten um weitere Spalten dieser zusätzlichen Dimensionen erweitert. Eine solche Flexibilität stellen bisher lediglich HENNIGH et al. (2021) und WANDEL, WEINMANN und KLEIN (2021) vor, die insbesondere die für diese Arbeit relevanten hohen Reynolds-Zahlen nicht betrachten. Durch die Verwendung weiterer Dimensionen am Hyperwürfel wurde es erforderlich, die heterogenen Eingaben und Ausgaben am neuronalen Netz zu skalieren, um einen Bias in der Optimierung der Gewichtungen zu verhindern. Diese Arbeit weist hierfür eine Methode aus, bei der das reale Fluid durch ein Fluid mit künstlicher Viskosität ersetzt wird, um die Größen zu skalieren und dennoch eine mechanische Ähnlichkeit der Strömung beizubehalten. Es wird belegt, dass der Ansatz eine hohe Stabilität und Genauigkeit über einen Reynolds-Zahlbereich von $1 \cdot 10^2$ bis $1 \cdot 10^6$ aufweist (HARMENING, PIOCH, FUHRIG et al., 2023), womit die Ergebnisse der Untersuchung von LAUBSCHER und ROUSSEAU (2021) bestätigt und um weitere Größenordnungen der Reynolds-Zahlen erweitert werden.

Die Entwicklung des Echtzeitmodells demonstriert eine neuartige Methode, mit der das Ergebnis einer RANS-Simulation bei hohen Reynolds-Zahlen in wenigen Millisekunden valide approximiert werden kann. Die Methode ist darüber hinaus universell auf die Strömung um unterschiedlichste Geometrien anwendbar und nicht auf den Anwendungsfall des frei belüfteten Kuhstalls beschränkt.

Forschungsfrage 3: *Lässt sich ein Kommunikationsschema für frei belüftete Kuhställe entwickeln, das die o. g. echtzeitfähigen Simulationen implementiert, somit den digitalen Zwilling ermöglicht, den Zugriff flexibel off- und online erlaubt sowie die Problematik der Datensicherheit und Mensch-Maschine-Interaktion berücksichtigt?*

Für die Entwicklung einer IoT-gerechten Kommunikationsarchitektur analysiert diese Arbeit zunächst die an der Kommunikation beteiligten Komponenten, Kommunikationsprotokolle und Datenbanksysteme. Die zentralen Elemente dieser Kommunikationsarchitektur sind zum einen die Webapplikation (*Frontend*) und zum anderen die Verarbeitungslogik im *Backend*. Um aufzuzeigen, dass diese Kommunikationsarchitektur flexibel online- und offline-fähig ist, wurde die programmierte Software auf unterschiedlicher Hardware ausgeführt und sowohl in lokalen Netzwerken als auch im Internet bereitgestellt. Diese Flexibilität bietet keine der analysierten Arbeiten zu digitalen Zwillingen in der Landwirtschaft (s. Tabelle 2.4, Seite 38). Der Zugang zum System ist nach aktuellen kryptografischen Standards sicher gestaltet. Dieser sichere Zugang zur Webapplikation ermöglicht u. a. den Zugriff auf die Visualisierung der aktuellen Strömungsbedingungen, die mithilfe des implementierten echtzeitfähigen Modells berechnet werden. Über diese Arbeit hinaus wird das Thema der IT-Sicherheit für digitale Zwillinge in der Landwirtschaft konkret nur von VERDOUW et al. (2021) behandelt. VERDOUW et al. (2021) stellen außerdem eine Definition der Abstufungen digitaler Zwillinge bereit (s. Kapitel 2.1). Gemäß dieser Definition kann die Realisierung des digitalen Zwillinges in dieser Arbeit als überwachender digitaler Zwilling bezeichnet werden. Darüber hinaus berücksichtigt die vorliegende Arbeit bereits mögliche Weiterentwicklungen zum prädiktiven oder präskriptiven digitalen Zwilling. Diese könnten ermöglicht werden, indem z. B. die Wettervorhersage in der Datenbank hinterlegt oder Regelalgorithmen für die Aktoren eingebunden werden. Für die Milchviehhaltung stellt diese Arbeit die erste, dem Autor bekannte, Realisierung eines digitalen Zwillinges zur Betrachtung der mikroklimatischen Bedingungen dar. In anderen Forschungsarbeiten zum Thema digitaler Zwillinge für die Tierhaltung lag der Fokus bisher hauptsächlich auf der Schweinehaltung (JO, D.-H. PARK, H. PARK und S.-H. KIM, 2018; JO, D.-H. PARK, H. PARK, KWAK et al., 2019; ERDÉLYI und JÁNOSI, 2019).

Die Methode dieser Arbeit reicht vom Konzept, über die Software bis zur Realisierung auf entsprechender Hardware. Der überwachende digitale Zwilling kann online oder offline bereitgestellt werden und implementiert Physik-informierte neuronale Netze, mit denen die echtzeitfähige Visualisierung der aktuellen Strömungsbedingungen ermöglicht wird.

7.2 Limitationen und Lösungsansätze

Dieses Kapitel analysiert die Grenzen der dargelegten Methoden und weist auf mögliche Lösungsansätze hin, die im Kapitel 8.2 für zukünftige Arbeiten zu möglichen Projekten zusammengefasst werden. Die Analyse wird dabei der Struktur der Hauptkapitel dieser Arbeit folgen.

7.2.1 Experimentelle Ergebnisse

Die Auswertung der via Ultraschallsensoren gemessenen Windgeschwindigkeiten zeigt im Boxplot deutliche Unterschiede an Luv und Lee sowie einen dezentralen Median der Messkampagne im Sommer. Dieser Umstand weist auf die zeitliche Veränderung der äußeren Umwelteinflüsse im Laufe der Messung hin. Aufgrund der stationären numerischen Simulation mussten die Messwerte der Kampagne daher für eine Validierung ausgeschlossen werden. Um diese Problematik zukünftig zu vermeiden, wäre es denkbar, weitere Ultraschallsensoren einzusetzen und eine zeitgleiche Messung an allen Positionen durchzuführen. Da die Geschwindigkeitskomponente der Elevation über alle Messkampagnen und Positionen Werte nahe null aufwies, wäre es denkbar, dies mit 2D-Ultraschallanemometern oder geeigneten Schalenkreuzanemometern kosteneffizient umzusetzen, wie von HEIDENREICH et al. (2008) dargelegt.

Die Messwerte der Luftfeuchtigkeit weisen zwischen dem Innenraum des Stalls und der Wetterstation Differenzen auf, die sich z. T. deutlich zwischen den Kampagnen unterscheiden. Die Differenzen der Übergangsmesskampagne fallen gegenüber der Sommer- und Wintermesskampagne deutlicher aus. Möglicherweise liegt dies daran, dass Trägheiten im System eine Rolle spielen, da sich die Temperatur und damit auch die emittierte Feuchte aus dem Lagerungssystem des Dungs erst allmählich der Umgebungstemperatur angleichen. Auch der Füllstand im Lagerungssystem und die Zusammensetzung sowie Konsistenz haben einen Einfluss auf die Emissionen und Luftfeuchte (EGGLESTON et al., 2006; LENGERS, SCHIEFLER und BÜSCHER, 2013). Diese Werte aufzuzeichnen, könnte zu weiteren Erkenntnissen über das System und zusätzlichen Möglichkeiten der Modellierung und Validierung führen.

Die Methansensoren wurden vor Beginn jeder Messkampagne, für die am jeweiligen Tag geltenden Bedingungen kalibriert, indem der Referenzwiderstand an der Umgebungsluft ermittelt wurde. Der Hersteller des Sensors stellt ergänzend den Einfluss von Temperatur und Luftfeuchte auf das Widerstandsverhältnis für einige Kombinationen dieser Größen dar. Eine Ergänzung dieser limitiert verfügbaren Kombinationen könnte möglicherweise für eine Interpolation genutzt werden, um eine kontinuierliche und korrigierte Messung der Methankonzentration hinsichtlich der aktuell vorherrschenden Bedingungen zu erfassen. Insbesondere für mehrtägige Messungen würde dies einen großen Vorteil bedeuten, da die Schwankungen von Lufttemperatur und Luftfeuchte ohne eine erneute Kalibrierung des Sensors zu berücksichtigen wären.

7.2.2 Numerische Ergebnisse

Die Ergebnisse dieser Arbeit und der Stand der Forschung zeigen auf, dass es in vielen Fällen möglich ist, die Durchströmung landwirtschaftlicher Gebäude über stationäre RANS-Simulationen valide abzubilden. Trotz der Validität der Ergebnisse zeigen sich Abweichungen zwischen den Messungen und der numerischen Simulation. Neben den Modellannahmen und -vereinfachungen könnten auch zeitabhängige Effekte und ungenaue oder vereinfachte Randbedingungen zu den Abweichungen zwischen Simulationen und Messungen beigetragen haben, die im Folgenden diskutiert werden.

Ein interessanter Aspekt wäre es, den temporären Einfluss einer Böe zu untersuchen, die vorübergehende und starke Änderungen der Durchströmung bewirken können. Um die Bögigkeit des Windes in der Simulation zu berücksichtigen, wäre es erforderlich, eine instationäre Simulation durchzuführen, welche wiederum mit einem höheren numerischen Aufwand einhergeht. Für eine geringe Anzahl zu untersuchender Varianten wäre der numerische Aufwand einer *Large Eddy Simulation* ggf. vertretbar, wohingegen es für die statistische Versuchsplanung anderer Lösungsansätze bedarf.

Die im Rahmen der Arbeit verwendete Software ließ eine Berücksichtigung der Kuhherde als poröses Medium für das verwendete Turbulenzmodell nicht zu. Sollte dies in Zukunft möglich werden, folgt daraus ein numerisch weniger aufwändigerer Ansatz, der allerdings auch weniger detailliert wäre (MONDACA, CHOI und COOK, 2019). Über anisotrope Einstellungen des porösen Materials bieten sich umfangreiche Möglichkeiten, um den Strömungswiderstand der Strukturen im Gehege mitzubersichtigen, wie DOUMBIA, JANKE, YI, AMON et al. (2021) darlegen. Indem diese kleineren Strukturen durch das poröse Material abgebildet werden, könnte der Bias der Querbelüftung, wie in Kapitel 4.5 dokumentiert, verringert werden. Des Weiteren sind die Wickellüftungen als Festkörper in der Simulation implementiert, wohingegen diese in der Realität aus dicht gewobenem Textil bestehen. Die Wickellüftungen könnten demnach über ein poröses Medium ebenfalls akkurater dargestellt werden.

Bei der Simulation der Wärmetransportprozesse ist im Rahmen der Arbeit ausschließlich die Konvektion betrachtet worden. Zwar ist dies mit einem Anteil von bis zu 80 % der dominante Wärmetransportmechanismus im System, jedoch könnte die ergänzende Berücksichtigung der Radiation, gerade für geringere Windgeschwindigkeiten, eine erhöhte Genauigkeit bedeuten, wie NORTON, GRANT et al. (2010) darlegen. Eine weitere mögliche Ergänzung wäre die detaillierte Betrachtung des Lagerungssystems in der Simulation, um die in Kapitel 7.2.1 erwähnte Trägheit und den Einfluss des Füllstands im Lagerungssystem und der Konsistenz des Dungs detailliert zu berücksichtigen. Die Genauigkeit der Ergebnisse könnte weiter gesteigert werden, indem ergänzend die Wärmetransmission zwischen dem Inneren des Stalls und seiner Umgebung berücksichtigt würde.

7.2.3 Echtzeitfähige Strömungsvorhersage

Die in dieser Arbeit entwickelte Methode für eine echtzeitfähige Approximation stationärer Strömungsfelder ist universell hinsichtlich unterschiedlicher Geometrien und Parameter anwendbar. Die Methode ist im Rahmen der Arbeit jedoch ausschließlich auf zweidimensionale Strömungsfelder angewendet worden, während die Arbeiten von WANDEL, WEINMANN und KLEIN (2021) und HENNIGH et al. (2021) nahelegen, dass eine Erweiterung der Methode auf den dreidimensionalen Raum denkbar wäre.

Eine Einschränkung der Methode ergibt sich aus der erreichten Genauigkeit. Die Fehler der Vorhersage sind auf die numerischen Eigenschaften des Optimierungsproblems zurückzuführen. Dieses Optimierungsproblem ist durch verschiedene Parameter, wie z. B. die zu Grunde liegenden Gleichungen, Randbedingungen, Optimierer, die Aktivierungsfunktion, Netztopologie, Lernrate, Trainingspunkte und Verlustfunktion gekennzeichnet (HARMENING, PIOCH, FUHRIG et al., 2023). In der daraus resultierenden nicht konvexen Optimierungslandschaft kann es vorkommen, dass die Optimierung nicht das globale Minimum erreicht, was zu einem signifikanten Fehler führen kann. Deshalb ist es wichtig, die Sensitivität der Parameter in Bezug auf die zu erreichende Genauigkeit zu betrachten.

Um den Einfluss der Netztopologie umfassender zu untersuchen, wäre es möglich, die in dieser Arbeit verwendeten *feed-forward* Netze durch U- oder Fourier-*Feature*-Netze zu ersetzen. U-Netze verwenden einen anderen Aufbau der verdeckten Schichten, womit für viele Anwendungsfälle der PINNs gute Ergebnisse erzielt werden konnten (Y. ZHU et al., 2019; R. WANG et al., 2020; WANDEL, WEINMANN und KLEIN, 2021). Fourier-*Feature*-Netze transformieren die Eingangsparameter in den Frequenzbereich, woraus sich insbesondere für periodische Strukturen Vorteile ergeben (HENNIGH et al., 2021).

Die Optimierung zeigte für die untersuchten Fälle auf, dass sich die großen Residuen auf kleine Regionen mit hohen Gradienten beschränken. Eine Möglichkeit, dem zu begegnen, wäre das von LU et al. (2021) vorgeschlagene *residual-based adaptive refinement*. Bei dieser Methode werden zusätzliche Trainingspunkte für die PDEs in Regionen mit großen Residuen zum Trainingssatz hinzugefügt, womit geringere finale Residuen erreicht werden (C. WU et al., 2023).

Die Gewichtungen der zusammengesetzten Verlustfunktion können a priori festgelegt werden, wie in dieser Arbeit dargelegt, oder während des Trainings angepasst werden. XIANG et al. (2022) stellen dafür eine selbst anpassende Methode vor, die auf einem Gaußschen Wahrscheinlichkeitsmodell beruht und erzielen so eine höhere Genauigkeit.

Die Limitierung durch die Hardware begrenzt den Umfang der Trainingsdaten und die Anzahl der Neuronen. Das Training mittels Grafikkarten beschleunigt den Trainingsprozess, erfordert aber im Gegenzug einen entsprechend großen Grafikkartenspeicher. Für den PINN-Ansatz zeigen HARMENING, PIOCH, FUHRIG et al. (2023) auf, dass der asymptotische Bereich der Genauigkeit für zweidimensionale Strömungsfelder ab 800 Punkten je Ebene erreicht wird. Die Hardwareanforderungen skalieren daher mit der Summe der Ebenen, die wiederum durch

die Anzahl und Auflösung der Parameter bestimmt werden. Dem könnte mit speziellen Grafikkarten begegnet werden, welche für das Training neuronaler Netze optimiert sind und einen großen Grafikspeicher und die zur Verarbeitung erforderlichen Tensor-Kerne bieten. Diese Grafikkarten sind üblicherweise parallelisierbar, woraus eine einfache Skalierung der Hardware resultiert. Die Limitierung bezieht sich ausschließlich auf den Trainingsprozess des neuronalen Netzes und stellt keine Hürde für die anschließende Bereitstellung des bereits trainierten Modells dar.

7.2.4 Digitaler Zwilling

Im Kapitel 6 wurde ein überwachender digitaler Zwilling für frei belüftete Kuhställe entwickelt, der die Sensordaten und ein echtzeitfähiges Simulationsmodell implementiert und über eine Programmierschnittstelle bereitstellt.

Eine besondere Herausforderung bei der Umsetzung war die Verknüpfung der Komponenten auf der Feld- und Prozessleitebene, die zum Teil mit nativen Kommunikationsschnittstellen ausgerüstet waren. Am Beispiel der OPC UA Schnittstelle konnte dargelegt werden, wie eine Übersetzung zwischen unterschiedlichen Schnittstellen umgesetzt werden kann. Für den Einsatz im Stall könnte es jedoch erforderlich sein, weitere Kommunikationsprotokolle zu ergänzen, wie das von MOSHREFZADEH, MACHL und GACKSTETTER (2020) eingesetzte *Long Range Wide Area Network*. So könnte das Konzept sukzessive interoperabel gestaltet werden.

Im Rahmen der Arbeit konnte die Stufe des überwachenden digitalen Zwillings nach VERDOUW et al. (2021) realisiert werden. Der historische digitale Zwilling ergibt sich aus dem Konzept ebenso, da die Messwerte in der Datenbank archiviert wurden und visualisiert werden können.

Die Interpretation des Strömungsfeldes und die Optimierung der Parametereinstellung obliegen in der aktuellen Umsetzung jedoch den anwendenden Personen der Webapplikation. Eine Erweiterung zum prädiktiven digitalen Zwilling wäre möglich, indem die standortspezifische Wetterprognose abgerufen und in der Datenbank hinterlegt würde, um darauf basierend die Durchströmung zu visualisieren. Für den präskriptiven digitalen Zwilling wäre es erforderlich, die Kontrolleinheiten und Aktoren einzubeziehen, um zu intervenieren und die Einstellung der Lüftungselemente vom digitalen Zwilling ausgehend vorzunehmen.

Hinsichtlich der IT-Sicherheit wurden aktuelle kryptografische Methoden implementiert. Allerdings wurden keine umfangreichen Penetrationstests oder ein Vulnerabilitätsscan durchgeführt, um etwaige Sicherheitslücken oder Schwachstellen aufzudecken. Für den Einsatz der Software in den landwirtschaftlichen Betrieben wäre dies zu ergänzen.

8.1 Zusammenfassung

Die Weltbevölkerung wächst stetig. Gleichzeitig erfordert der Klimawandel, dass wir effizienter mit den vorhandenen Ressourcen umgehen, um die Ernährung der Menschen sicherzustellen. Eine Möglichkeit, die Effizienz zu steigern, besteht in einer umfassenderen Digitalisierung der Landwirtschaft. Beispiele für essenzielle Werkzeuge zur Digitalisierung der Landwirtschaft sind die Simulation landwirtschaftlicher Anwendungsfälle, die Implementierung maschinellen Lernens und eine durchgängige Datenverfügbarkeit.

Die vorliegende Arbeit untersuchte drei zentrale Forschungsfragen. Erstens, ob die multiphysikalischen Effekte von Fluid-, Stoff- und Wärmetransport in und um einen frei belüfteten Kuhstall valide simuliert werden können und der Einfluss von Parametern auf das Tierwohl, den Emissionsausstoß und die Luftwechselrate systematisch bestimmt werden kann. Zweitens, ob die Methode der Physik-informierten neuronalen Netze erweitert werden kann, um echtzeitfähige Strömungssimulationen für komplexe Geometrien und variable Eingangsparameter, insbesondere für hohe Reynolds-Zahlen, zu implementieren. Und drittens, ob eine Kommunikationsarchitektur für frei belüftete Kuhställe entwickelt werden kann, die die echtzeitfähige Strömungssimulation integriert, den Zugriff flexibel offline und online erlaubt sowie die Problematik der Datensicherheit und Mensch-Maschine-Interaktion berücksichtigt. Um diese drei Forschungsfragen zu beantworten, waren vier Entwicklungsschritte erforderlich, die in Abbildung 8.1 zusammenfassend visualisiert sind.

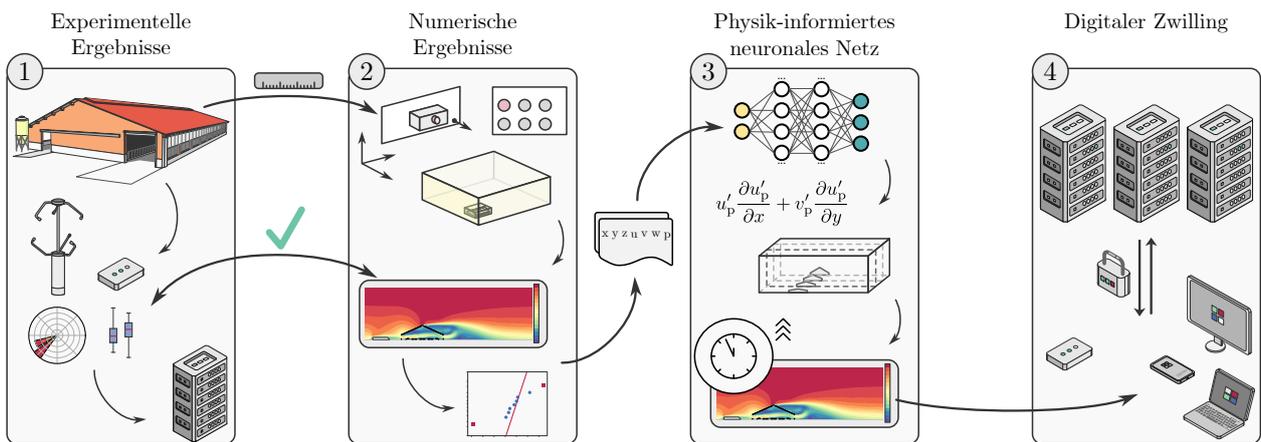


Abbildung 8.1: Visuelle Zusammenfassung der Arbeit (eigene Darstellung)

Der erste Entwicklungsschritt dieser Arbeit hatte das Ziel, die realen Bedingungen in einem Versuchsstall messtechnisch zu erfassen, um eine spätere Validierung des digitalen Modells zu ermöglichen. Um dieses Ziel zu erreichen, wurden drei saisonal unterschiedliche Messkampagnen im Versuchsstall durchgeführt – im Sommer, Übergang und Winter. In diesen Kampagnen wurden Messwerte zur Windgeschwindigkeit, Windrichtung, Lufttemperatur, Luftfeuchte und Methankonzentration erfasst. Zur Messung der Methankonzentration wurde im Rahmen dieser Arbeit ein Methansensor entwickelt und kalibriert, während die übrigen Messwerte mithilfe industriell verfügbarer Sensoren erhoben wurden. Die Sensoren im Versuchsstall umfassten zwei 3D-Ultraschallanemometer, zwei Methansensoren sowie acht Temperatur- und Luftfeuchtesensoren. Die Messwerte außerhalb des Stalls wurden durch eine lokale Wetterstation und eine Wetterstation des Deutschen Wetterdienstes erfasst. Anhand der Messdaten der Wetterstationen wurden die umweltbezogenen Randbedingungen für die nachfolgende numerische Simulation abgeleitet.

Im zweiten Entwicklungsschritt wurde das digitale Modell erstellt, mit dem Ziel, verschiedene Randbedingungen gezielt einzustellen und den Einfluss dieser Randbedingungen auf die Strömung systematisch zu untersuchen.

Da die Umweltbedingungen, wie Windgeschwindigkeit und -richtung, naturgegeben sind, bietet die numerische Strömungssimulation eine Möglichkeit, diese Umwelteinflüsse in einem digitalen Modell gezielt zu variieren. Die numerische Strömungssimulation umfasst unter anderem die geometrische Beschreibung des Strömungsraums sowie die Einstellungen für das Lösungsverfahren und die damit verbundenen Randbedingungen. Die geometrische Beschreibung des Strömungsraums wurde mithilfe eines CAD-Modells realisiert. Um verschiedene Geometrien des Stalls systematisch zu untersuchen und deren Einfluss auf die Strömung zu analysieren, wurde das CAD-Modell so konstruiert, dass es parametrierbar ist. Durch die Änderung eines der zehn geometrischen Parameter wurde das CAD-Modell automatisch neu aufgebaut.

Um die Strömung für die komplexen Geometrien des Stalls numerisch zu lösen, musste

der Strömungsraum diskretisiert (vernetzt) werden. Die Lösung der Reynolds-gemittelten Navier-Stokes-Gleichungen auf diesem Rechenetz lieferte die stationären Ergebnisse für die Geschwindigkeitskomponenten und den Druck. Anhand dieser Ergebnisse konnte die Luftwechselrate für den Stall berechnet werden. Ferner sollten zwei weitere Zielgrößen analysiert werden, die das Tierwohl und die Methan-Emissionen quantifizieren. Das Tierwohl wurde anhand des Temperatur-Luftfeuchte-Index objektiv bewertet, während die Methan-Emissionen als Netto-Bilanz des ein- und ausströmenden Massestroms an Methan bestimmt wurden. Diese zusätzlichen Zielgrößen konnten durch eine multiphysikalisch gekoppelte Berechnung ermittelt werden, bei der im numerischen Simulationsmodell der Fluidtransport in einer sequentiellen Berechnung mit dem Wärme- und Stofftransport gekoppelt wurde.

Damit die Genauigkeit der Ergebnisse des digitalen Modells bewertet werden konnte, wurde eine Verifizierung und Validierung durchgeführt. Die Verifizierung belegte, dass die Iterations- und Diskretisierungsfehler hinreichend klein waren. Der Grenzwert für das Konvergenzkriterium von $1 \cdot 10^{-4}$ wurde eingehalten und die Extrapolation nach Richardson wies genügend geringe Fehler ($\bar{\epsilon}_{\text{extr.}} \leq 5,78 \%$) und Unsicherheiten ($\overline{\text{GCI}} \leq \pm 7,84 \%$) auf. In der darauf folgenden Validierung wurden stochastische und systematische Fehler zwischen dem digitalen Modell und den messtechnisch ermittelten realen Werten quantifiziert. Die Grenzwerte der Metriken wurden eingehalten.

Das validierte und verifizierte Modell bildete den Ausgangspunkt für die statistische Versuchsplanung. Das Ziel der Versuchsplanung war es, die Einflüsse unterschiedlicher Faktoren auf die oben genannten Zielgrößen systematisch zu untersuchen, um letztlich eine Optimierung zu ermöglichen. In einem *Screening* wurden dazu 14 Faktoren in 20 verschiedenen numerischen Simulationen variiert und die Ergebnisse der Zielgrößen statistisch ausgewertet, um signifikante Faktoren zu ermitteln. Im darauf folgenden Box-Behnken-Versuchsplan wurden die Faktoren Wickellüftung, Dachneigungswinkel und Windgeschwindigkeit auf Wechselwirkungen und quadratische Zusammenhänge hin analysiert. Die Auswertung des Indizes für das Tierwohl ergab eine Wechselwirkung zwischen dem Dachneigungswinkel und der Einstellung der Wickellüftung. Diese Erkenntnis ist für die Planung neuer Ställe und die Regelung der Lüftungselemente von entscheidender Bedeutung, da sich dadurch das Tierwohl energieeffizient um 4,4 % optimieren und die Methan-Emission um 6,9 % reduzieren lässt. Um dieses Potenzial voll ausschöpfen zu können, ist es jedoch erforderlich, die Luftführung im Stall bei beliebigen Windgeschwindigkeiten, Dachneigungswinkeln und Einstellung der Wickellüftungen in Echtzeit zu analysieren und daraufhin optimal einzustellen.

Der dritte Entwicklungsschritt ermöglichte die Analyse der Luftführung in Echtzeit. Die Echtzeitanforderung wurde als Zeitspanne einer Böe definiert, woraus sich der Anspruch ergab, das Strömungsfeld innerhalb von $\Delta t_{\text{max}} \leq 20$ s zu berechnen und zu visualisieren. Zur Realisierung dieser Anforderung nutzte die entwickelte Methode die Simulationsergebnisse der Durchläufe des Box-Behnken-Versuchsplans, um damit ein Physik-informiertes neuronales Netz zu trainieren. Das Physik-informierte neuronale Netz glich die limitiert verfügbaren Referenzdaten aus, indem die zugrunde liegenden Reynolds-gemittelten Navier-Stokes-Gleichungen in die Verlustfunktion des neuronalen Netzes inkorporiert wurden. Um das

Schließungsproblem der Reynolds-gemittelten Navier-Stokes-Gleichungen speziell für Physik-informierte neuronale Netze zu lösen, wurden verschiedene klassische und spezifisch für Physik-informierte neuronale Netze entwickelte Turbulenzmodelle analysiert. Das neuartige pseudo-Reynolds-Spannungs-Turbulenzmodell zeigte in der Analyse eine höhere Stabilität hinsichtlich der Varianz zwischen den Neuinitialisierungen des neuronalen Netzes und konnte die Strömungen im Stall mit einer maximalen Reynolds-Zahl von $3,1 \cdot 10^6$ valide approximieren.

Die heterogenen Werte der Ein- und Ausgangsgrößen am neuronalen Netz führten zu einem Bias in der Optimierung. Dies lässt sich mit der daraus resultierenden unterschiedlichen Neigung in den Dimensionen der Optimierungslandschaft des neuronalen Netzes erklären. Dem Bias konnte durch die Anwendung der Ähnlichkeitsgesetze nach Reynolds begegnet werden, anhand dessen die Ein- und Ausgangsgrößen auf einen Wertebereich zwischen etwa null und eins skaliert wurden. Es konnte belegt werden, dass durch die Skalierung der Werte über einen Bereich der Reynolds-Zahlen von $1 \cdot 10^2$ bis $1 \cdot 10^6$ niedrigere Verlustmetriken erreicht werden. Diese Skalierung ist zudem reversibel, sodass die Visualisierung in den realen Einheiten erfolgt.

Ebenso konnte gezeigt werden, dass unterschiedliche Strategien zur Auswahl der Trainingspunkte die Genauigkeit der Verlustmetriken beeinflussen. Über eine randomisierte Auswahl der Punkte aus dem Rechenetz konnte die größte Genauigkeit erreicht werden.

Damit das echtzeitfähige Modell die Strömung für unterschiedliche Einstellungen der Wickelöffnungen, des Dachneigungswinkels und der Windgeschwindigkeit berechnen konnte, wurde das Konzept des Hyperwürfels auf die Problematik angewendet. Der Hyperwürfel erweitert die Dimensionen der kartesischen Koordinaten um die genannten Parameter. Dafür war es einerseits erforderlich, für jeden Parameter ein zusätzliches Neuron am Eingang des Physik-informierten neuronalen Netzes hinzuzufügen, und andererseits mussten die Werte der Parameter in den Trainingsdaten berücksichtigt werden.

Es konnte insgesamt eine valide Methode für die Interpolation im Hyperwürfel nachgewiesen werden, wobei relative Fehler im Bereich von 5,0 % bis 15,0 % im Vergleich zur Referenzgeschwindigkeit auftraten. Diese valide Vorhersage erfolgte für $1 \cdot 10^4$ Punkte im Strömungsraum innerhalb von $6,5 \cdot 10^{-3}$ s.

Der abschließende vierte Entwicklungsschritt umfasste die Entwicklung des digitalen Zwillings für frei belüftete Kuhställe. Der digitale Zwilling sollte archivierte Messergebnisse und die Visualisierung der echtzeitfähigen Strömungsvorhersage für landwirtschaftlich Beschäftigte online oder offline zur Verfügung stellen und einen ergänzenden Zugang zu den Komponenten der Prozess- und Feldebene bieten. Um dies zu realisieren, wurden zunächst die relevanten Komponenten identifiziert, Kommunikationsprotokolle analysiert, ein Datenbanksystem ausgewählt und der Zugang mithilfe moderner kryptografischer Methoden abgesichert. Diese Arbeitsschritte führten zu einer Kommunikationsarchitektur, die im *Frontend* eine Webapplikation zur Mensch-Maschine-Interaktion bereitstellt. Diese Webapplikation ermöglicht landwirtschaftlich Beschäftigten den Zugriff auf die Funktionalitäten im *Backend* der Kommunikationsarchitektur. Die Funktionalitäten umfassen unter anderem die Abfrage der

archivierten Messergebnisse oder die echtzeitfähige Visualisierung der Strömungsergebnisse. Das *Backend* übermittelt die Antwort zu den Funktionen im Format des *Representational State Transfer*, entweder an das *Frontend* oder an die Komponenten der Prozess- und Feldebene. Die entwickelte Webapplikation bietet somit in Zusammenarbeit mit dem *Backend*, der NoSQL-Datenbank und einer Programmierschnittstelle die notwendige Datenverfügbarkeit für die Darstellung auf den Endgeräten bzw. die Beantwortung der Anfragen aus der Prozess- und Feldebene. Die Erprobung des digitalen Zwillinges erfolgte offline und online und belegt ein flexibles Konzept.

Diese Arbeit verfolgte einen innovativen Ansatz, um die Herausforderungen der echtzeitfähigen Strömungsvorhersage in und um Stallungen zu bewältigen und über einen digitalen Zwilling bereitzustellen. Der Fokus der Arbeit lag stets auf der Milchviehhaltung, mit dem Ziel, eine nachhaltige Ernährung der wachsenden Weltbevölkerung sicherzustellen. Die vorliegende Arbeit eröffnet jedoch nicht nur neue Perspektiven für die Tierhaltungsindustrie, sondern liefert mit den Erkenntnissen zu Physik-informierten neuronalen Netzen einen allgemeinen Beitrag zur Forschung.

8.2 Zukünftige Arbeiten

Die vorliegende Arbeit demonstriert eine leistungsstarke und effektive Methode, mit der die Strömungen in und um frei belüftete Kuhställe in Echtzeit vorhergesagt und diese für landwirtschaftlich Beschäftigte über einen digitalen Zwilling zur Verfügung gestellt werden können. In Kapitel 7.2 wurden die Grenzen der Methode diskutiert und bereits Lösungsansätze dazu vorgestellt. In diesem Kapitel werden die einzelnen Lösungsansätze aufgegriffen und in einem größeren Kontext zu Konzepten für zukünftige Forschungsarbeiten zusammengefasst.

Mögliche zukünftige Arbeiten könnten die Methode durch umfassendere Messungen ergänzen und das numerische Simulationsmodell dahin gehend erweitern. Die Messungen könnten auf weitere relevante Messgrößen ausgeweitet werden, wie den Füllstand des Lagerungssystems und dessen Temperatur. Dies ermöglicht, akkuratere Randbedingungen der Luftfeuchte und des emittierten Methans zu implementieren oder diese als Funktion der lokalen Temperaturen in die Simulation einzubinden. Eine numerisch aufwändigere Weiterführung wäre die detaillierte geometrische Berücksichtigung des Lagerungssystems in der Simulation. Der hierbei entstehende zusätzliche numerische Aufwand könnte durch die Approximation der Kuhherde über ein poröses Material reduziert werden. Hierbei gilt es zu berücksichtigen, dass die Möglichkeiten durch die jeweilige Software limitiert sein können und der Übergang zwischen dem porösen Material und der freien Strömung u. U. eigens formuliert werden müsste. Auch das Verhalten der Wickellüftung könnte über ein poröses Material realitätsnaher simuliert werden. Das poröse Material würde allerdings die in dieser Arbeit lokal aufgeprägten Wärmerandbedingungen der Kühe konterkarieren. Der daraus resultierende geringere numerische

Aufwand könnte aber die Möglichkeit eröffnen, die Wärmestrahlung und -transmission zu ergänzen, indem die Materialparameter des Gebäudes ermittelt und entsprechende Randbedingungen in der Simulation angelegt würden. Diese Erweiterungen könnten insgesamt die Möglichkeit bieten, den frei belüfteten Kuhstall detaillierter zu simulieren oder den numerischen Aufwand zu reduzieren.

Die entwickelte Methode für Physik-informierte neuronale Netze führte bereits zu validen Ergebnissen bei der Approximation der Reynolds-gemittelten Navier-Stokes-Gleichungen für hohe Reynolds-Zahlen. Zukünftige Arbeiten könnten daher zum einen die Dreidimensionalität der Strömung, und zum anderen die Optimierung der Modellgenauigkeit fokussieren. Die zusätzliche dritte kartesische Dimension erfordert eine Anpassung der zusammengesetzten Verlustfunktion, indem die dritte Impulsgleichung ergänzt und die Massenerhaltung angepasst wird. Dem daraus resultierenden größeren Trainingsdatensatz und der einhergehend zusätzlichen Hardwareanforderung könnte über spezielle parallelisiert geschaltete Grafikkarten begegnet werden. Für eine Verbesserung der Modellgenauigkeit könnten zukünftige Arbeiten unterschiedliche Aspekte untersuchen. Eine Möglichkeit könnte die Anpassung der inhärenten Eigenschaften des neuronalen Netzes bieten, indem z. B. die Netztopologie als U-Netz oder Fourier-*Feature*-Netz angelegt würde. Andere Arbeiten zeigen bereits auf, dass die Genauigkeit der Vorhersage durch das neuronale Netz über einen angepassten Trainingsprozess erhöht werden kann. Das *residual-based adaptive refinement* prüft während des Trainings auf Punkte mit hohen Residuen, um in der Region Trainingspunkte zu ergänzen. Eine weitere Methode passt die Gewichtungen der zusammengesetzten Verlustfunktion auf Basis eines Gaußschen Wahrscheinlichkeitsmodells an. Beide Methoden können auf die hier gezeigte Vorgehensweise angewendet werden. Über den Anwendungsfall des frei belüfteten Kuhstalls hinaus könnten diese Optimierungen auf die Approximation Reynolds-gemittelter Navier-Stokes-Gleichungen im Allgemeinen angewendet werden.

Im Rahmen der vorliegenden Arbeit konnte ein überwachender digitaler Zwilling für frei belüftete Kuhställe entwickelt und erprobt werden. Für die Realisierung eines vorausschauenden digitalen Zwillings wäre der nächste Entwicklungsschritt die Einbindung der Wettervorhersage in die Datenbank, um den zukünftigen Zustand der Strömung zu prognostizieren. Ein präskriptiver digitaler Zwilling wäre durch eine automatisiert abgeleitete Handlungsempfehlung möglich, wofür es einer automatisierten Analyse der Strömung in Echtzeit bedürfte. Ein erster Ansatz hierfür wäre ein Ampelsystem, auf Basis der vom Physik-informierten neuronalen Netz vorhergesagten Strömungsgeschwindigkeiten im Gehege. Um die Ergebnisse der Arbeit in die industrielle Anwendung zu übertragen, wäre es erforderlich, einen Vulnerabilitätsscan durchzuführen, um Schwachstellen der IT-Sicherheit im System auszuschließen.

Die aufgeführten Aspekte für zukünftige Arbeiten eröffnen neue Möglichkeiten, die Leistungsfähigkeit der entwickelten Methoden weiter zu verbessern. Durch die fortgesetzte Forschung in diesen Bereichen können das Tierwohl, die Emissionen und der Energiebedarf in landwirtschaftlichen Betrieben effektiver optimiert und weitere Anwendungsbereiche erschlossen werden.

Ergänzende Auswertung der Ultraschallanemometer

Tabelle A.1 stellt alle mittleren Strömungsgeschwindigkeiten, die Turbulenzintensität $I_v = \frac{\sqrt{\frac{1}{3}(\sigma_u^2 + \sigma_v^2 + \sigma_w^2)}}{U}$ und turbulente kinetische Energie $k = \frac{1}{2}(\overline{u'^2} + \overline{v'^2} + \overline{w'^2})$ für alle Sensorpositionen und Messkampagnen dar. Diese Methodik zur Auswertung von Ultraschallanemometern wird ebenfalls in der Arbeit von FIEDLER et al. (2013) dokumentiert. Die Berechnung der Turbulenzgrößen wurde mittels des gesamten Roh-Datensatzes durchgeführt. Die mittleren Geschwindigkeitskomponenten sind in das lokale Koordinatensystem des Stalls transformiert, also entsprechend der Ausrichtung des Versuchsstalls um -15° rotiert.

Tabelle A.1: Mittlere Geschwindigkeitskomponenten im lokalen Koordinatensystem des Stalls, turbulente kinetische Energie und Turbulenzintensität

Kampagne	Quantität	Süd-Ost	Süd	Süd-West	Mitte	Nord-Ost	Nord	Nord-West
①	\bar{u} in m/s	0,378	0,412	0,494	0,515	1,043	1,808	0,931
	\bar{v} in m/s	-0,281	-0,454	-0,243	-0,250	-0,653	-0,591	-1,113
	\bar{w} in m/s	0,169	0,053	0,049	0,234	0,183	0,210	0,210
	k in m^2/s^2	0,248	0,217	0,256	0,368	0,936	0,916	1,589
	I_v	0,813	0,618	0,748	0,800	0,635	0,408	0,702
②	\bar{u} in m/s	-1,823	-2,251	-2,472	-0,588	-0,206	-0,413	-0,682
	\bar{v} in m/s	0,066	-0,168	0,029	-0,085	0,253	0,343	0,445
	\bar{w} in m/s	-0,202	0,185	0,275	0,038	-0,092	-0,054	0,079
	k in m^2/s^2	0,523	0,754	0,925	0,131	0,056	0,065	0,193
	I_v	0,322	0,313	0,316	0,496	0,569	0,387	0,438
③	\bar{u} in m/s	-1,508	-1,901	-2,305	-0,097	0,022	-0,149	-0,907
	\bar{v} in m/s	0,131	0,091	-0,137	0,313	0,182	0,299	-0,008
	\bar{w} in m/s	0,208	0,507	0,342	0,041	0,217	0,159	0,274
	k in m^2/s^2	0,407	0,310	0,555	0,104	0,198	0,244	0,588
	I_v	0,341	0,231	0,261	0,795	1,274	1,090	0,661

ANHANG **B**

Sitzungsprotokoll

Im Folgenden befindet sich das Sitzungsprotokoll eines online stattgefundenen Projekttreffens mit Mitarbeitern der *HUESKER Synthetic GmbH*, am Freitag, dem 07.10.2022 von 10:00 Uhr bis 12:00 Uhr.

Tagesordnungspunkte:

TOP1 Vorstellung aktueller Ergebnisse

TOP2 Besprechung der Grenzwerte des Versuchsplans

TOP3 Klärung fehlender Informationen zur Berechnung des Methans

Ergebnisse TOP2

Tabelle B.1: Parameterliste mit den Grenzwerten für die statistische Versuchsplanung

Parameter	+	–	Einheit	Anmerkung
Lufttemperatur	23,5	–0,5	°C	Aus statistischer Auswertung der Wetterstation
Luftfeuchte	100	57	%	Aus statistischer Auswertung der Wetterstation
Windgeschwindigkeit	6,5	0,5	$\frac{m}{s}$	Aus statistischer Auswertung der Wetterstation
Windrichtung	Nord	West		Bester und schlechtester anzunehmender Fall
Entfernung Windschattengebäude	n. v.	15	m	Bester und schlechtester anzunehmender Fall
Axialventilatoren	35.000	0	$\frac{m^3}{h}$	Volllast und ausgestellt
Deckenventilator	500.000	0	$\frac{m^3}{h}$	Volllast und ausgestellt
Schlauchlüftung	200	0	$\frac{m^3}{h \cdot m}$	Volllast und ausgestellt
Traufhöhe	5,0	2,7	m	Typische hohe und niedrige Werte
Dachneigung	25,0	15,0	°	Typische hohe und niedrige Werte
Länge	120,0	30,0	m	Typische hohe und niedrige Werte
Breite	60,0	24,0	m	Typische hohe und niedrige Werte
Wickellüftung oben	90	0	%	Berücksichtigung baubedingter Resthöhe
Wickellüftung unten	90	0	%	Berücksichtigung baubedingter Resthöhe

Ergebnisse TOP3

Durchschnittliche Lebendmasse je Tier der Herde: 670 kg ist ein guter Durchschnittswert für gemolkene Kühe, eine junge frisch abgekalbte Kuh startet mit 625 kg, aber die Herde schwankt dann mit 650 bis 700 kg.

Durchschnittliche Milchproduktion je Tag: 30 kg/(Kuh·Tag). Man findet häufig eine Jahresleistung von 10.000 kg Herdenschnitt, wobei dies von 8.000 bis 13.000 kg schwanken kann.

Fettanteil der Milch: 4,2 % ist ein gutes Mittel.

Wie viele der Kühe in der Herde gebären je Jahr: Man spricht von einer Zwischenkalbezeit von 390 Tagen und kommt somit auf etwa 93 % der Herde.

Welche Futterart wird verwendet: Milchkühe bekommen 55 bis 70 % Raufutter und 30 bis 45 % Kraftfutter, je nach Milchleistung und Laktationsphase.

Bilder des Versuchsstalls

Abbildung C.1 stellt die Außenansichten auf den Versuchsstall dar. In der Luftaufnahme (Markierung 1) ist das südlich des Stalls gelegene Silo erkennbar. Die Luftaufnahme zeigt auch, dass der auf der Betonplatte positionierte Stall durch Grasland umgeben ist. Das in Markierung 2 abgebildete Tor ist der westliche Zugang zur außen liegenden Futterplatte auf der Südseite des Gebäudes. Ergänzend ist die zu etwa 75 % geöffnete Wickellüftung zu erkennen. In der rechtsseitigen Abbildung ist die östliche Stirnseite des Stalls zu sehen und die lokale Wetterstation markiert.



①



②



③

Abbildung C.1: Außenansichten des Versuchsstalls als Luftaufnahme (1), Ansicht der Tore für den Zugang zur Futterplatte (2) und die Ostseite des Gebäudes mit Blick auf die Wetterstation (3), mit freundlicher Genehmigung der Firma WELSING MILCHVIEH

Abbildung C.2 zeigt in der linken Darstellung den Blick aus dem Büro auf das Gehege im Stall (Markierung 1), worin die in den Liegeboxen befindlichen Kühe, die Gerüststrukturen sowie der Spaltenboden zu erkennen sind. Markierung 2 zeigt den Stall von der nördlichen Futterplatte aus mit Blick auf das westliche Tor, wohingegen in Markierung 3 der Blick nach Osten gerichtet ist.

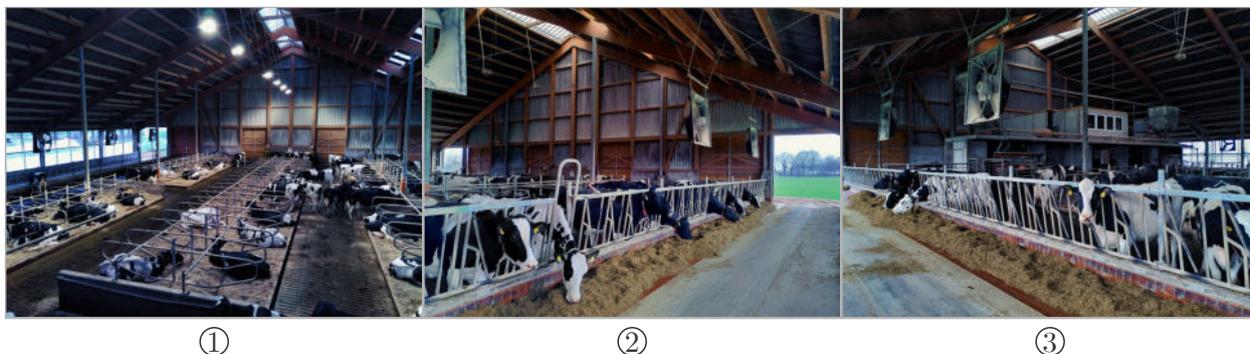


Abbildung C.2: Innenansichten des Versuchsstalls mit dem Blick vom Büro (1), der nördlichen Futterplatte in Richtung Westen (2) und in Richtung Osten (3), mit freundlicher Genehmigung der Firma WELSING MILCHVIEH

In Abbildung C.3 ist eine Aufnahme des für die Messkampagnen eingerichteten mobilen Arbeitsplatzes zu sehen. Markierung 1 kennzeichnet den tragbaren Rechner, auf dem die entwickelte Webapplikation in einer offline Version angelegt war und so die Sensordaten zentral in der Datenbank erfasst und beobachtet wurden. An der Position der Markierung 2 ist die Auswerteeinheit des 3D Ultraschallanemometers platziert. Ergänzend wurde ein Entfernungsmesser für die Erfassung der Sensorpositionen mitgeführt (Markierung 3).



Abbildung C.3: Mobiler Arbeitsplatz im Versuchsstall, mit dem Rechner (1), der Auswerteeinheit des USA (2) und dem Entfernungsmesser (3), mit freundlicher Genehmigung der Firma WELSING MILCHVIEH

Konturdiagramme der numerischen Simulationen

In den folgenden Kapiteln sind die Konturdiagramme jedes Versuchsdurchlaufs der statistischen Versuchsplanung dokumentiert. Dargestellt werden jeweils der Geschwindigkeitsbetrag, die Temperaturverteilung, die Methankonzentration und Luftfeuchte. Für die x - y -Ebene wird jeweils auf der Höhe der Messwerte des 3D Ultraschallanemometers bei $z = 1,72$ m geschnitten. Für den Fall der Einstellung der Windrichtung α beim niedrigen Wert (laterale Einströmbedingung auf die Stirnfläche des Stalls) ist die z - y -Ebene bei $x = 0$ m an der Symmetrieebene des Stalls geschnitten. Bei der hohen Einstellung der Windrichtung (Querbelüftung) wird die Schnittebene x - z etwa mittig in das Gehege gelegt, sodass der Querschnitt möglichst die Umströmung der außen stehenden und innen liegenden Kühe darstellt und den Einfluss der zusätzlichen Lüftungselemente erfasst.

D.1 Konturdiagramme der Durchläufe des *Screening*-Versuchsplans

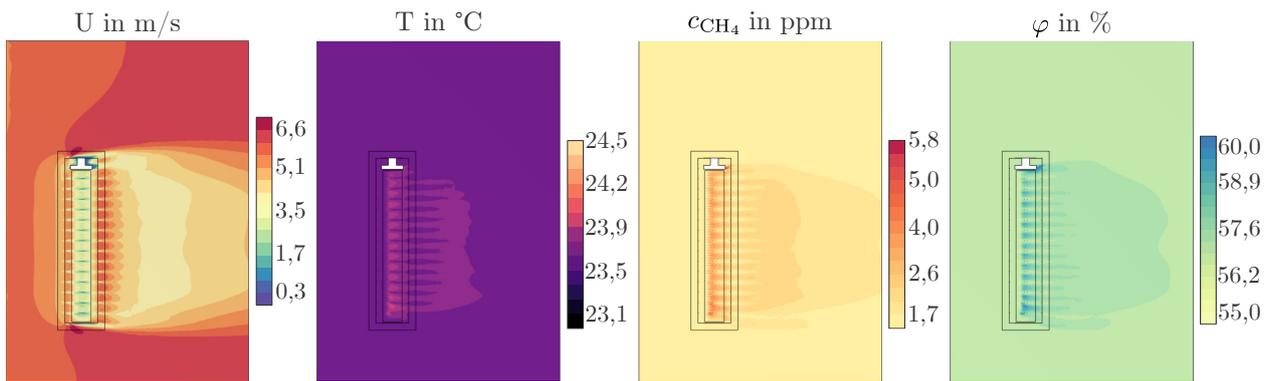


Abbildung D.1: CFD-Ergebnisse des Durchlaufes N1 auf der Schnittebene x - y , bei $z = 1,7$ m (eigene Darstellung)

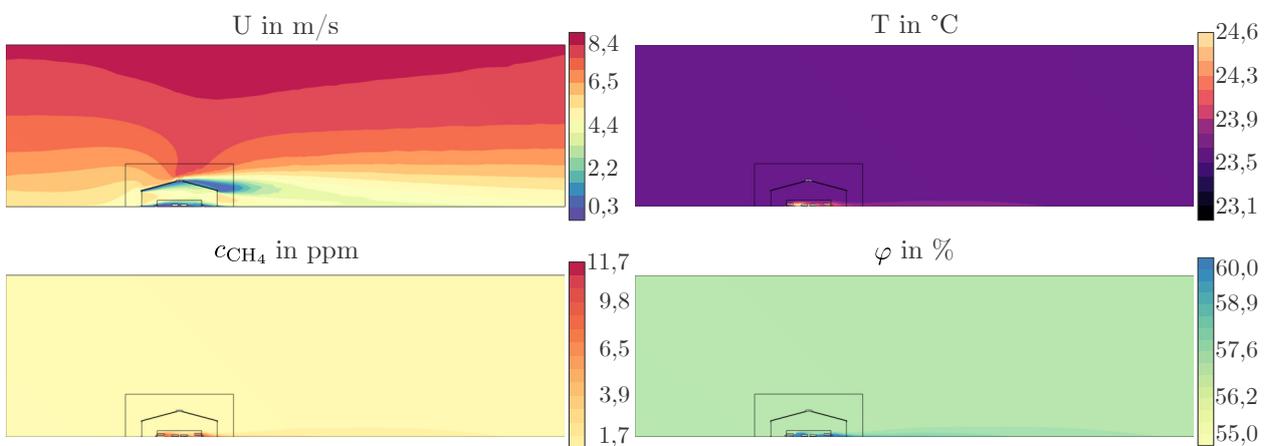


Abbildung D.2: CFD-Ergebnisse des Durchlaufes N1 auf der Schnittebene x - z , bei $y = 53,2$ m (eigene Darstellung)

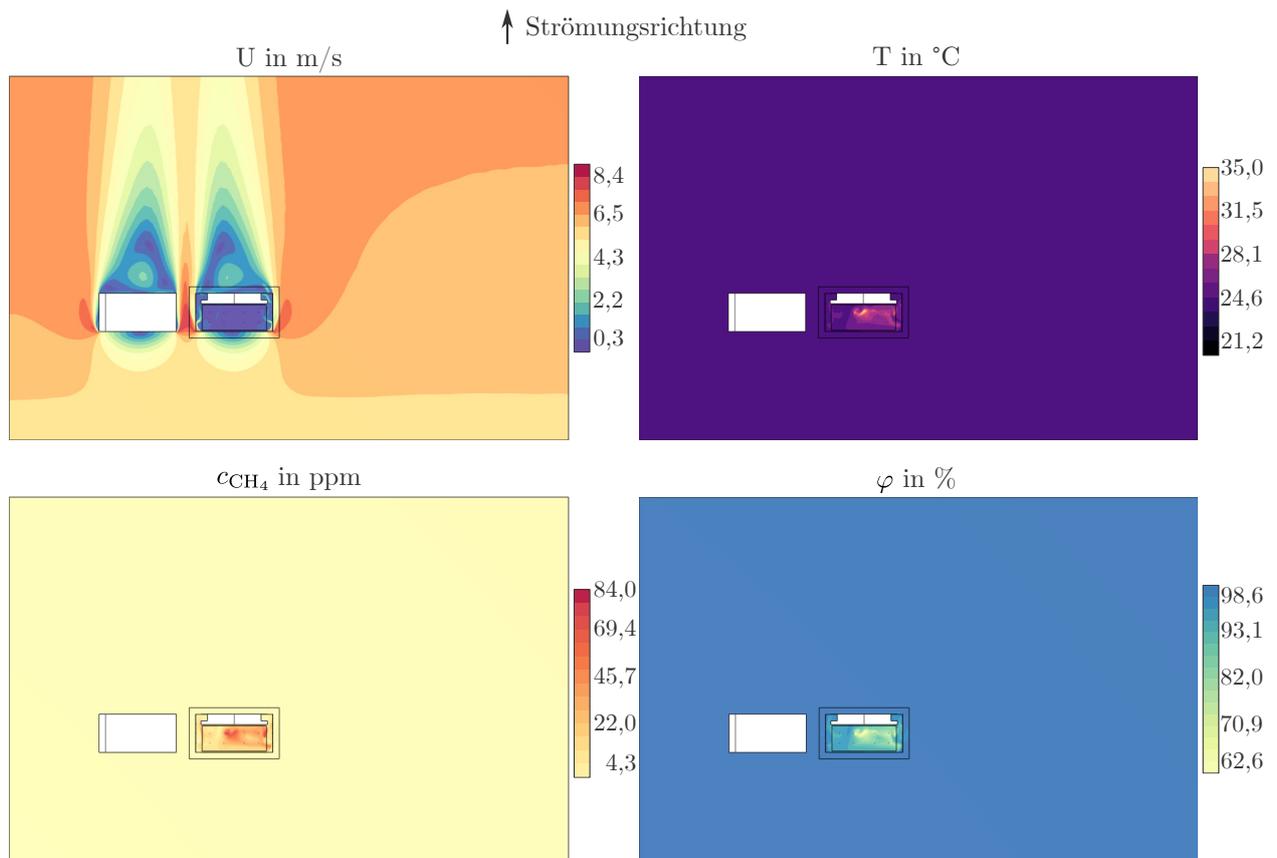


Abbildung D.3: CFD-Ergebnisse des Durchlaufes N₂ auf der Schnittebene x - y , bei $z = 1,7$ m (eigene Darstellung)

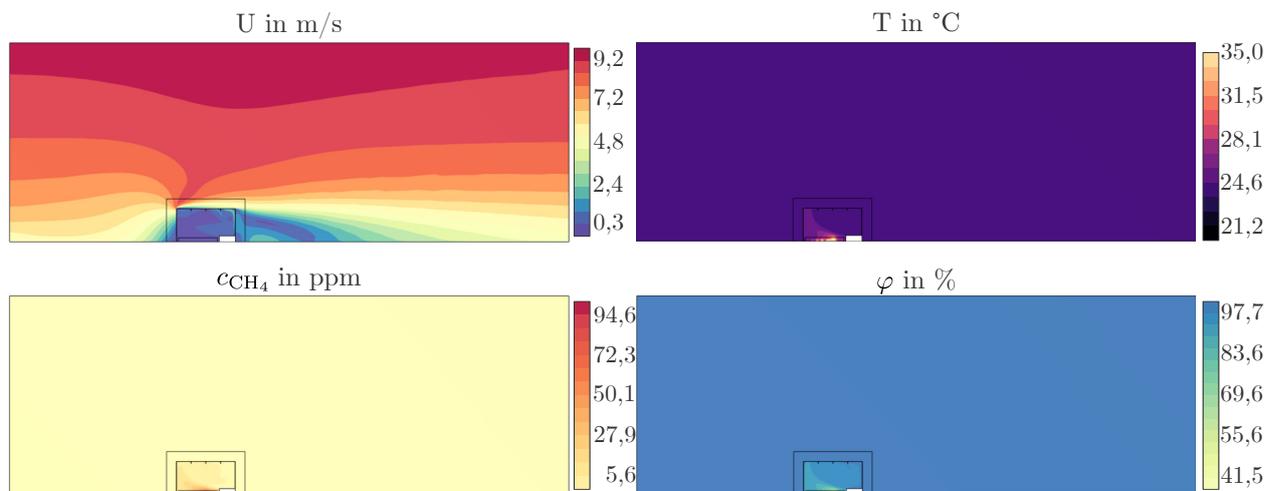


Abbildung D.4: CFD-Ergebnisse des Durchlaufes N₂ auf der Schnittebene y - z , bei $x = 0,0$ m (eigene Darstellung)

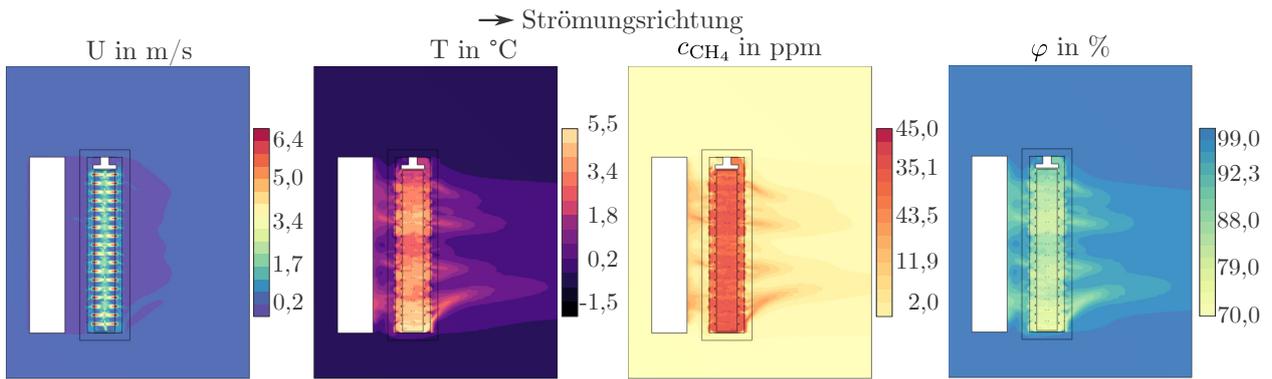


Abbildung D.5: CFD-Ergebnisse des Durchlaufes N3 auf der Schnittebene x - y , bei $z = 1,7$ m (eigene Darstellung)

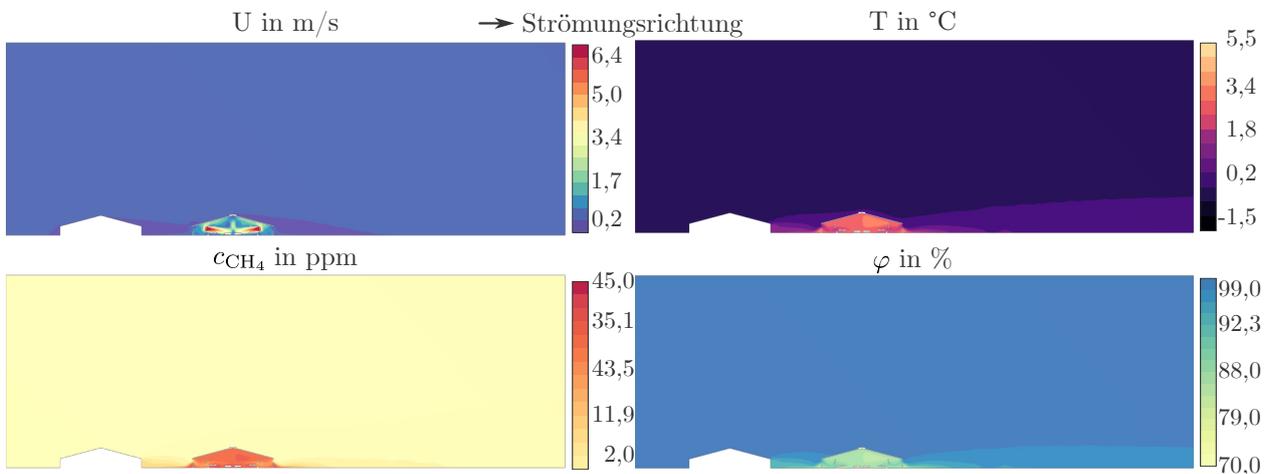


Abbildung D.6: CFD-Ergebnisse des Durchlaufes N3 auf der Schnittebene x - z , bei $y = 66,0$ m (eigene Darstellung)

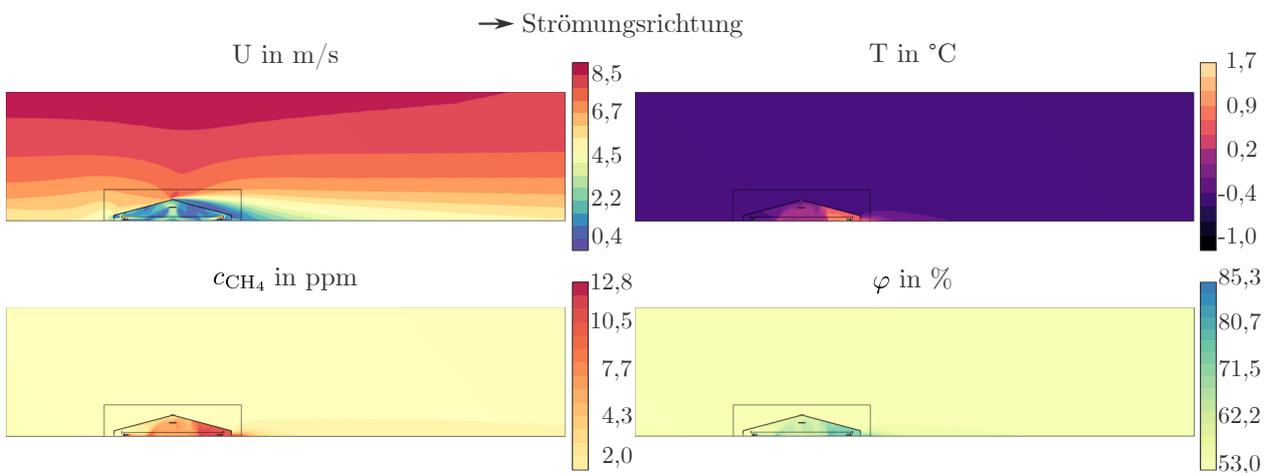


Abbildung D.7: CFD-Ergebnisse des Durchlaufes N4 auf der Schnittebene x - z , bei $y = 12,0$ m (eigene Darstellung)

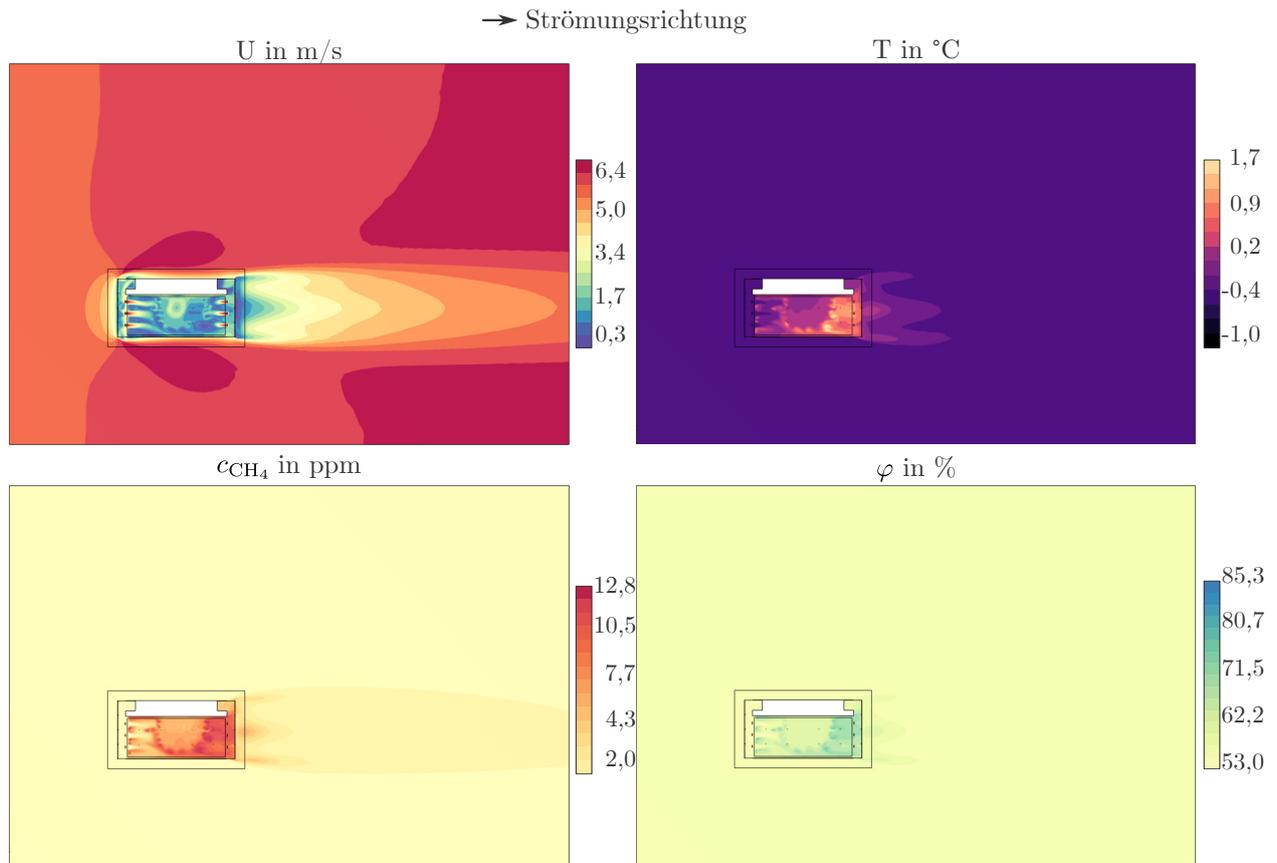


Abbildung D.8: CFD-Ergebnisse des Durchlaufes N4 auf der Schnittebene x - y , bei $z = 1,7$ m (eigene Darstellung)

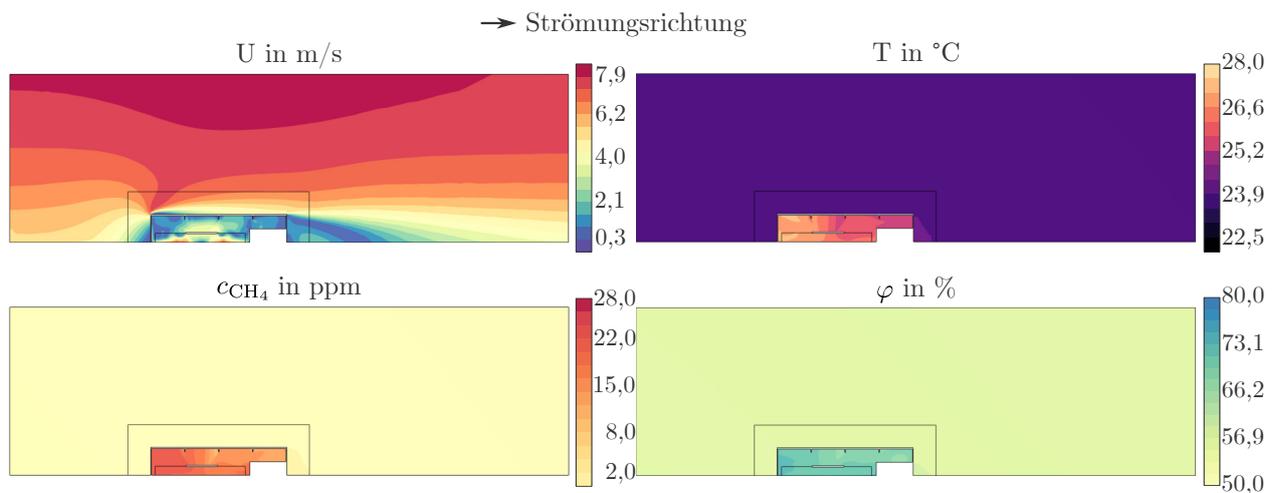


Abbildung D.9: CFD-Ergebnisse des Durchlaufes N5 auf der Schnittebene z - y , bei $x = 0,0$ m (eigene Darstellung)

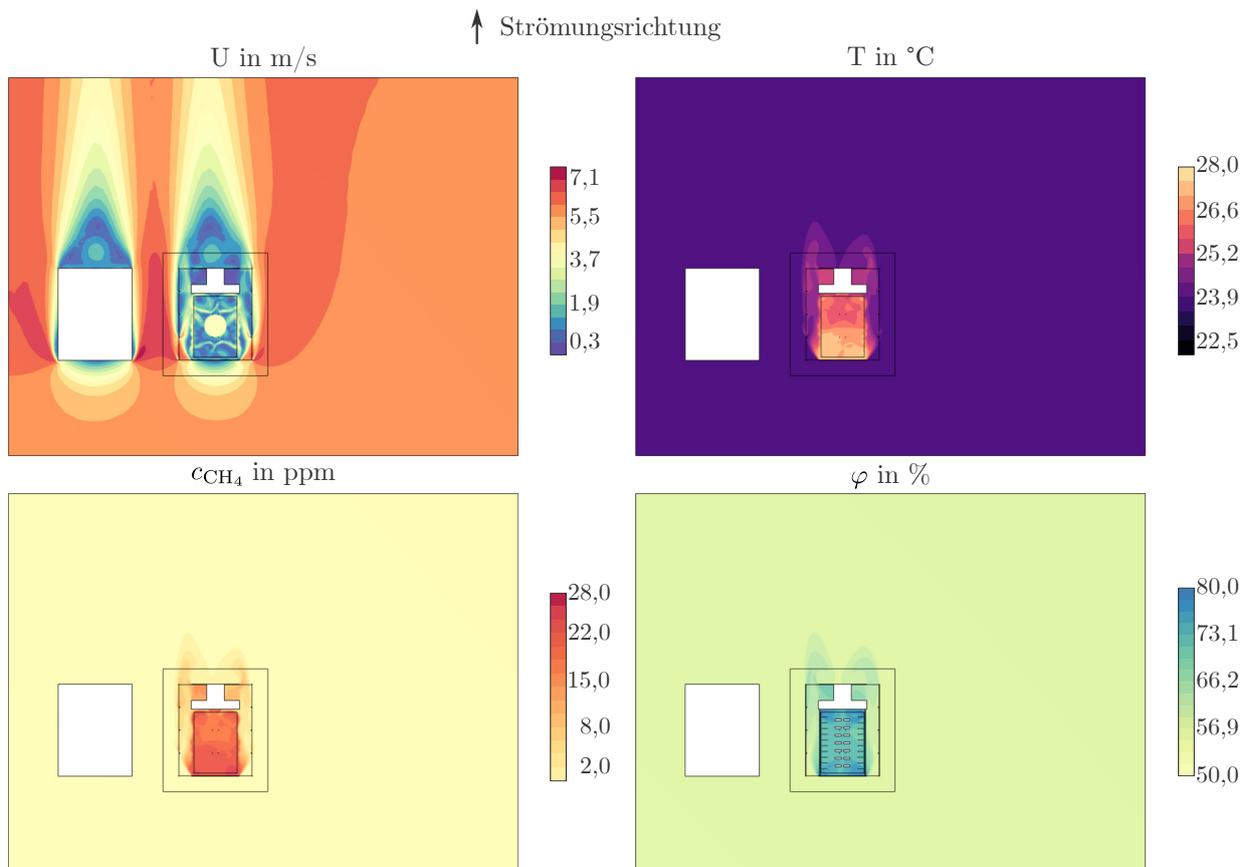


Abbildung D.10: CFD-Ergebnisse des Durchlaufes N5 auf der Schnittebene $x-y$, bei $z = 1,7$ m (eigene Darstellung)

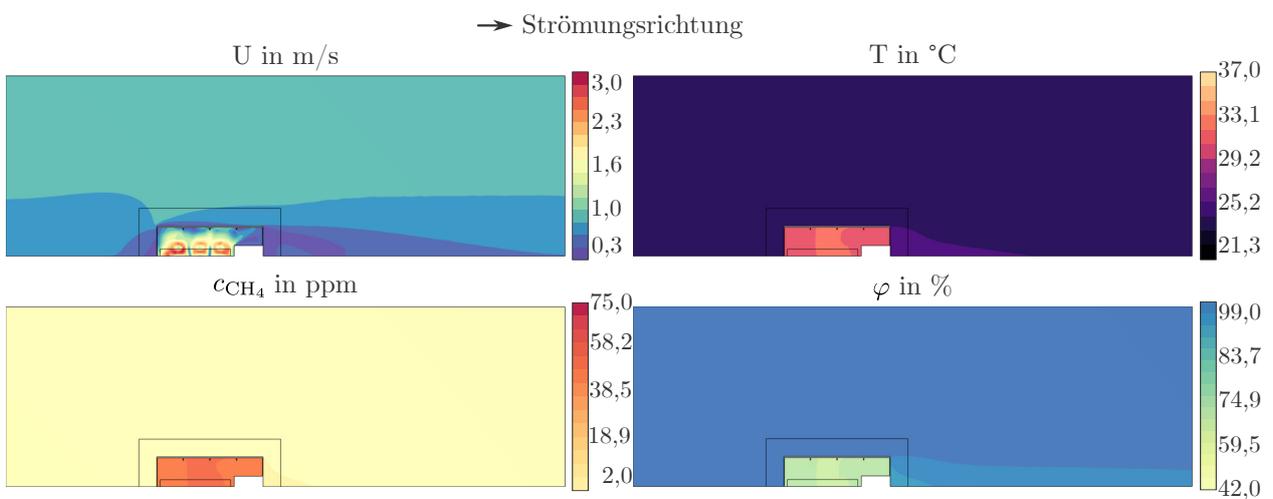


Abbildung D.11: CFD-Ergebnisse des Durchlaufes N6 auf der Schnittebene $z-y$, bei $x = 0,0$ m (eigene Darstellung)

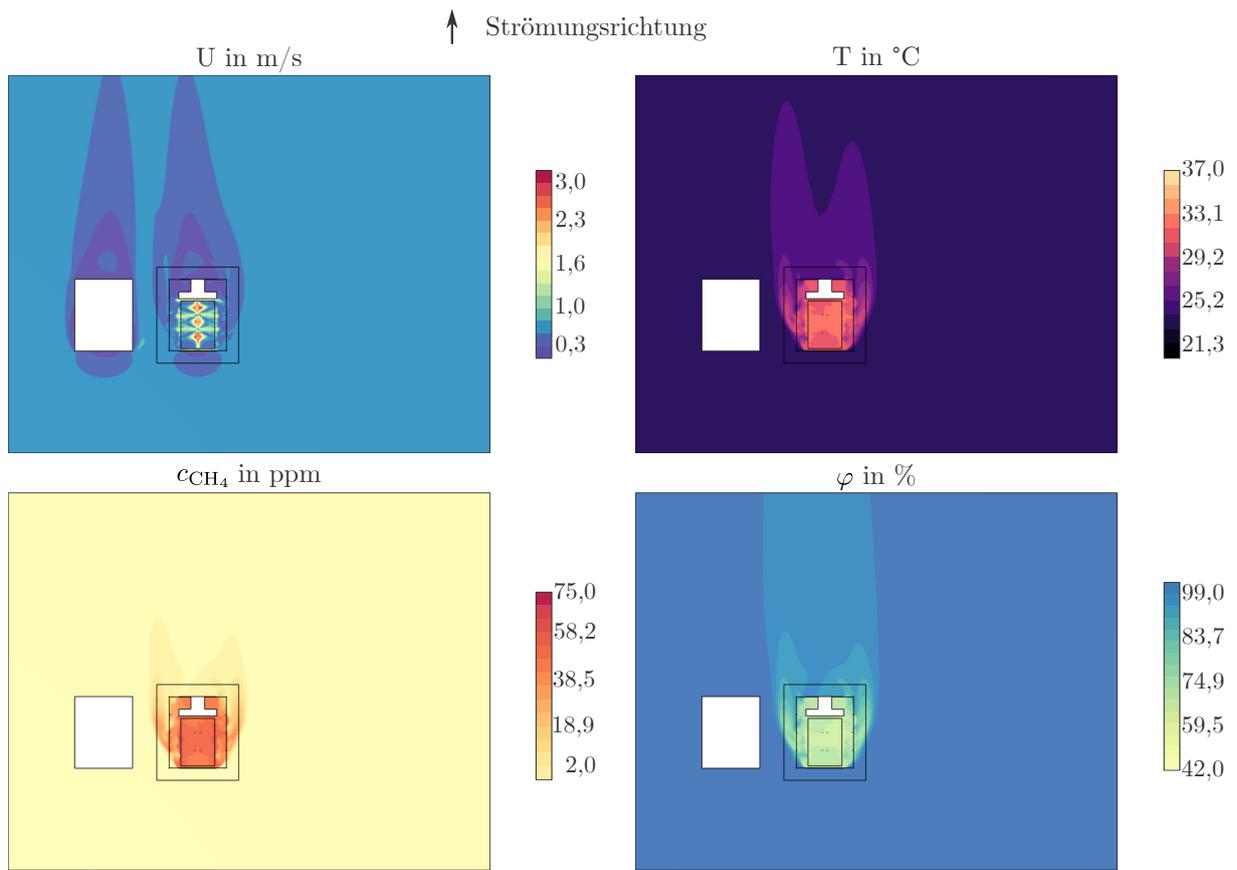


Abbildung D.12: CFD-Ergebnisse des Durchlaufes N6 auf der Schnittebene x - y , bei $z = 1,7$ m (eigene Darstellung)

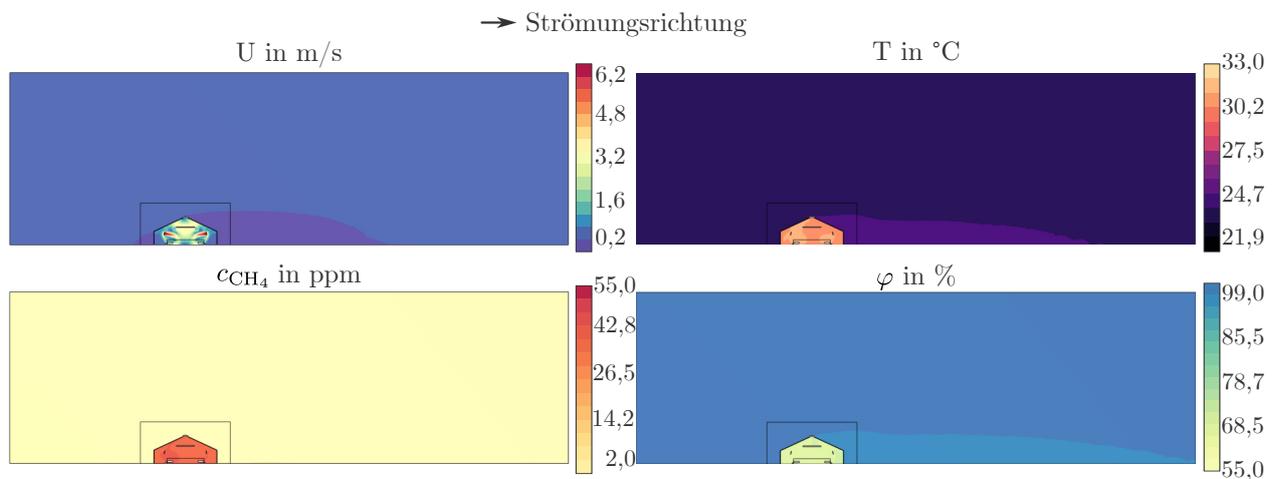


Abbildung D.13: CFD-Ergebnisse des Durchlaufes N7 auf der Schnittebene x - z , bei $y = 11,8$ m (eigene Darstellung)

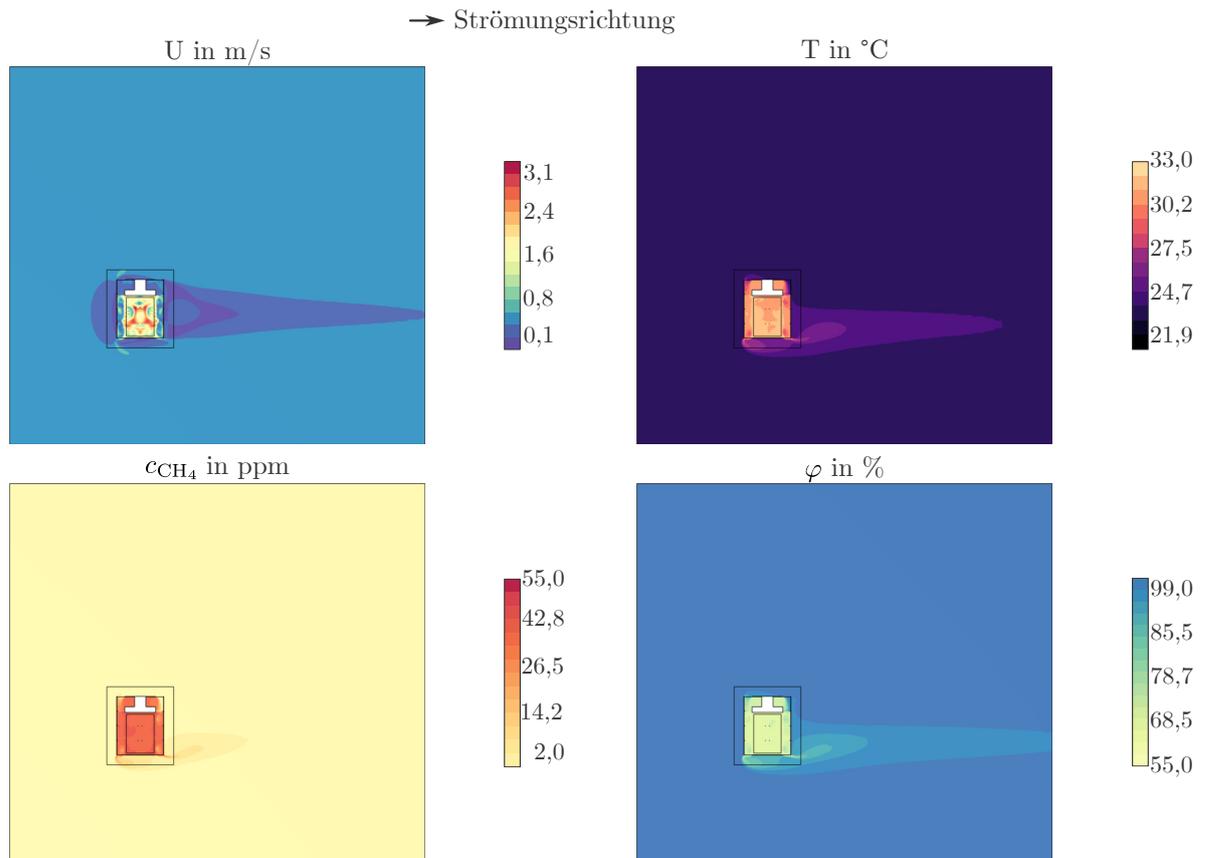


Abbildung D.14: CFD-Ergebnisse des Durchlaufes N7 auf der Schnittebene $x-y$, bei $z = 1,7$ m (eigene Darstellung)

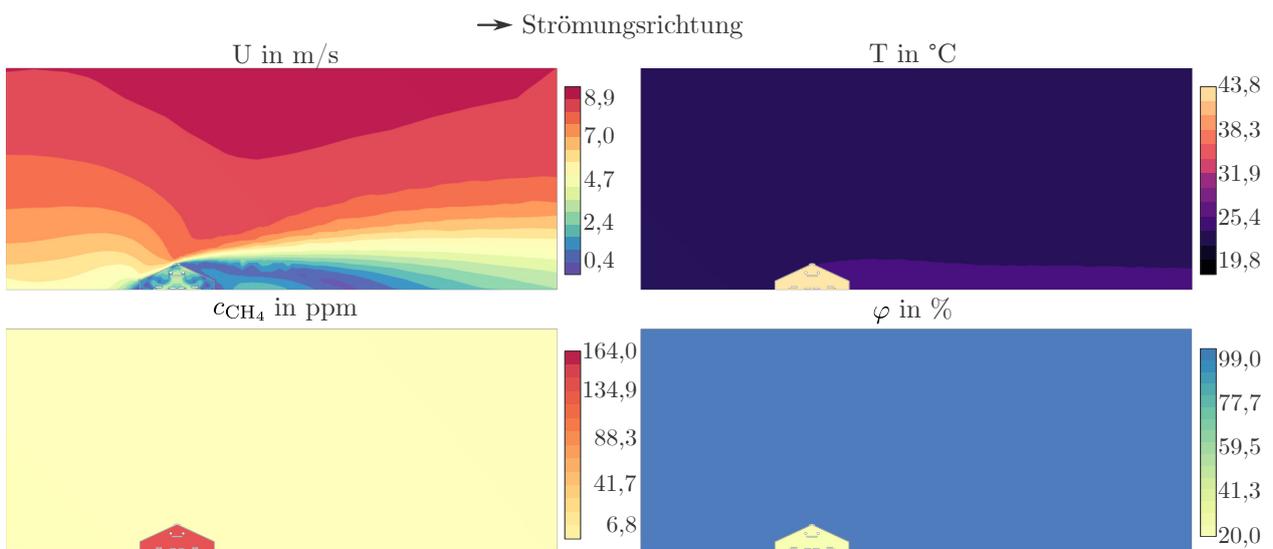


Abbildung D.15: CFD-Ergebnisse des Durchlaufes N8 auf der Schnittebene $x-z$, bei $y = 53,2$ m (eigene Darstellung)

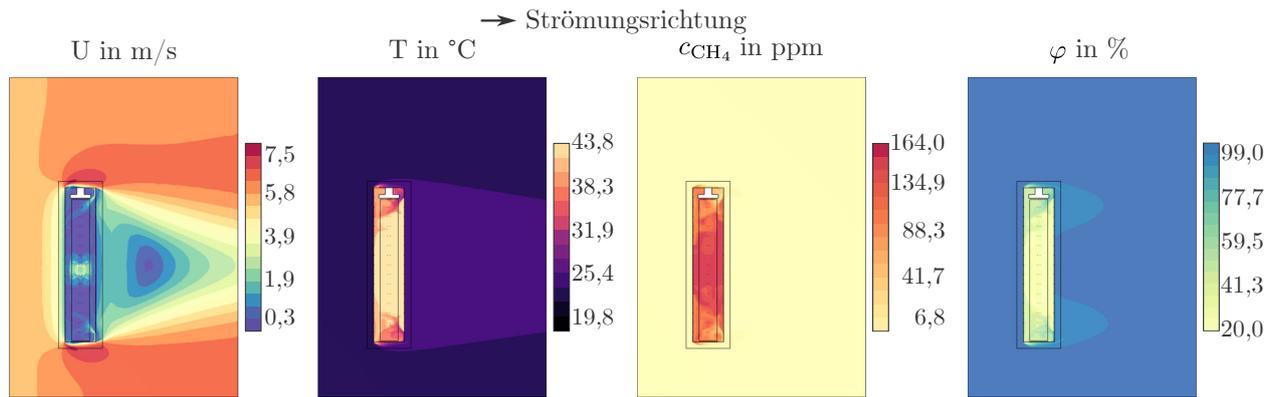


Abbildung D.16: CFD-Ergebnisse des Durchlaufes N8 auf der Schnittebene $x-y$, bei $z = 1,7$ m (eigene Darstellung)

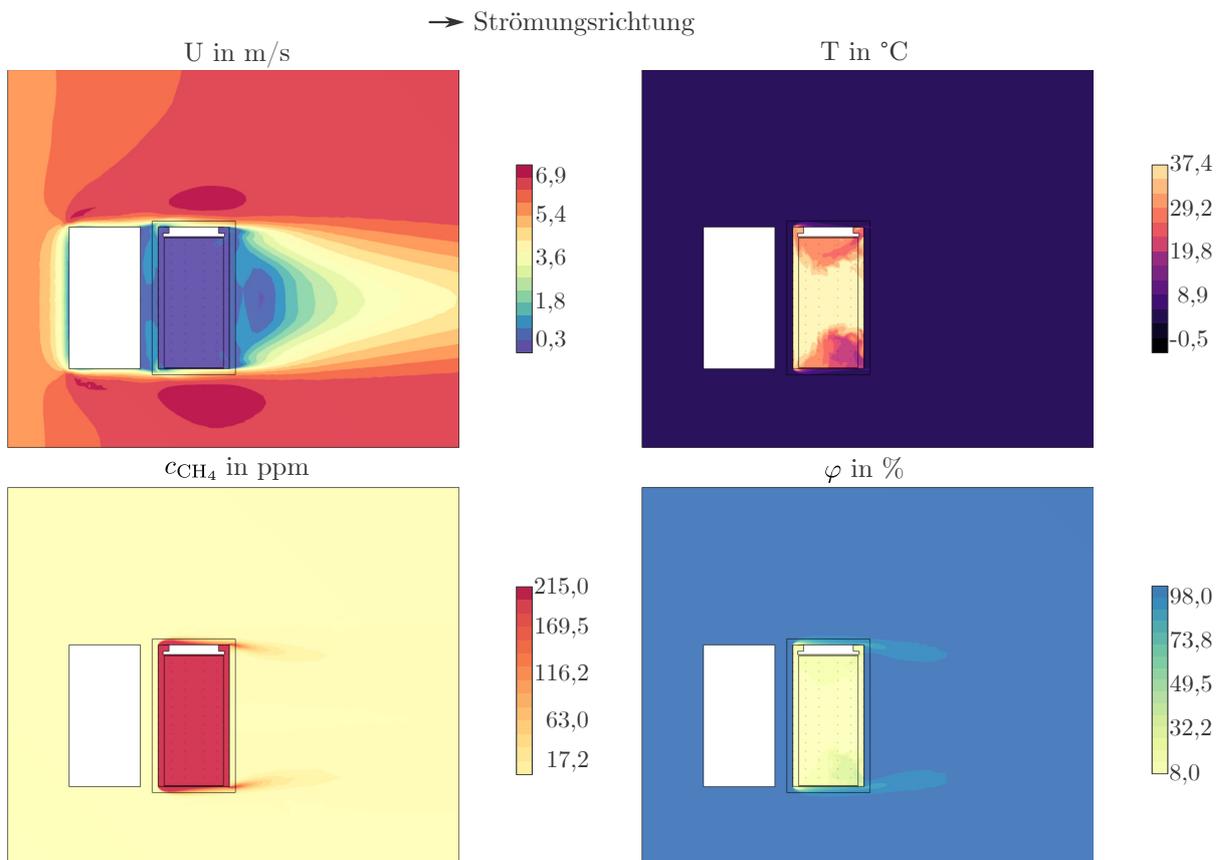


Abbildung D.17: CFD-Ergebnisse des Durchlaufes N9 auf der Schnittebene $x-y$, bei $z = 1,7$ m (eigene Darstellung)

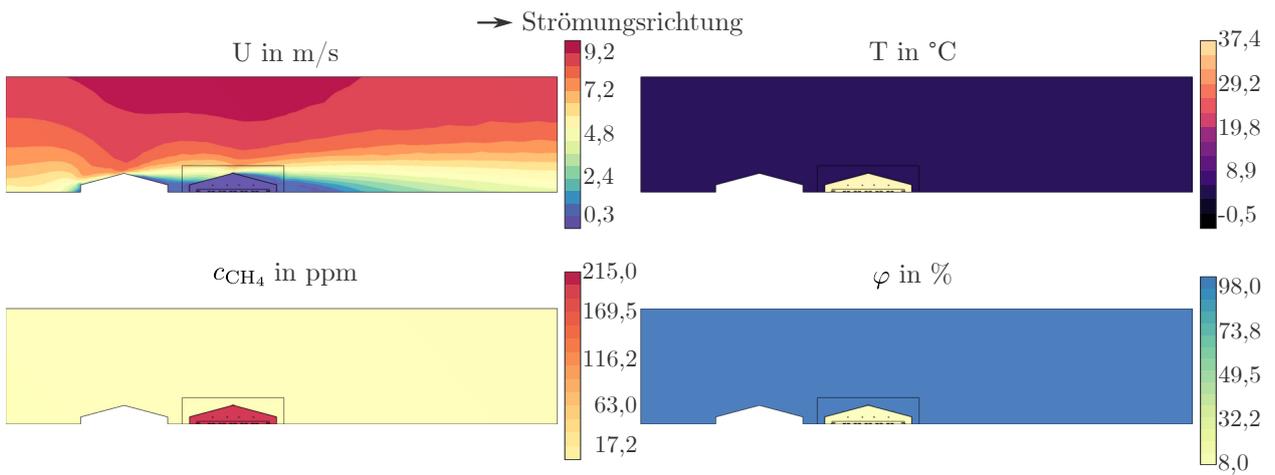


Abbildung D.18: CFD-Ergebnisse des Durchlaufes N9 auf der Schnittebene x - z , bei $y = 54,6$ m (eigene Darstellung)

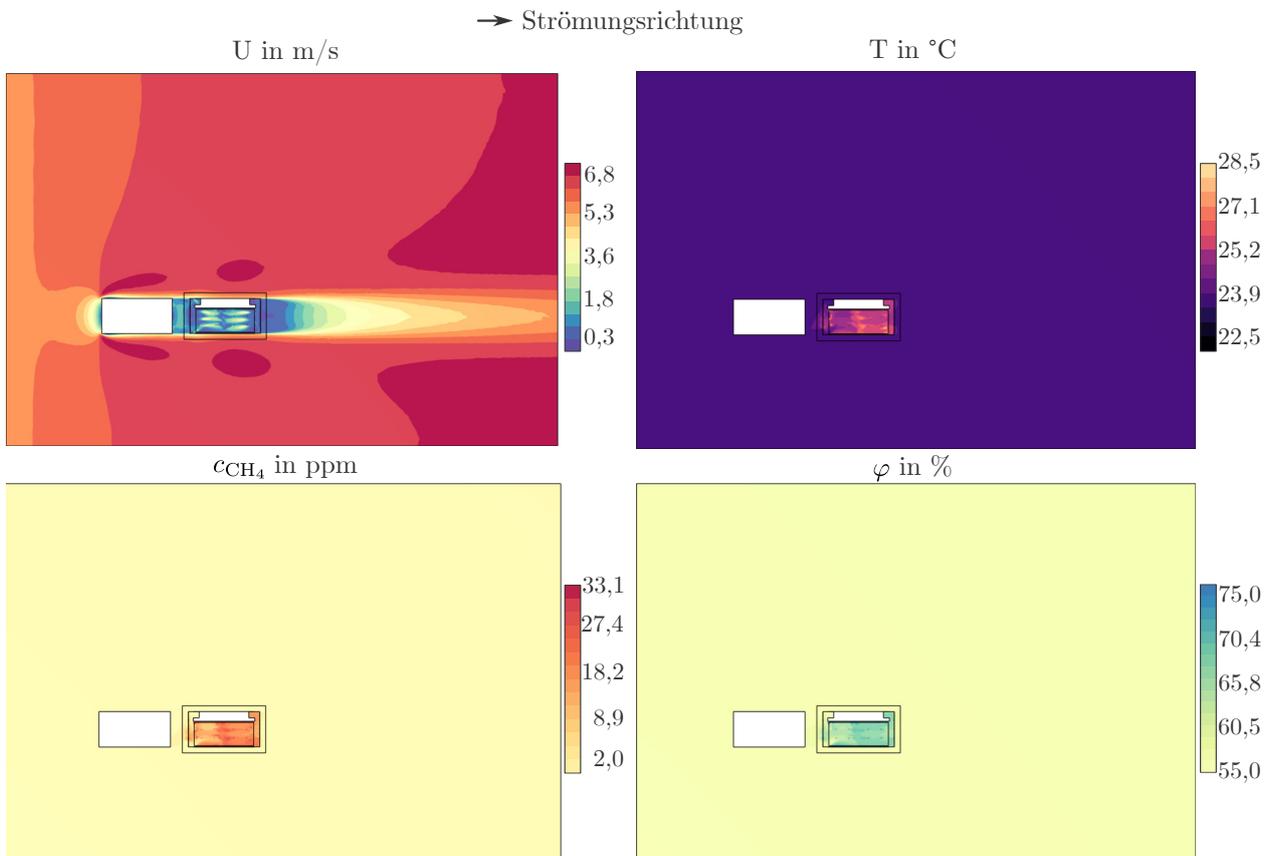


Abbildung D.19: CFD-Ergebnisse des Durchlaufes N10 auf der Schnittebene x - y , bei $z = 1,7$ m (eigene Darstellung)

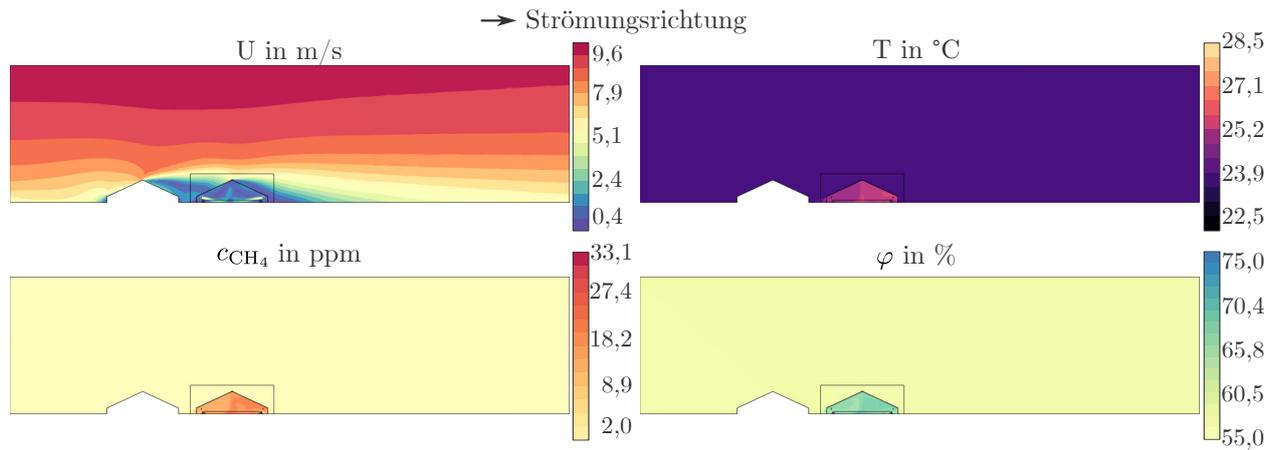


Abbildung D.20: CFD-Ergebnisse des Durchlaufes N10 auf der Schnittebene x - z , bei $y = 12,0$ m (eigene Darstellung)

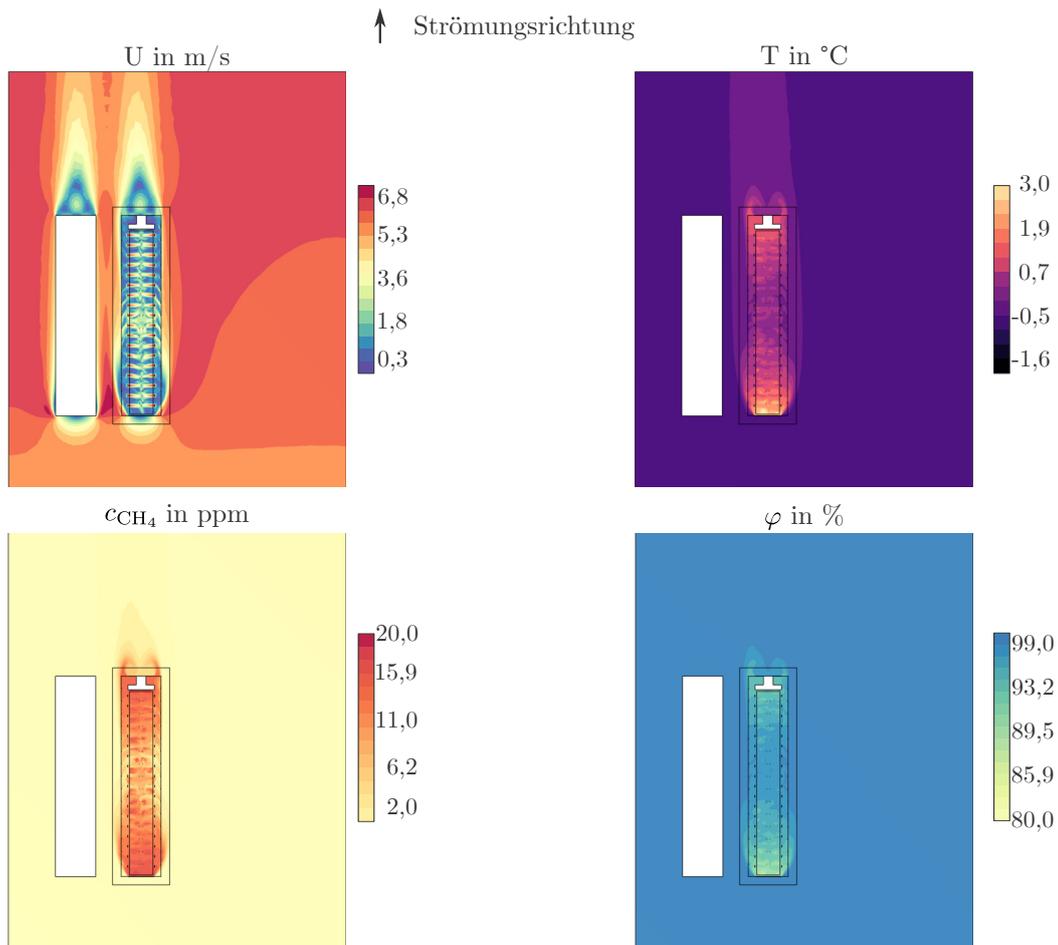


Abbildung D.21: CFD-Ergebnisse des Durchlaufes N11 auf der Schnittebene x - y , bei $z = 1,7$ m (eigene Darstellung)

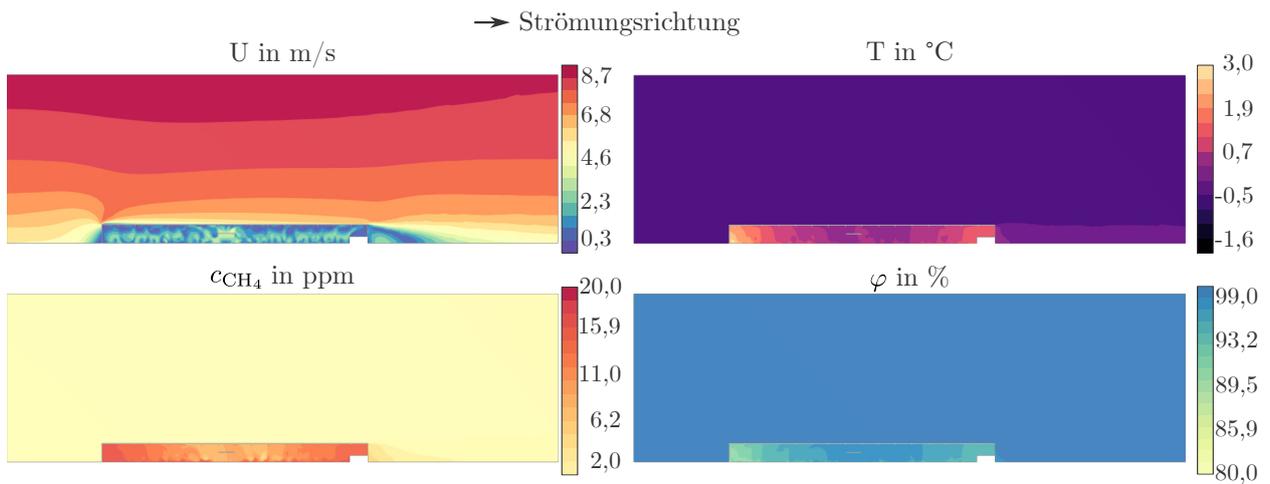


Abbildung D.22: CFD-Ergebnisse des Durchlaufes N11 auf der Schnittebene $y-z$, bei $x = 0,0$ m (eigene Darstellung)

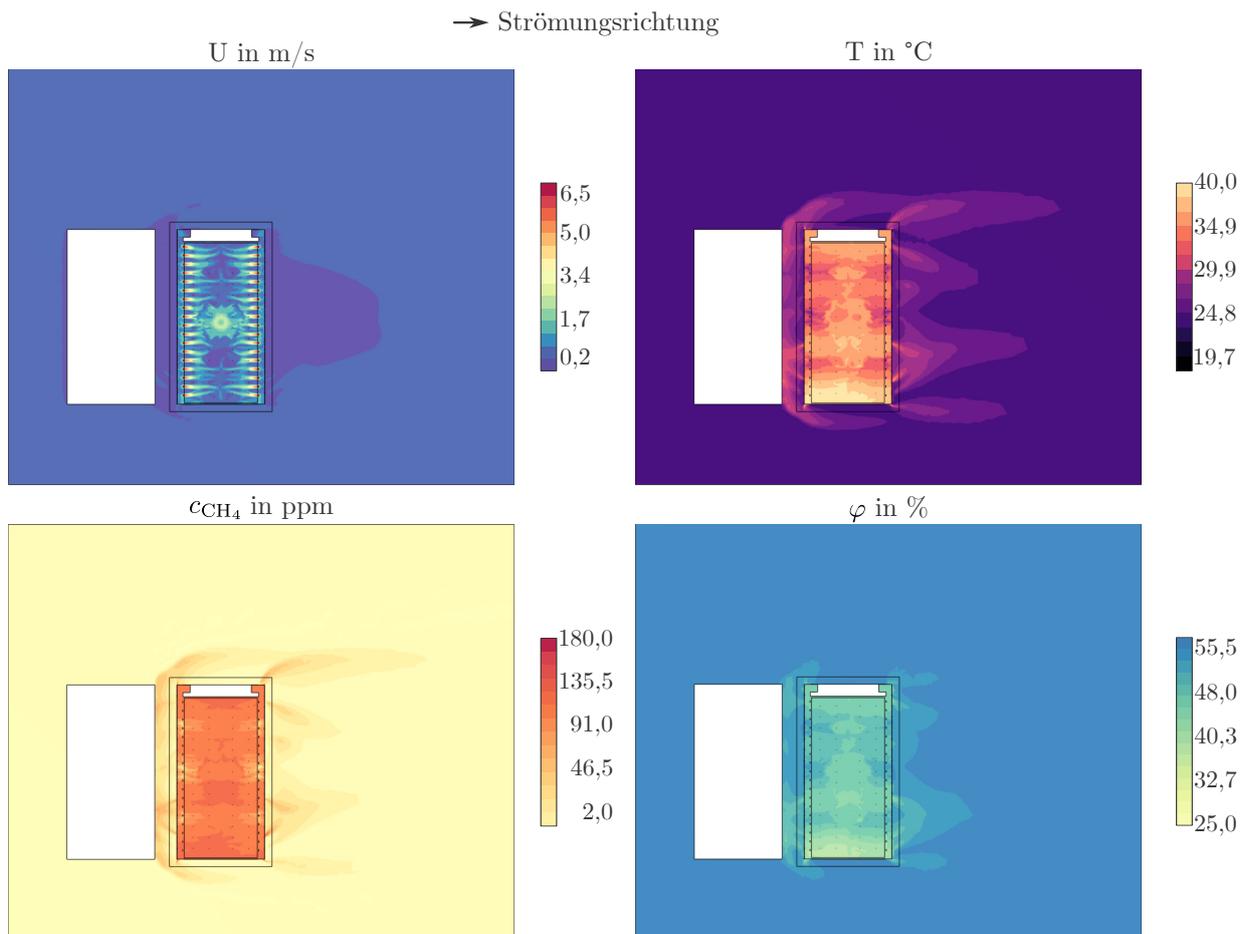


Abbildung D.23: CFD-Ergebnisse des Durchlaufes N12 auf der Schnittebene $x-y$, bei $z = 1,7$ m (eigene Darstellung)

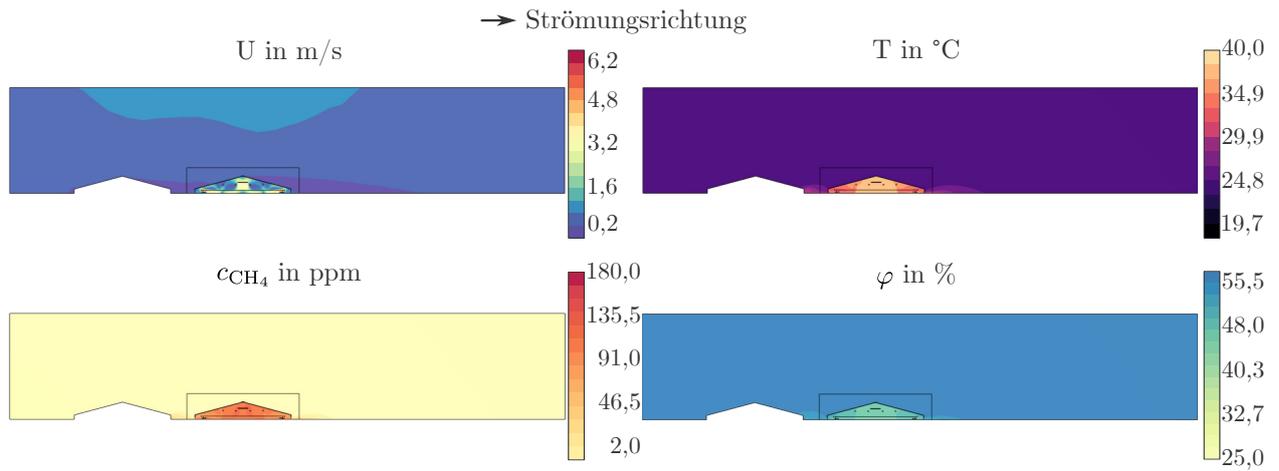


Abbildung D.24: CFD-Ergebnisse des Durchlaufes N12 auf der Schnittebene x - z , bei $y = 54,7$ m (eigene Darstellung)

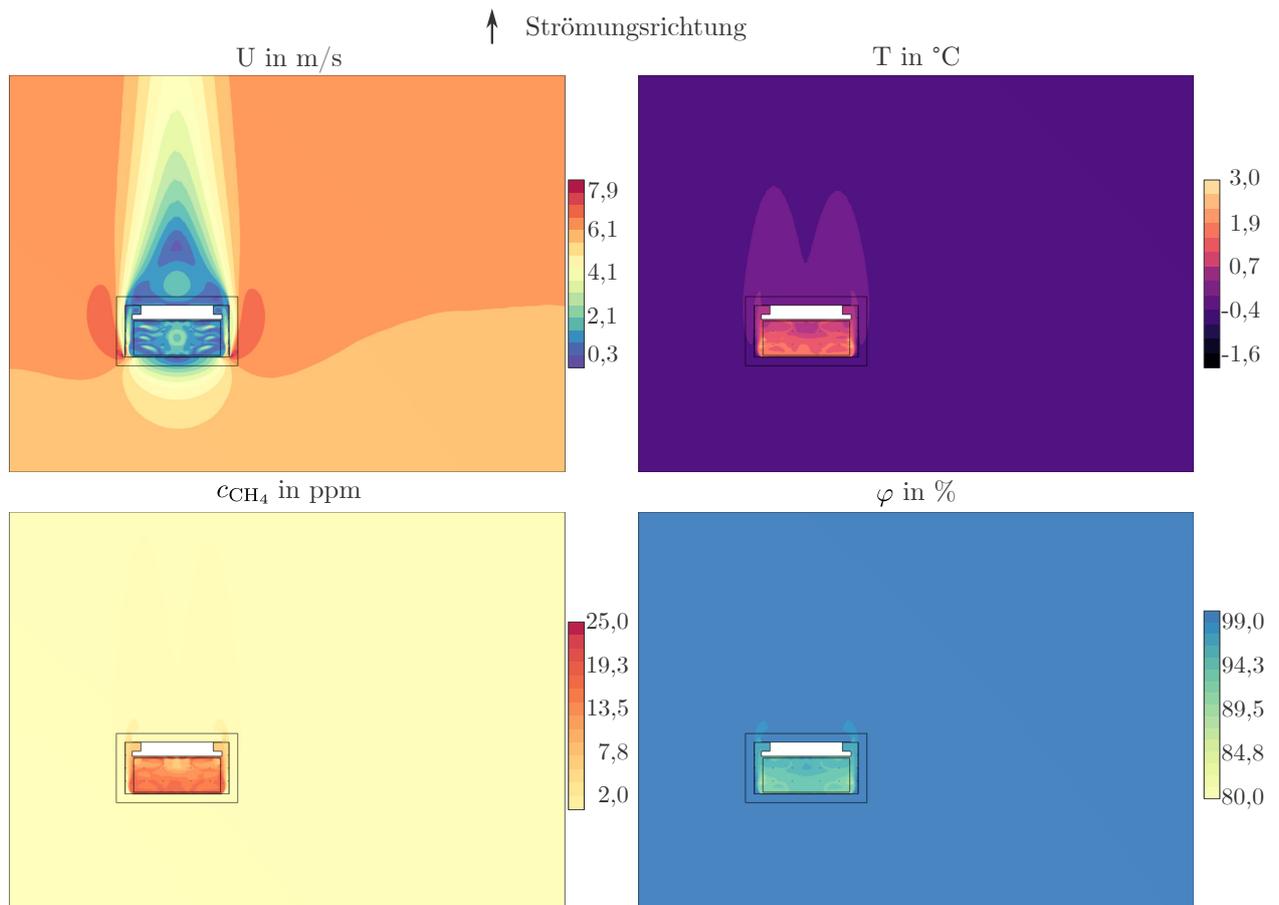


Abbildung D.25: CFD-Ergebnisse des Durchlaufes N13 auf der Schnittebene x - y , bei $z = 1,7$ m (eigene Darstellung)

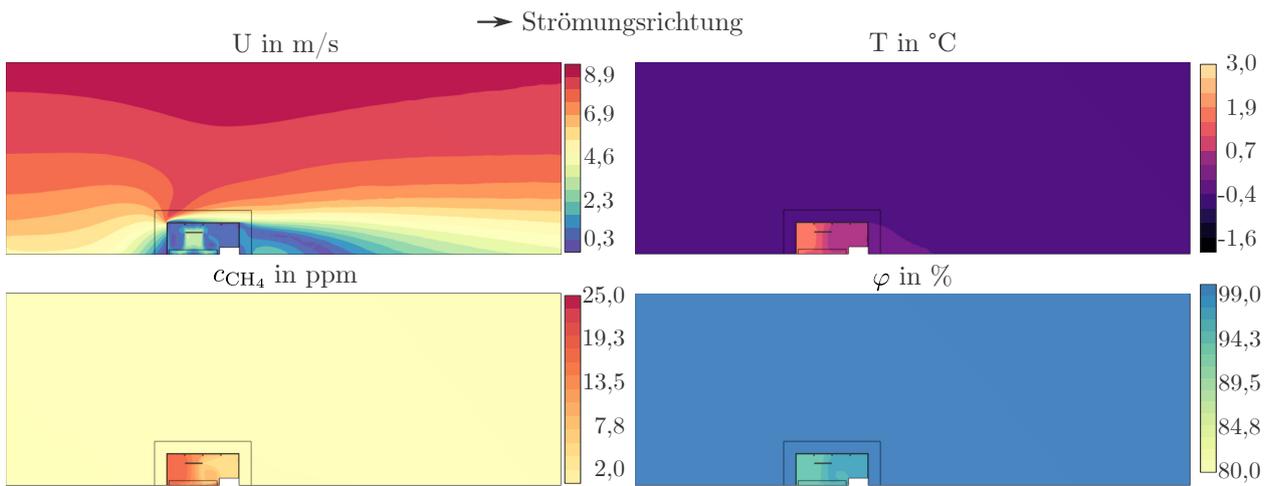


Abbildung D.26: CFD-Ergebnisse des Durchlaufes N13 auf der Schnittebene $y-z$, bei $x = 0,0$ m (eigene Darstellung)

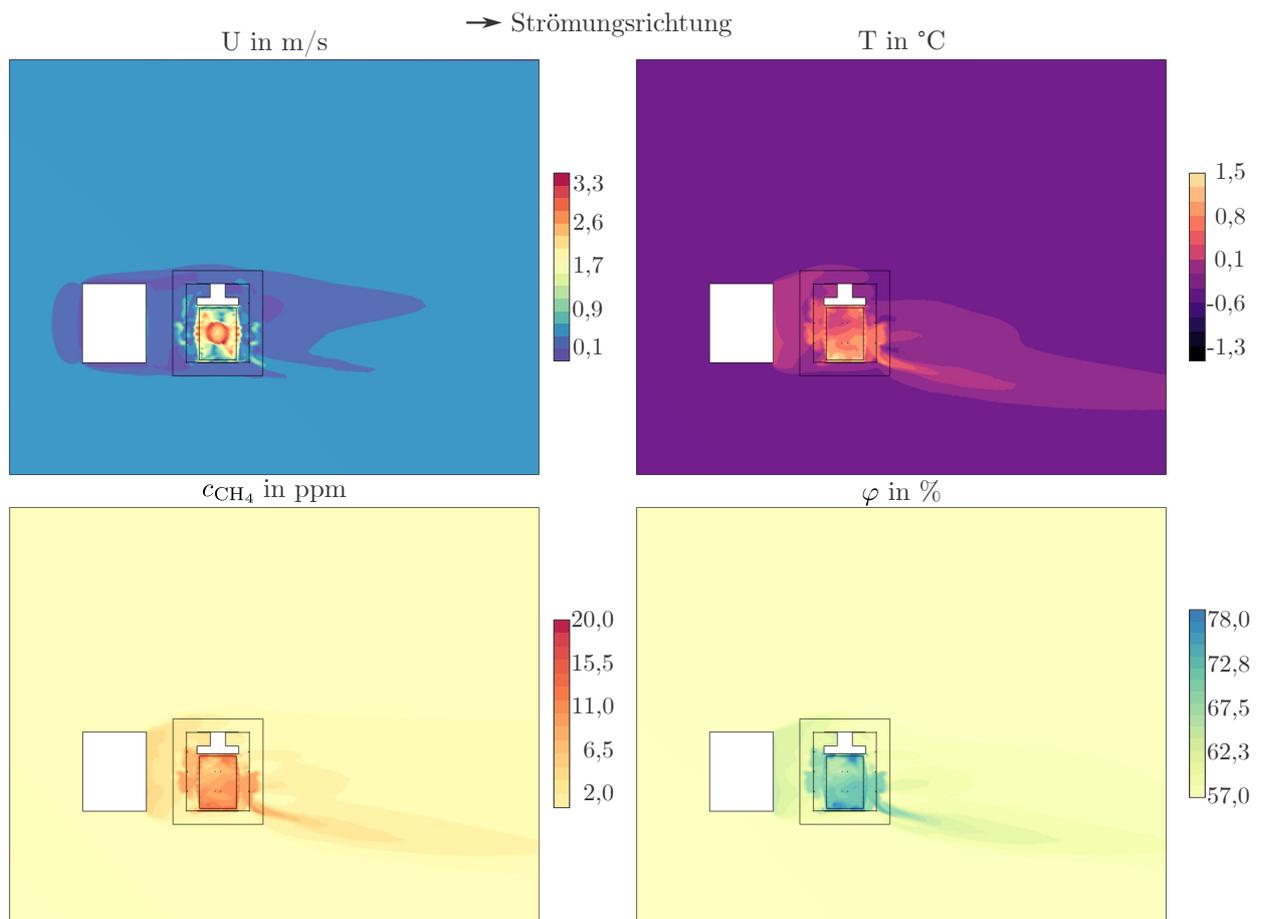


Abbildung D.27: CFD-Ergebnisse des Durchlaufes N14 auf der Schnittebene $x-y$, bei $z = 1,7$ m (eigene Darstellung)

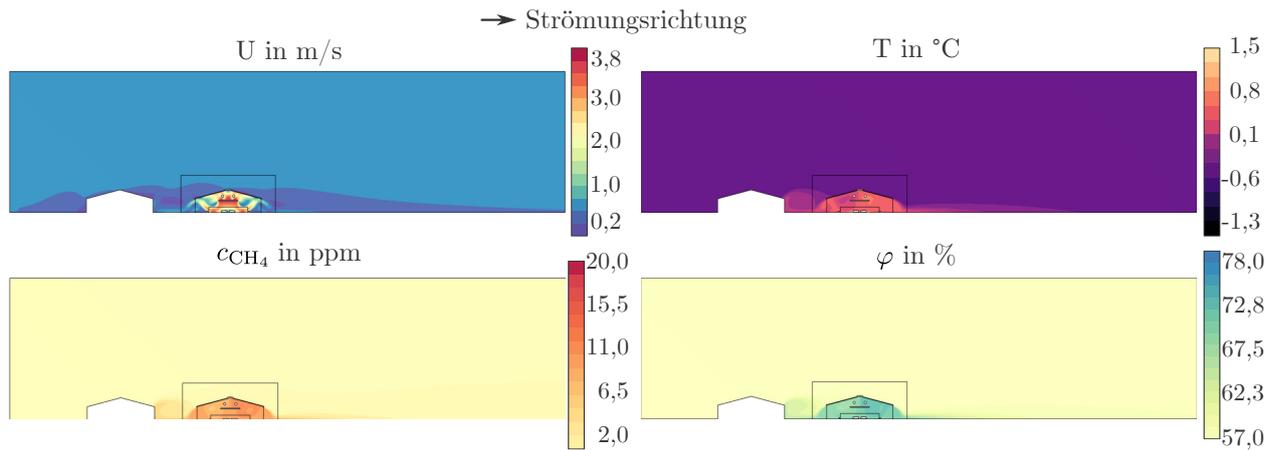


Abbildung D.28: CFD-Ergebnisse des Durchlaufes N14 auf der Schnittebene $x-z$, bei $y = 10,8$ m (eigene Darstellung)

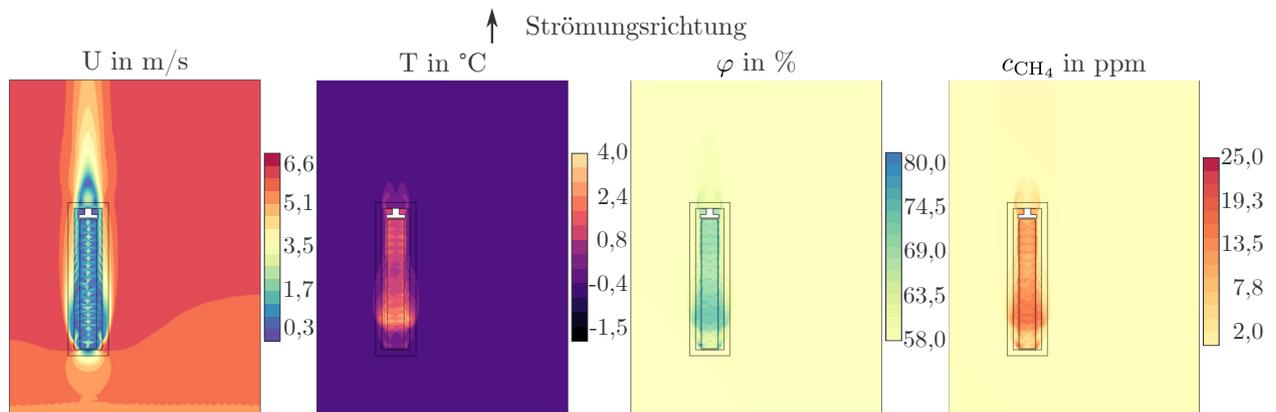


Abbildung D.29: CFD-Ergebnisse des Durchlaufes N15 auf der Schnittebene $x-y$, bei $z = 1,7$ m (eigene Darstellung)

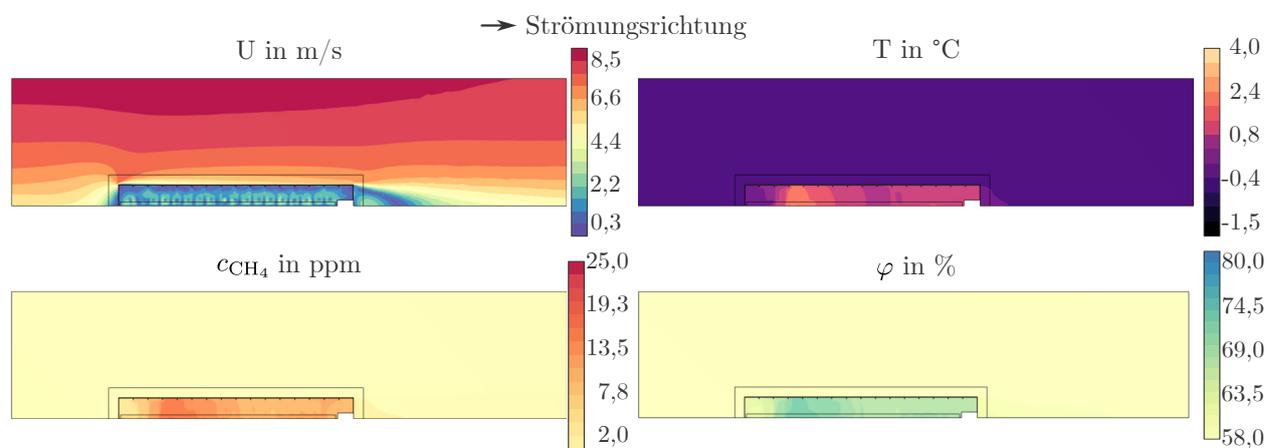


Abbildung D.30: CFD-Ergebnisse des Durchlaufes N15 auf der Schnittebene $y-z$, bei $x = 0,0$ m (eigene Darstellung)

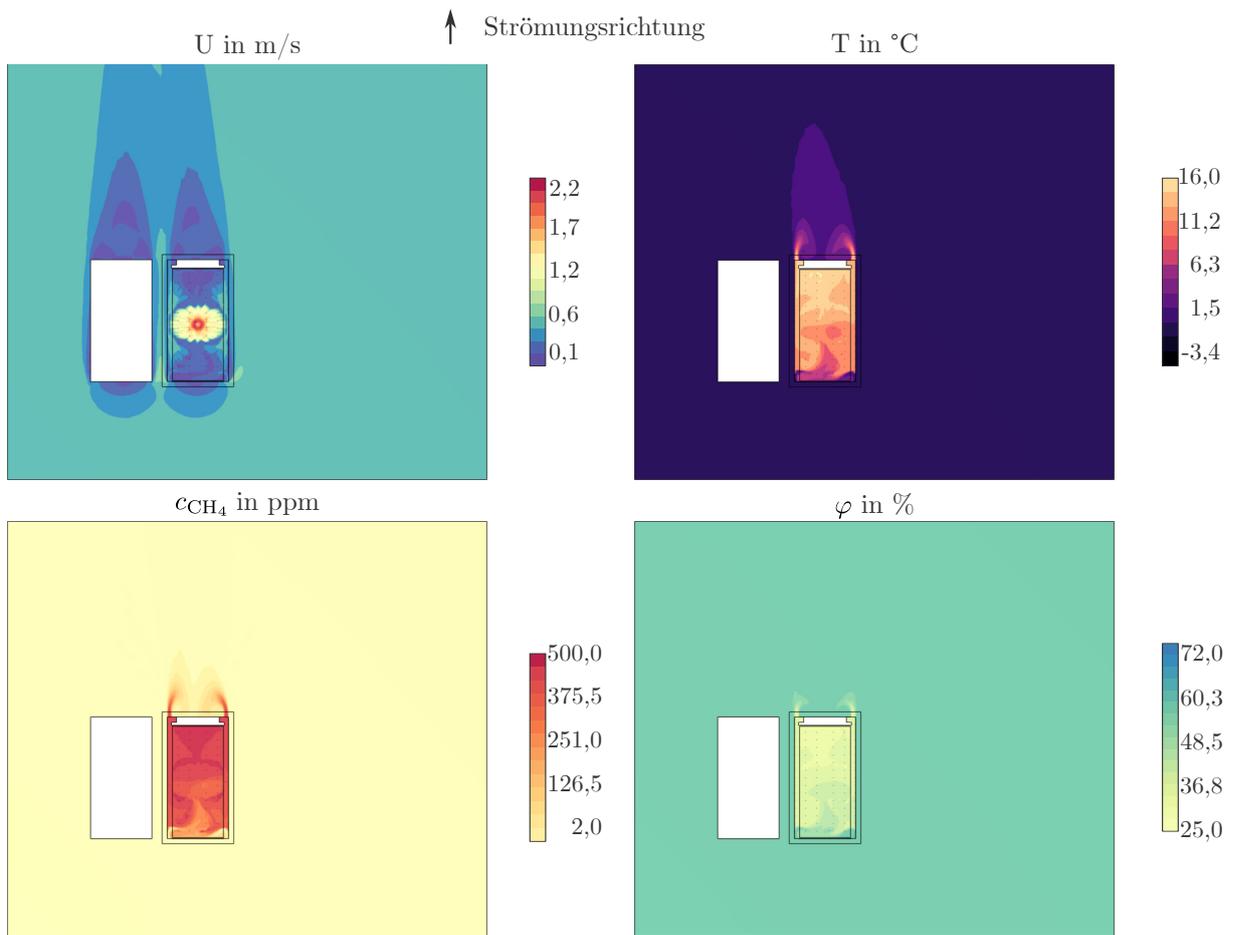


Abbildung D.31: CFD-Ergebnisse des Durchlaufes N16 auf der Schnittebene x - y , bei $z = 1,7$ m (eigene Darstellung)

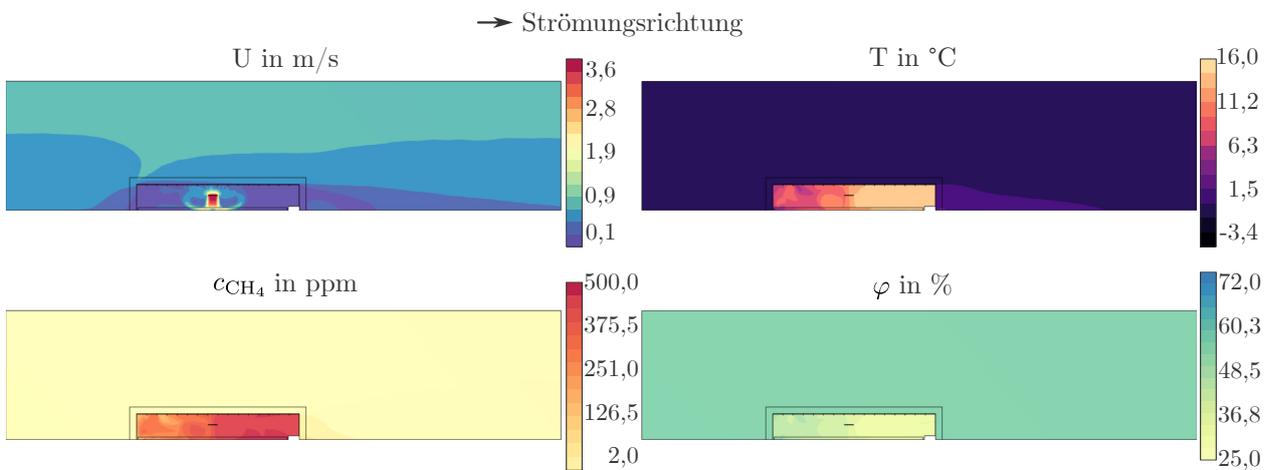


Abbildung D.32: CFD-Ergebnisse des Durchlaufes N16 auf der Schnittebene y - z , bei $x = 0,0$ m (eigene Darstellung)

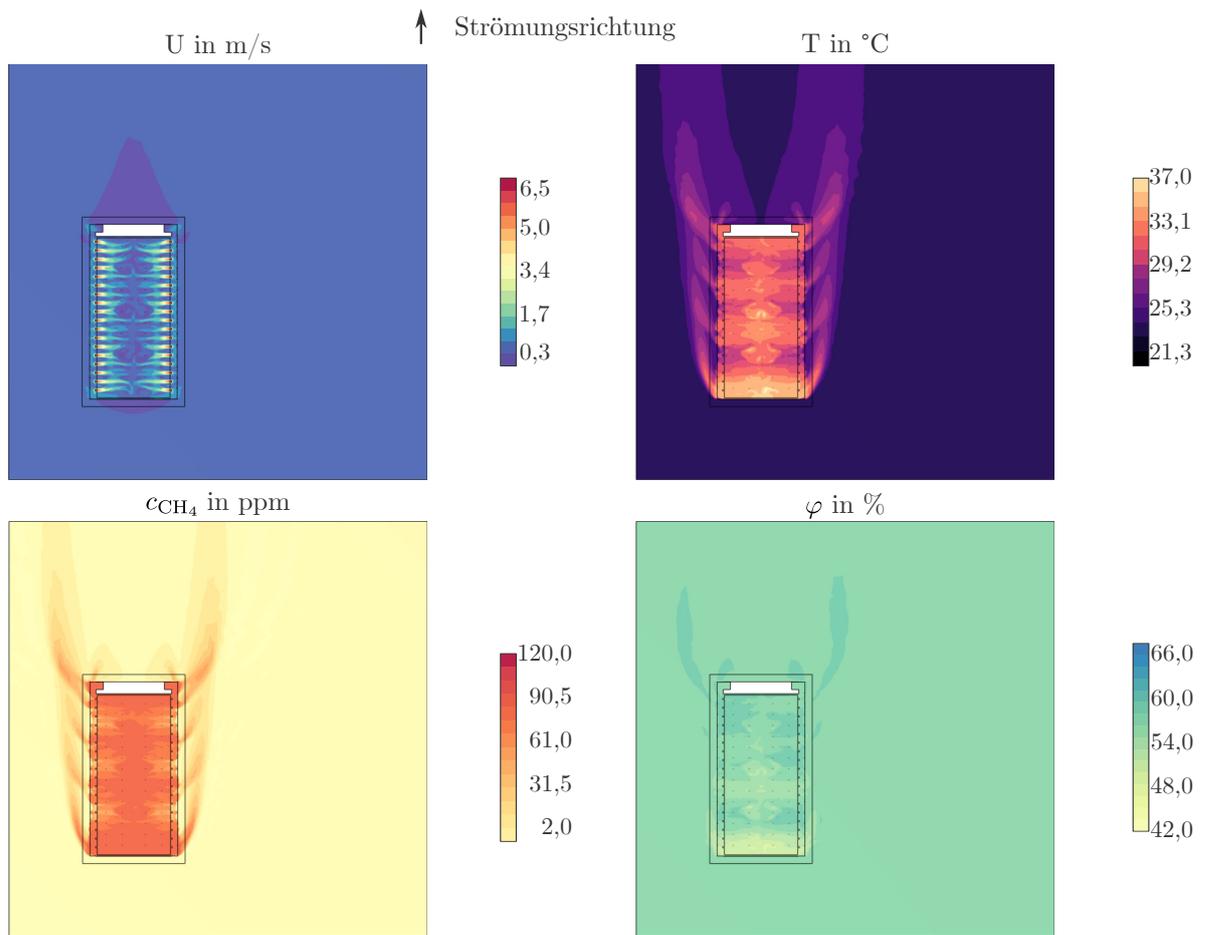


Abbildung D.33: CFD-Ergebnisse des Durchlaufes N17 auf der Schnittebene $x-y$, bei $z = 1,7$ m (eigene Darstellung)

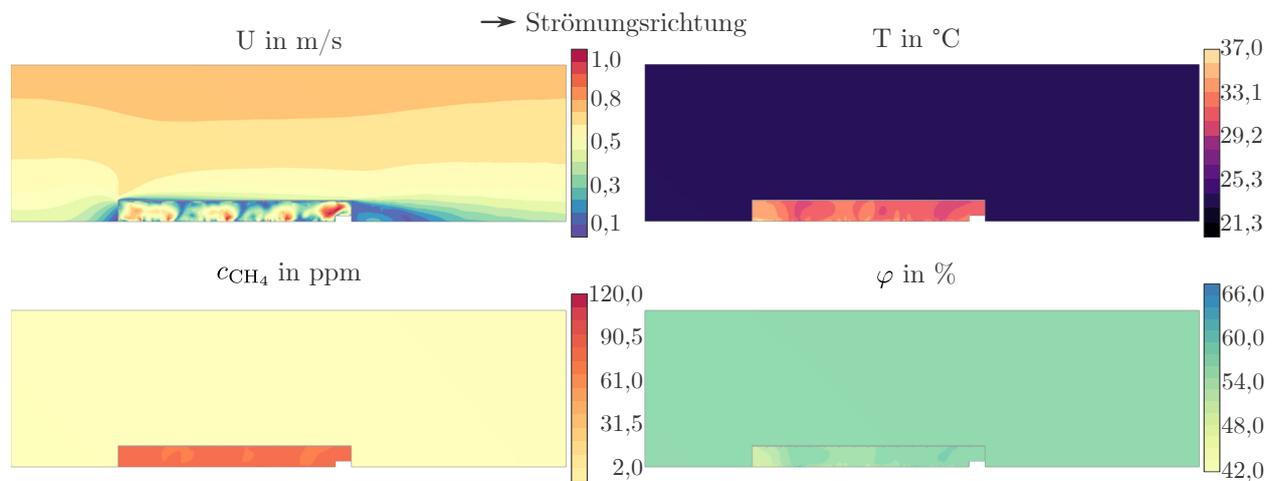


Abbildung D.34: CFD-Ergebnisse des Durchlaufes N17 auf der Schnittebene $y-z$, bei $x = 0,0$ m (eigene Darstellung)

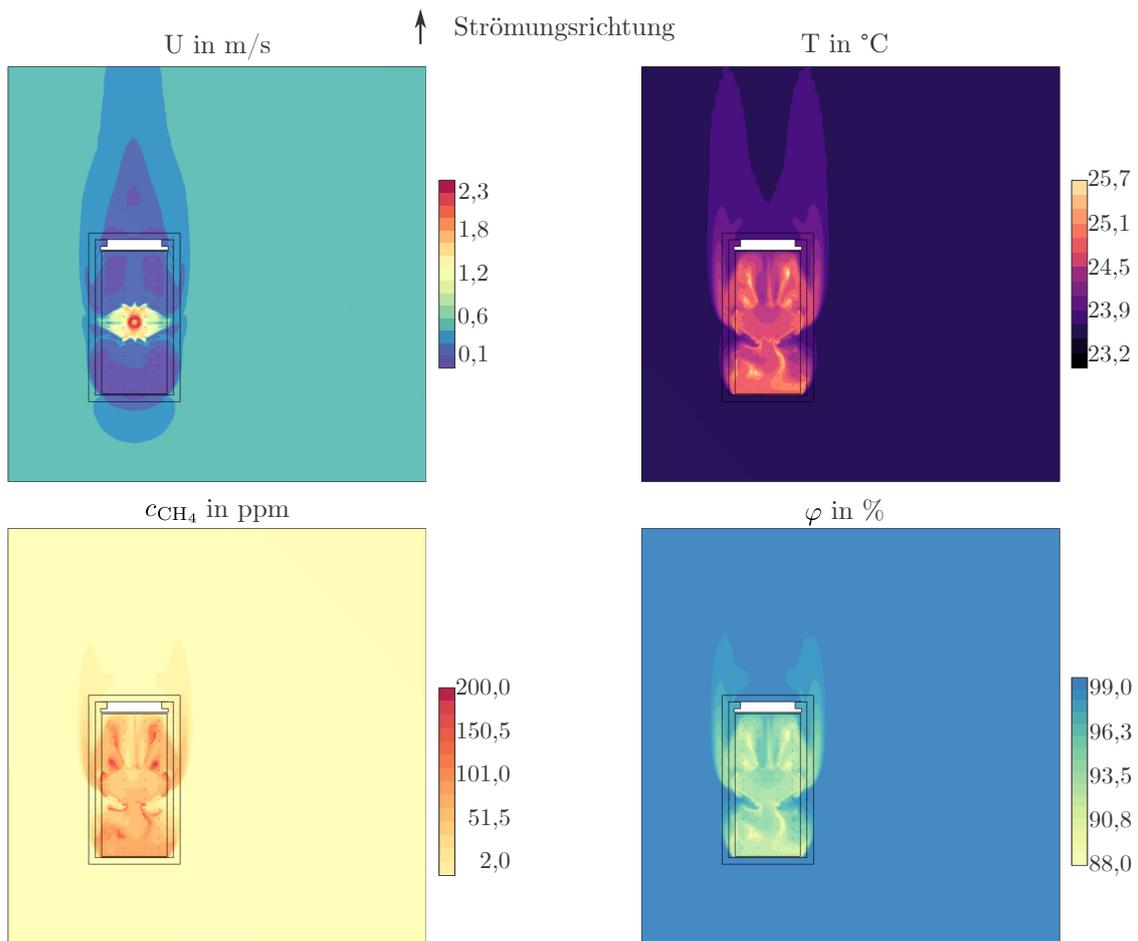


Abbildung D.35: CFD-Ergebnisse des Durchlaufes N18 auf der Schnittebene $x-y$, bei $z = 1,7$ m (eigene Darstellung)

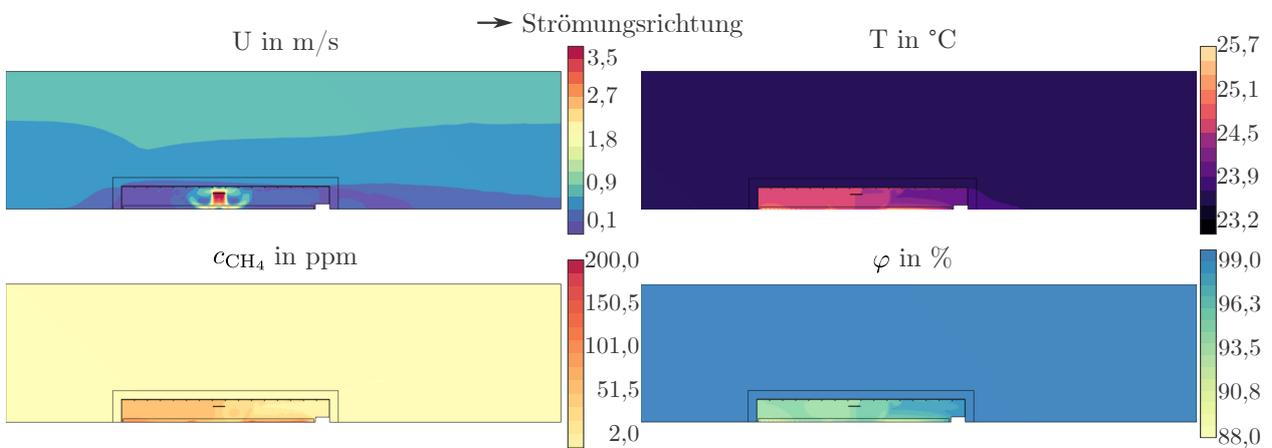


Abbildung D.36: CFD-Ergebnisse des Durchlaufes N18 auf der Schnittebene $y-z$, bei $x = 0,0$ m (eigene Darstellung)

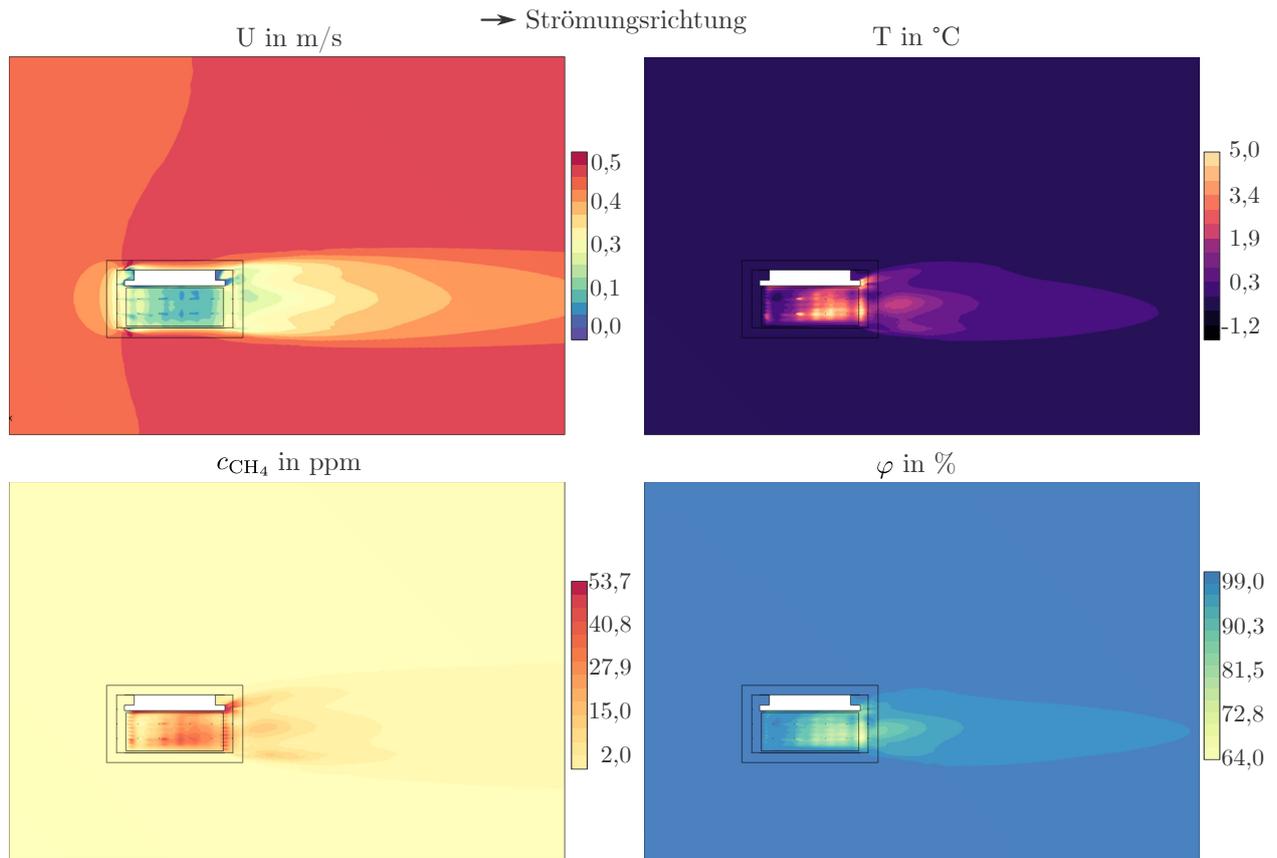


Abbildung D.37: CFD-Ergebnisse des Durchlaufes N19 auf der Schnittebene $x-y$, bei $z = 1,7$ m (eigene Darstellung)

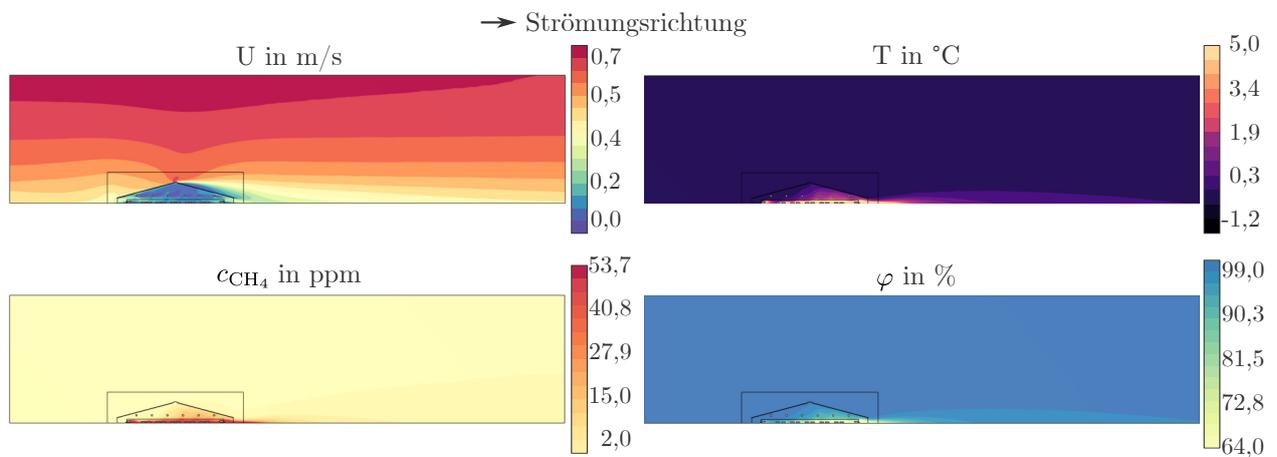


Abbildung D.38: CFD-Ergebnisse des Durchlaufes N19 auf der Schnittebene $x-z$, bei $y = 9,4$ m (eigene Darstellung)

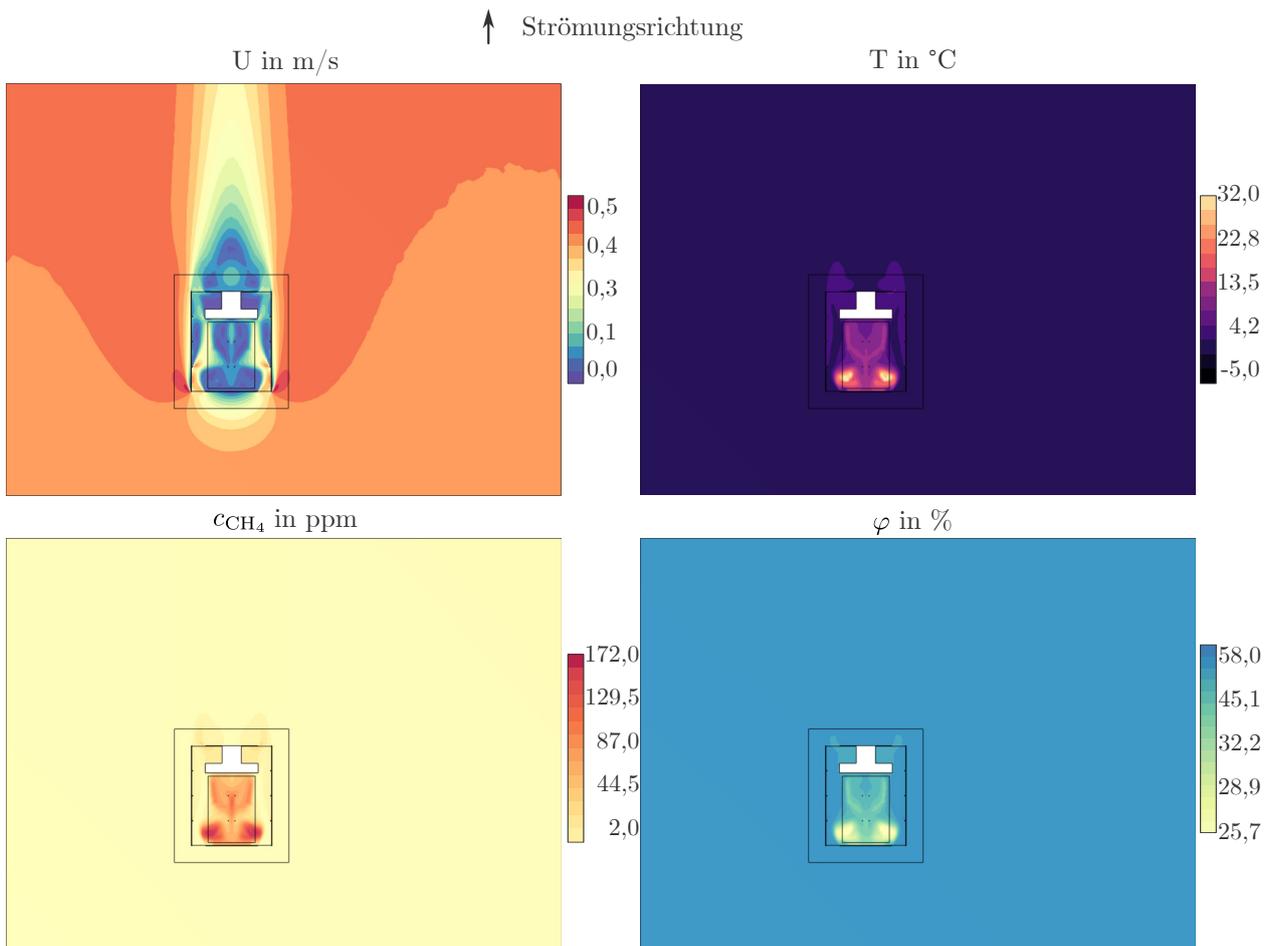


Abbildung D.39: CFD-Ergebnisse des Durchlaufes N20 auf der Schnittebene $x-y$, bei $z = 1,7$ m (eigene Darstellung)

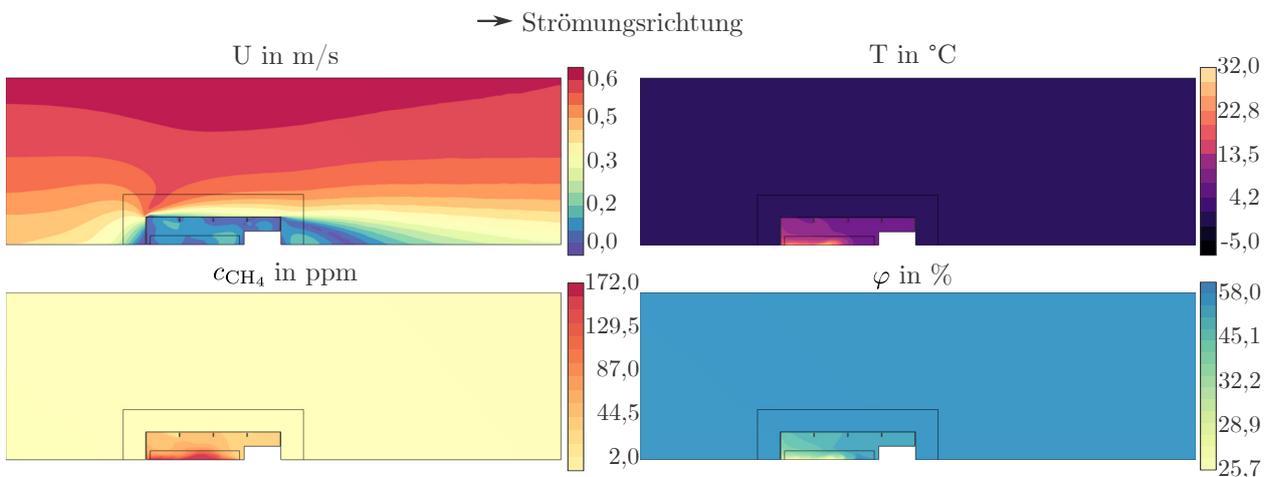


Abbildung D.40: CFD-Ergebnisse des Durchlaufes N20 auf der Schnittebene $y-z$, bei $x = 0,0$ m (eigene Darstellung)

D.2 Konturdiagramme des Box-Behnken-Versuchsplans

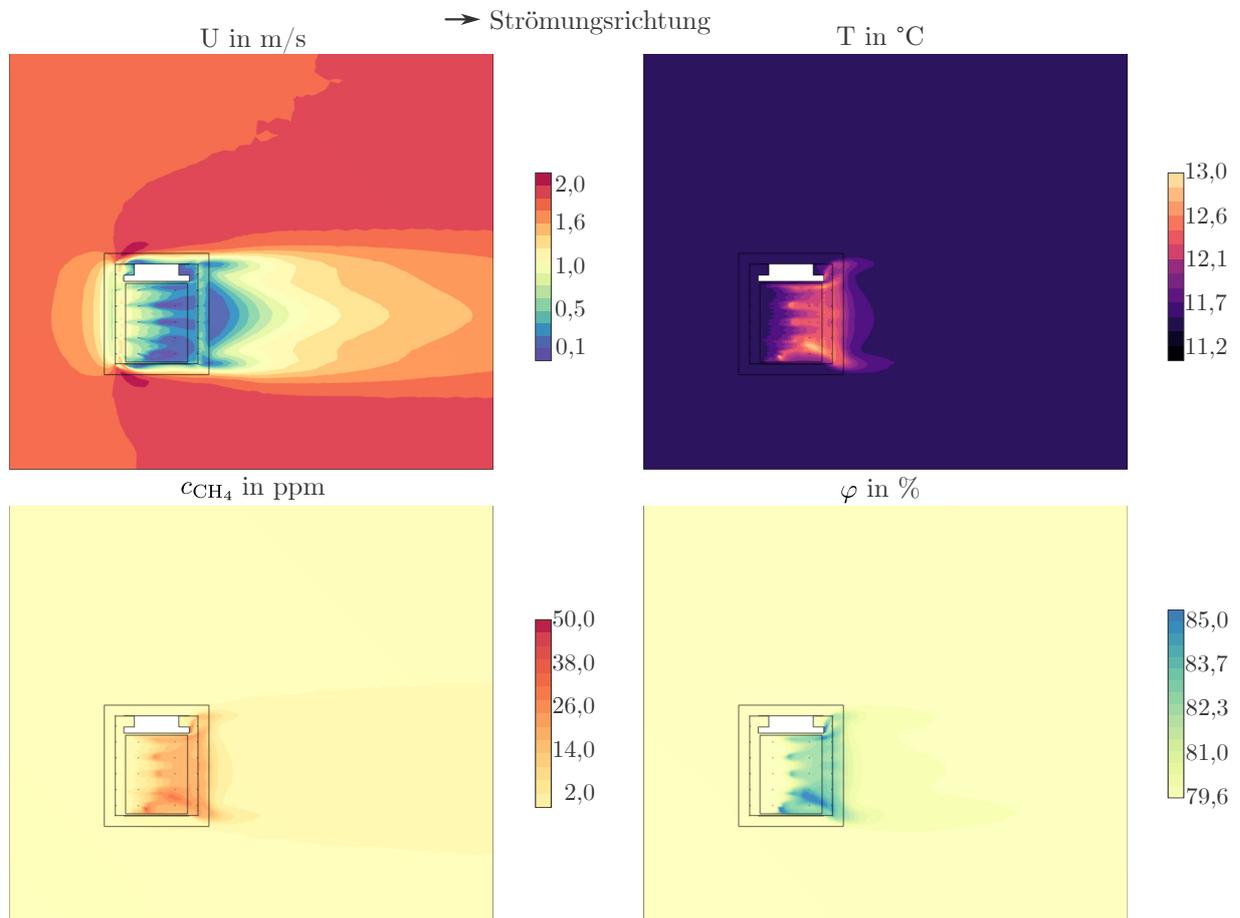


Abbildung D.41: CFD-Ergebnisse des Durchlaufes N1 auf der Schnittebene $x-y$, bei $z = 1,7$ m (eigene Darstellung)

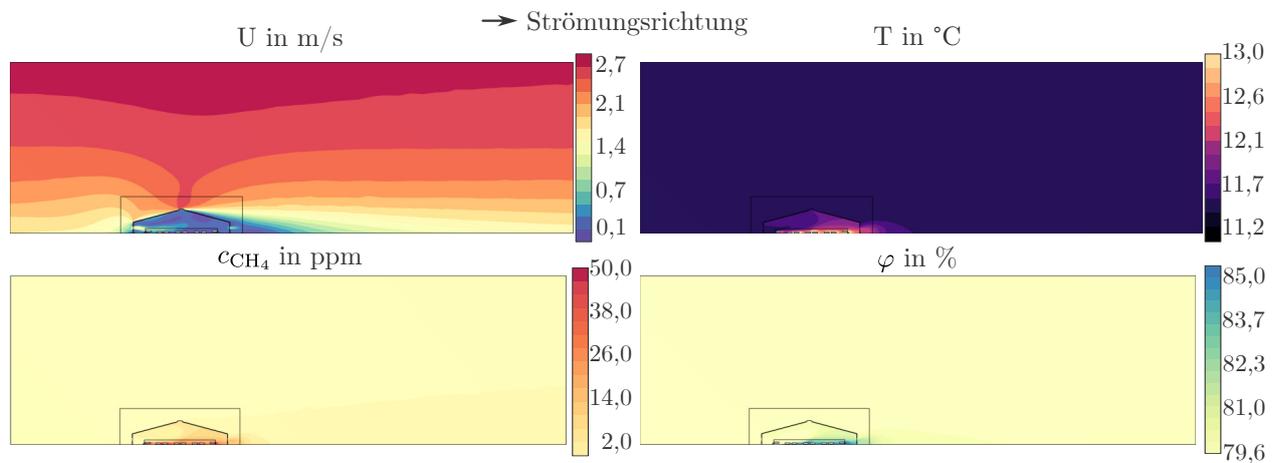


Abbildung D.42: CFD-Ergebnisse des Durchlaufes N1 auf der Schnittebene $x-z$, bei $y = 20,8$ m (eigene Darstellung)

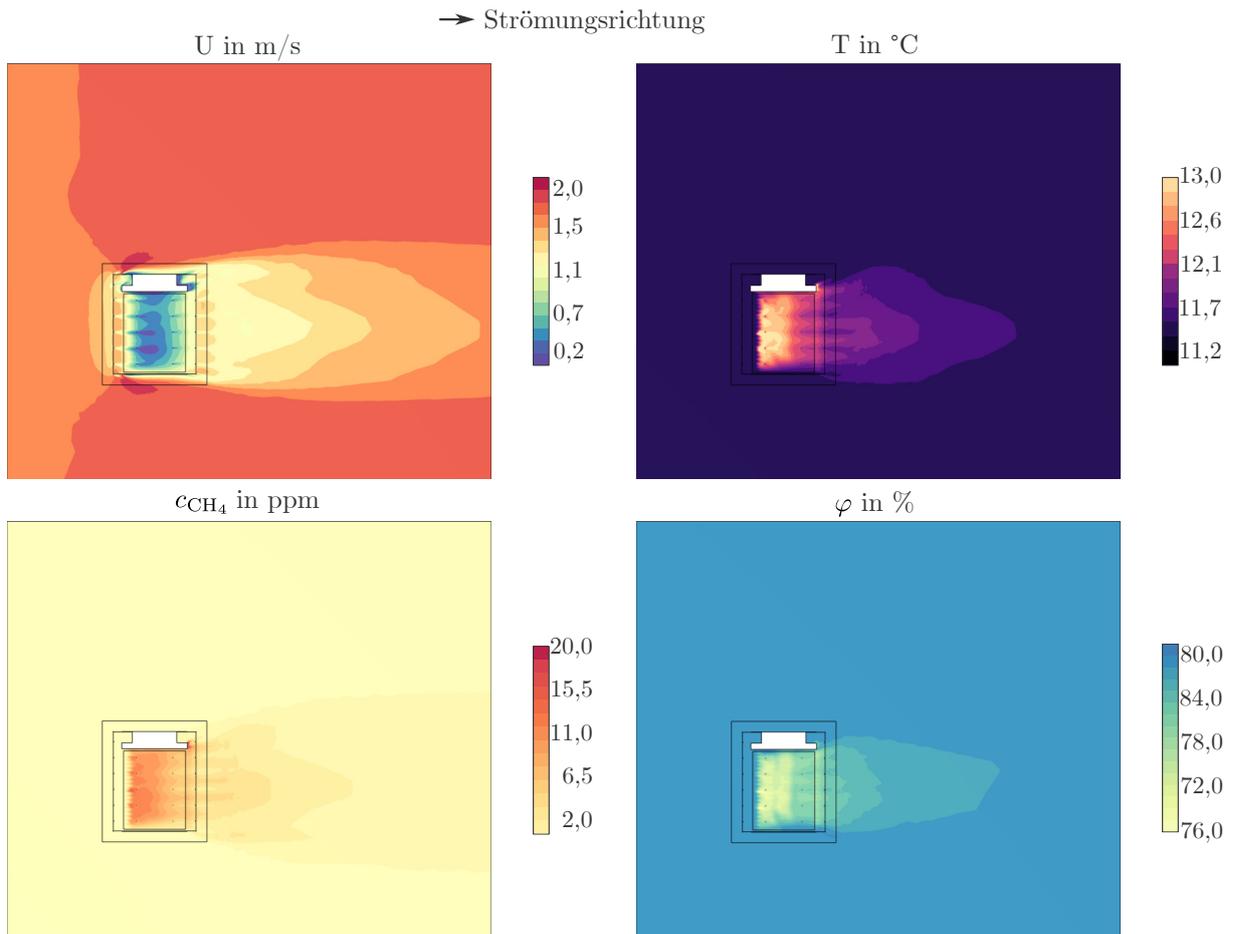


Abbildung D.43: CFD-Ergebnisse des Durchlaufes N2 auf der Schnittebene $x-y$, bei $z = 1,7$ m (eigene Darstellung)

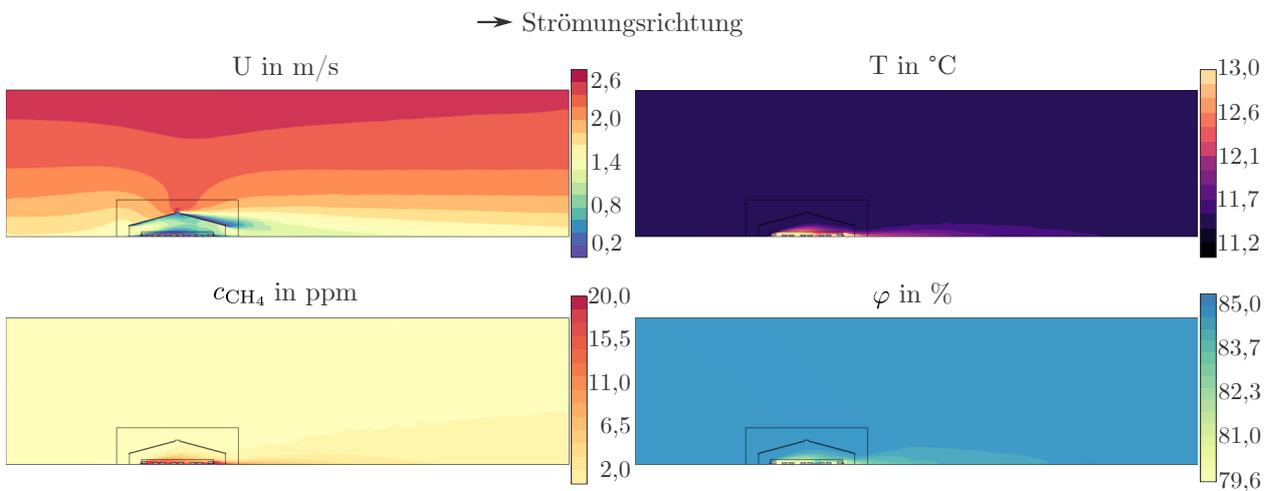


Abbildung D.44: CFD-Ergebnisse des Durchlaufes N2 auf der Schnittebene $x-z$, bei $y = 20,8$ m (eigene Darstellung)

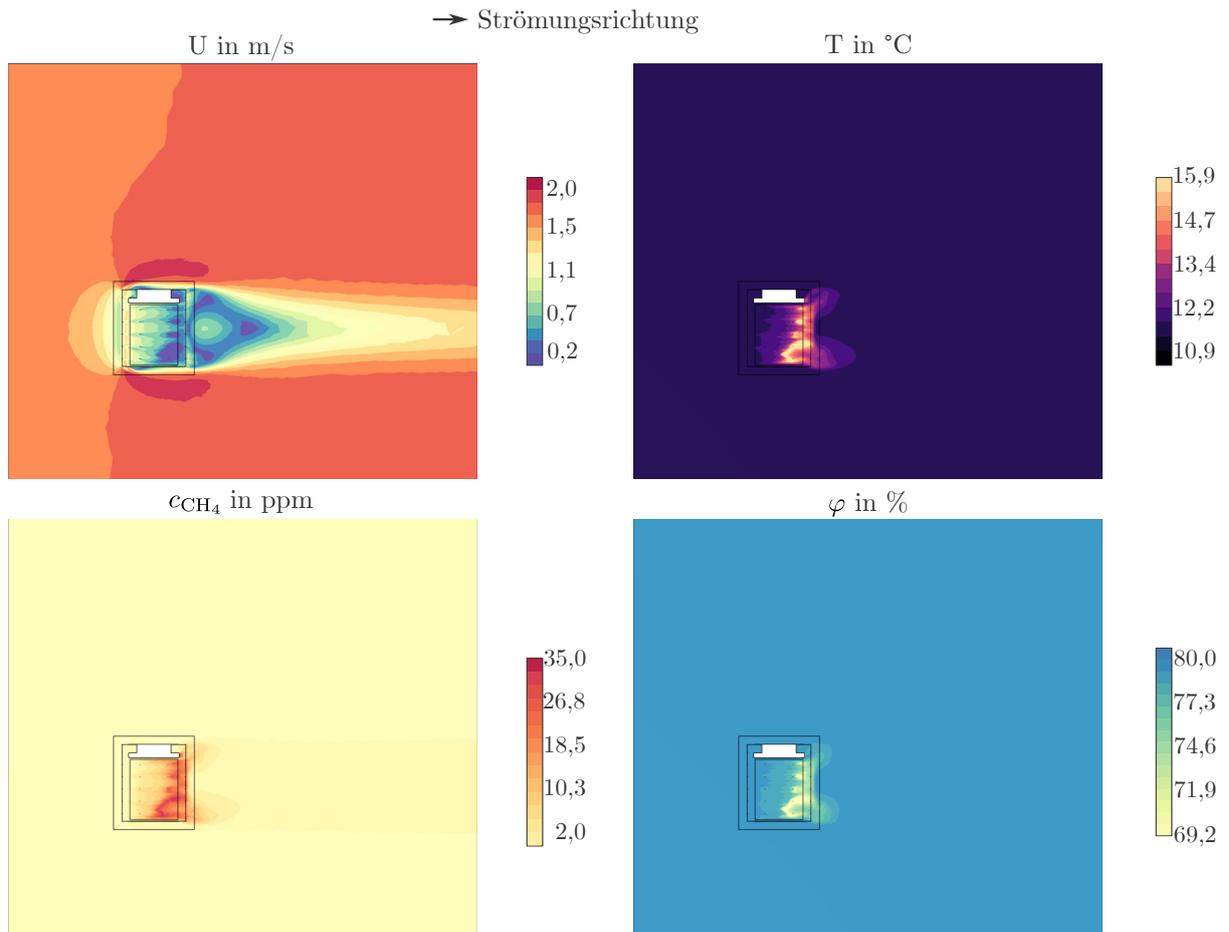


Abbildung D.45: CFD-Ergebnisse des Durchlaufes N3 auf der Schnittebene $x-y$, bei $z = 1,7$ m (eigene Darstellung)

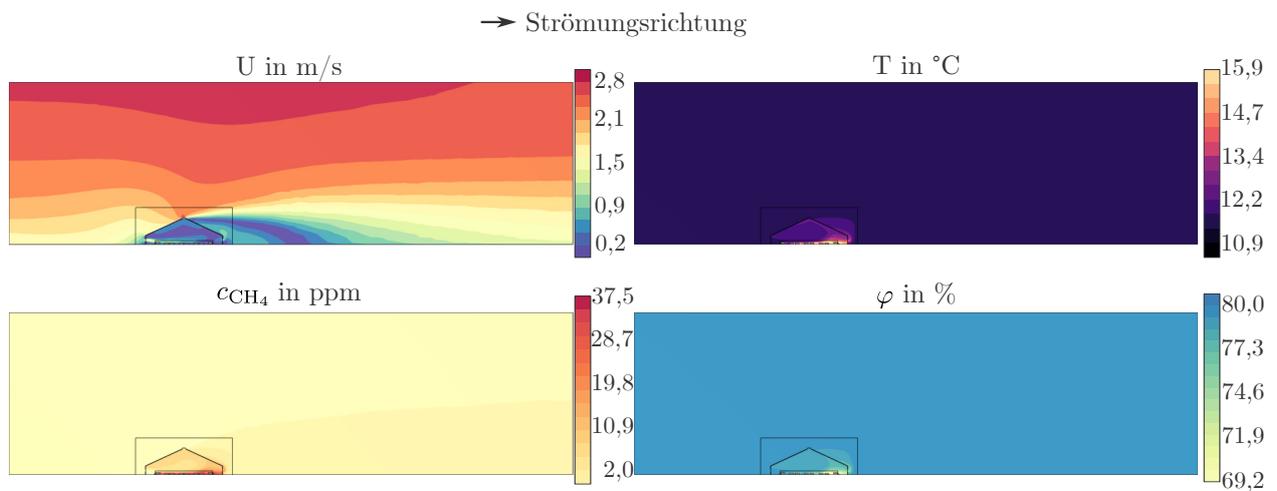


Abbildung D.46: CFD-Ergebnisse des Durchlaufes N3 auf der Schnittebene $x-z$, bei $y = 20,8$ m (eigene Darstellung)

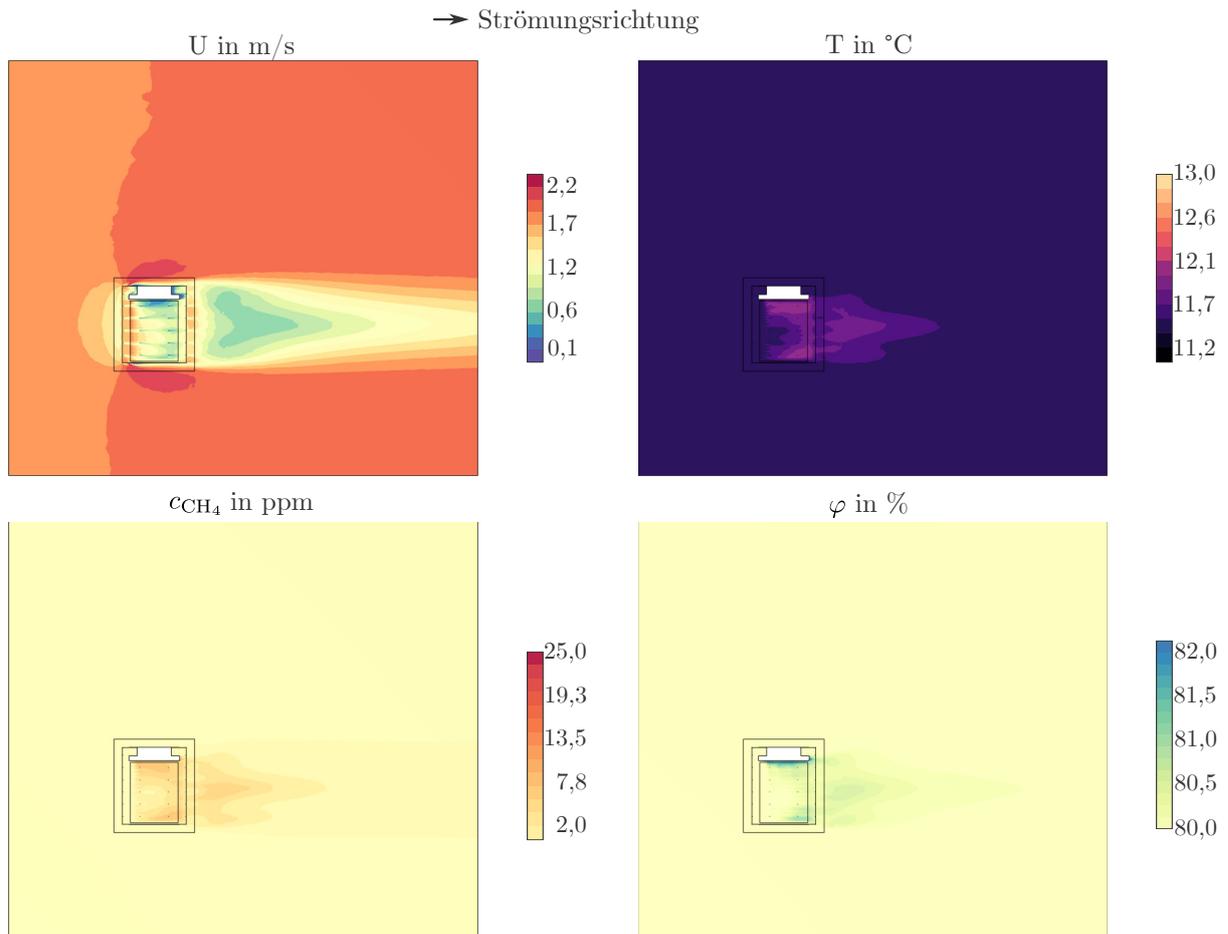


Abbildung D.47: CFD-Ergebnisse des Durchlaufes N4 auf der Schnittebene $x-y$, bei $z = 1,7$ m (eigene Darstellung)

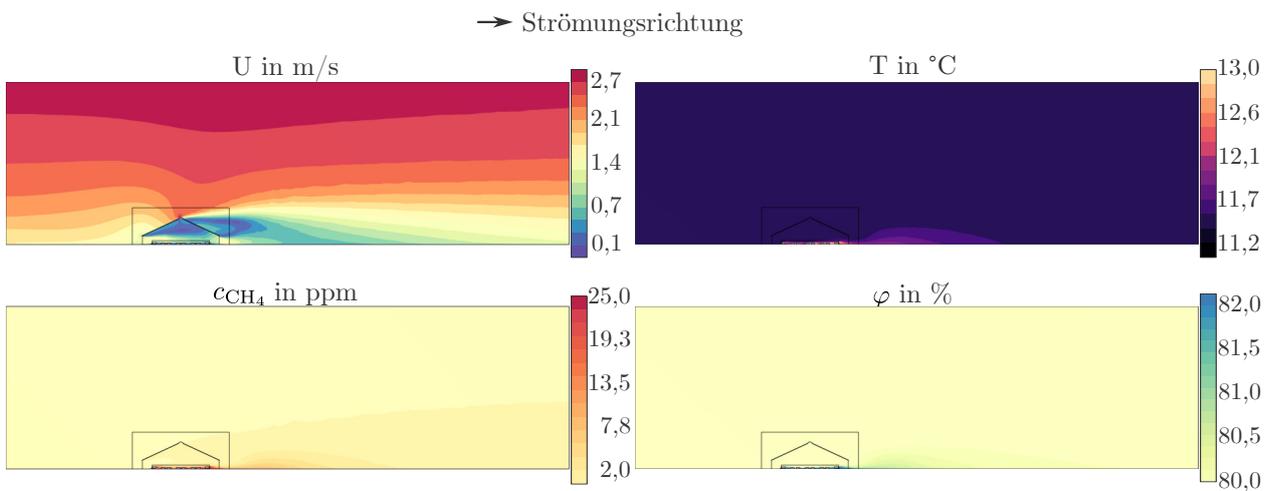


Abbildung D.48: CFD-Ergebnisse des Durchlaufes N4 auf der Schnittebene $x-z$, bei $y = 20,8$ m (eigene Darstellung)

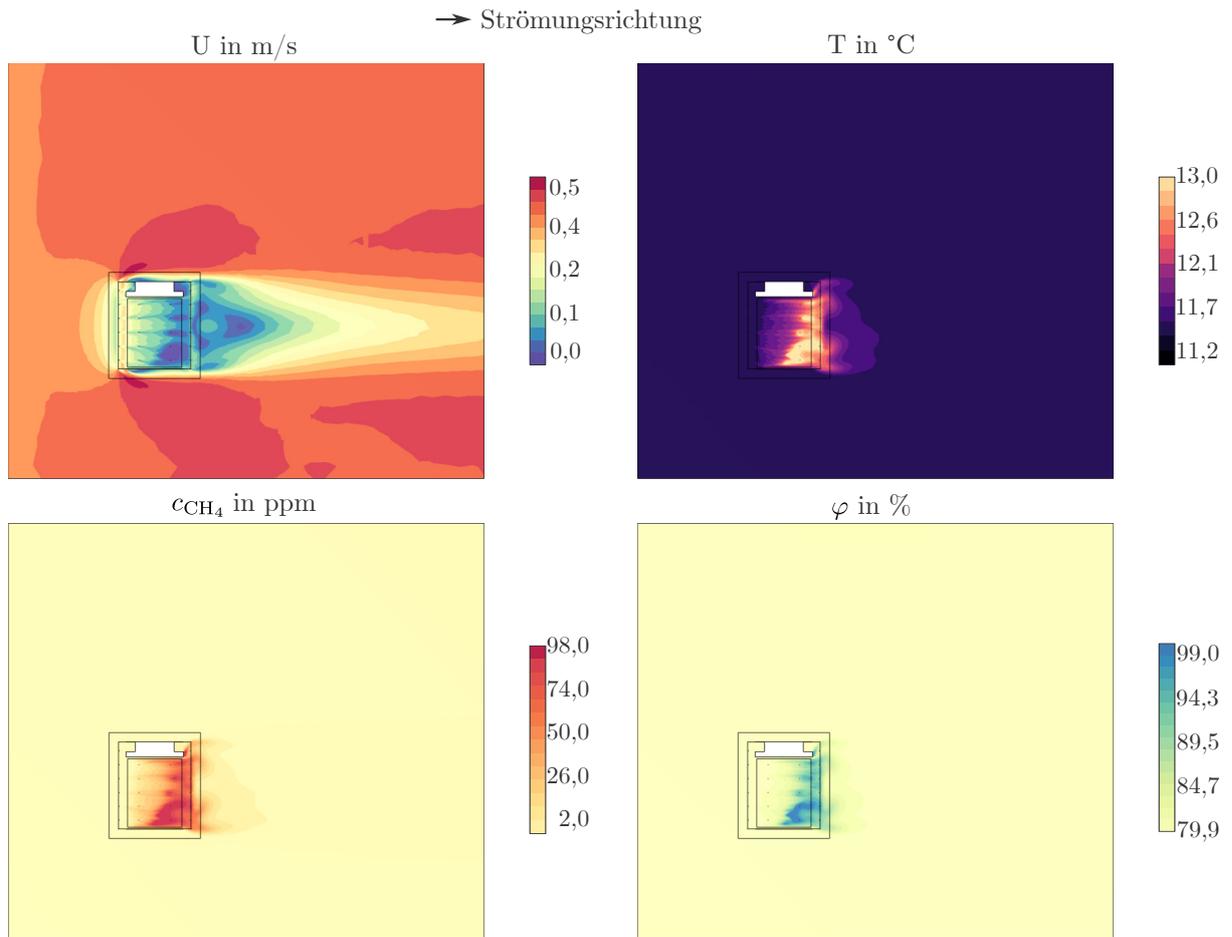


Abbildung D.49: CFD-Ergebnisse des Durchlaufes N5 auf der Schnittebene $x-y$, bei $z = 1,7$ m (eigene Darstellung)

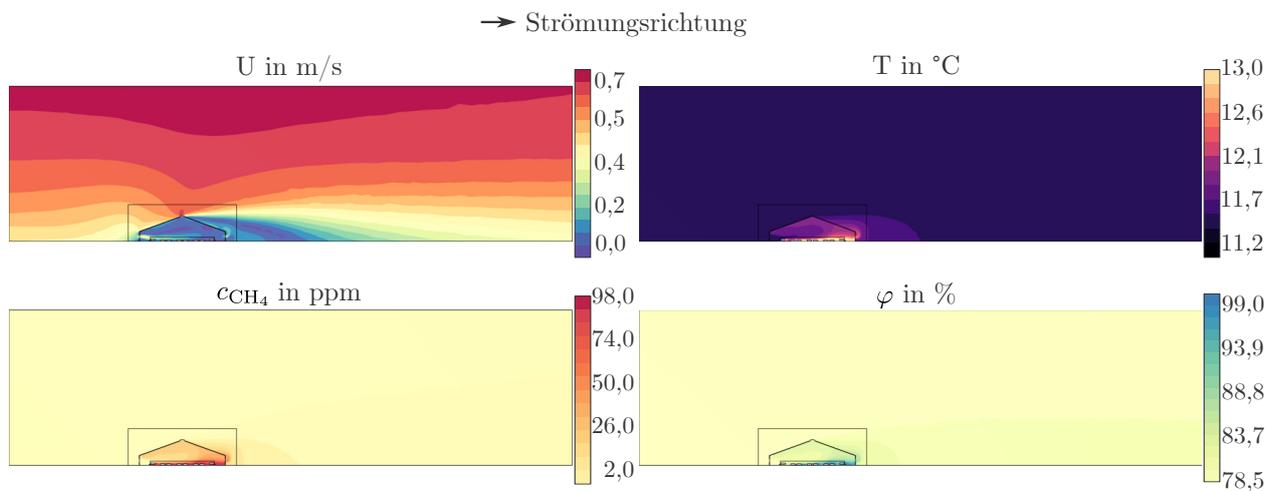


Abbildung D.50: CFD-Ergebnisse des Durchlaufes N5 auf der Schnittebene $x-z$, bei $y = 20,8$ m (eigene Darstellung)

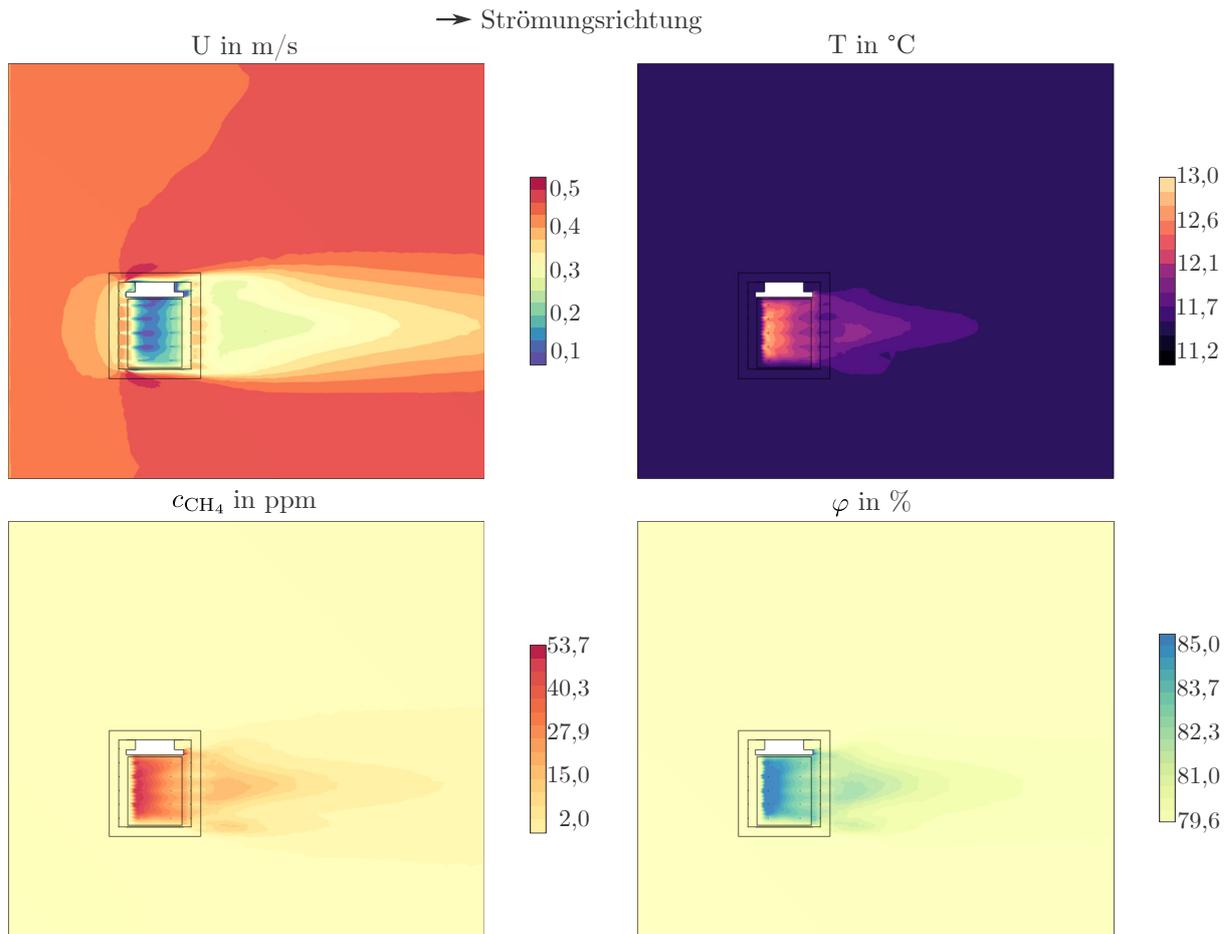


Abbildung D.51: CFD-Ergebnisse des Durchlaufes N6 auf der Schnittebene $x-y$, bei $z = 1,7$ m (eigene Darstellung)

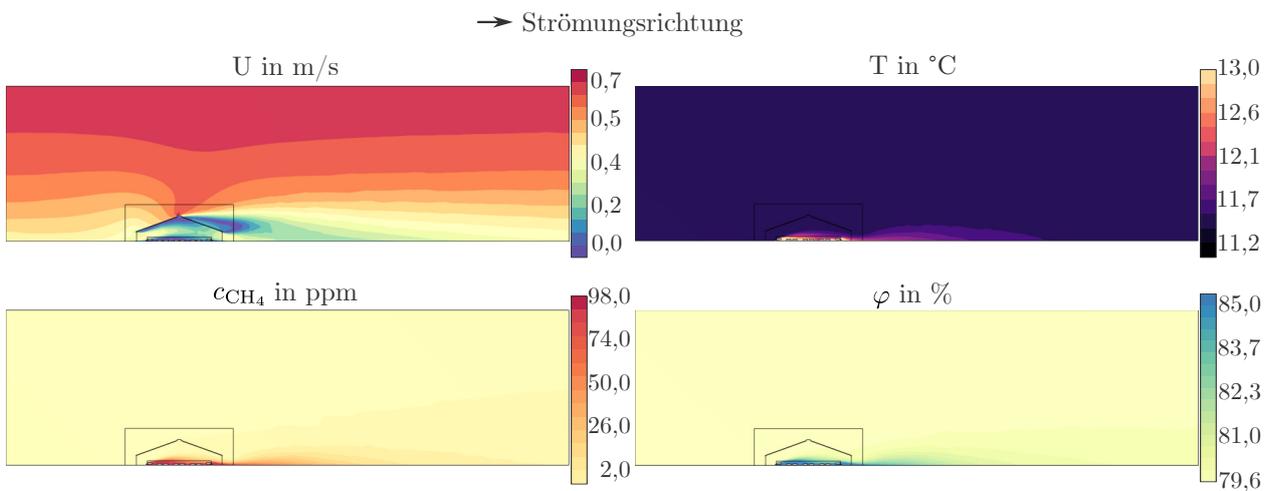


Abbildung D.52: CFD-Ergebnisse des Durchlaufes N6 auf der Schnittebene $x-z$, bei $y = 20,8$ m (eigene Darstellung)

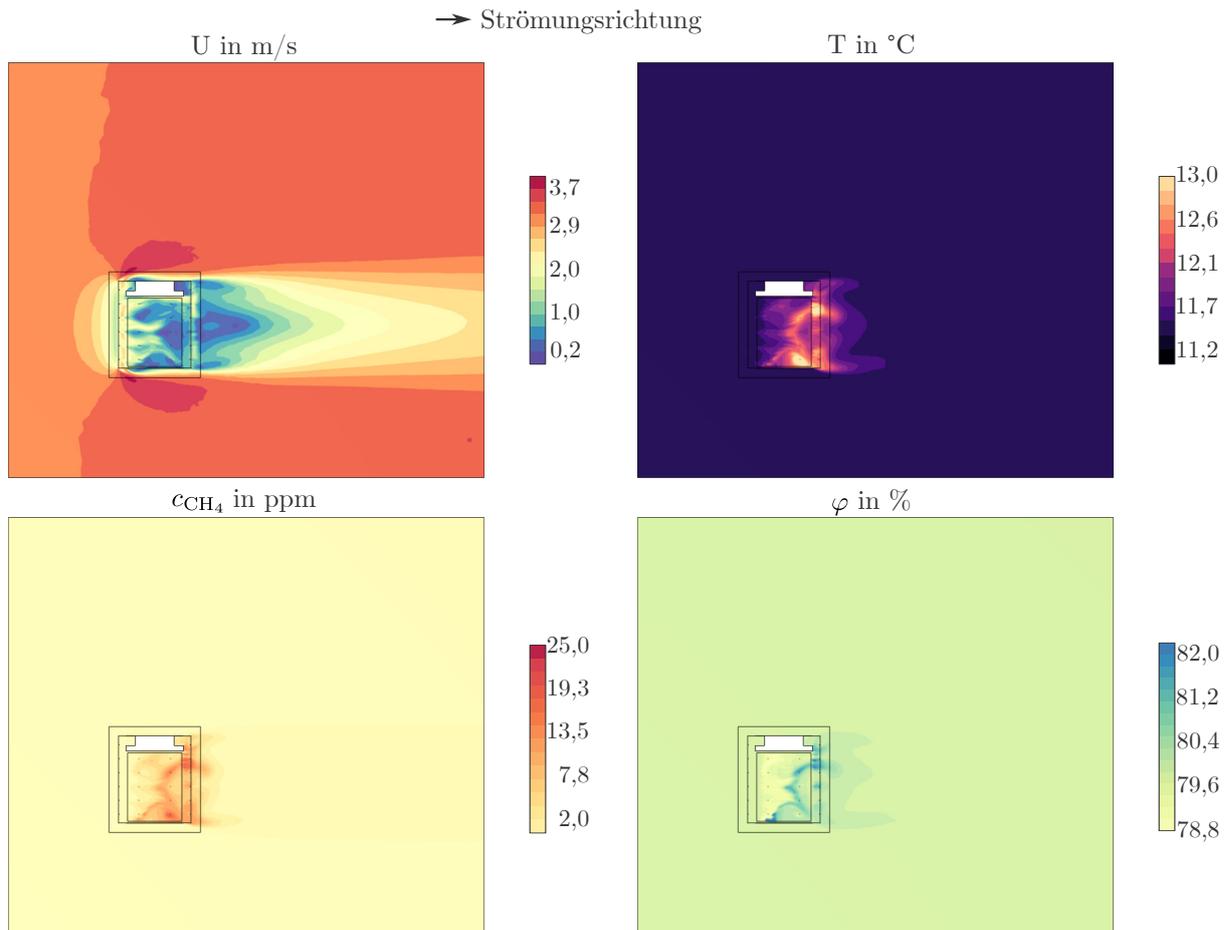


Abbildung D.53: CFD-Ergebnisse des Durchlaufes N7 auf der Schnittebene $x-y$, bei $z = 1,7$ m (eigene Darstellung)

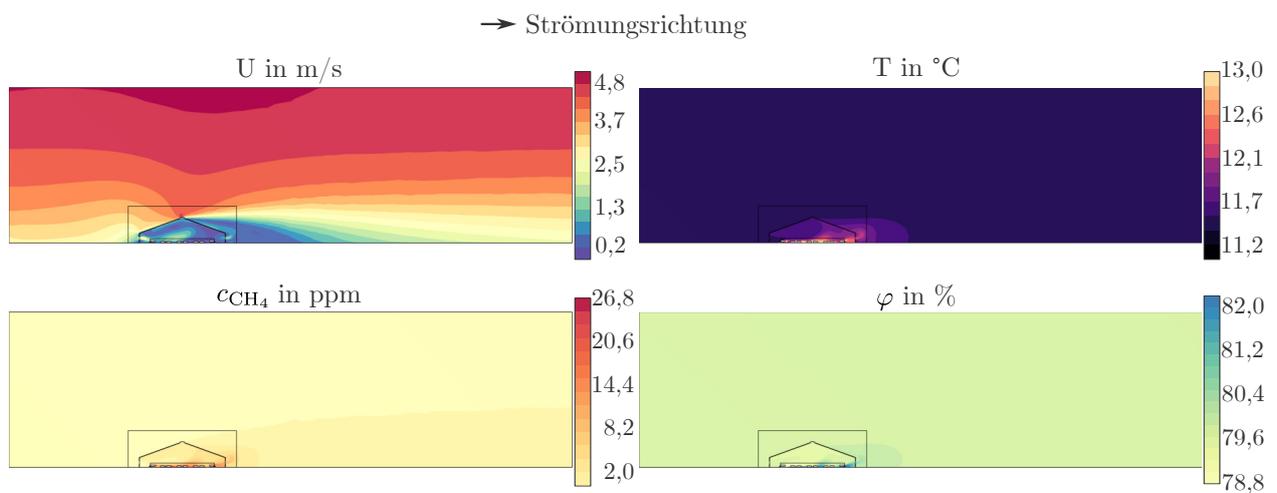


Abbildung D.54: CFD-Ergebnisse des Durchlaufes N7 auf der Schnittebene $x-z$, bei $y = 20,8$ m (eigene Darstellung)

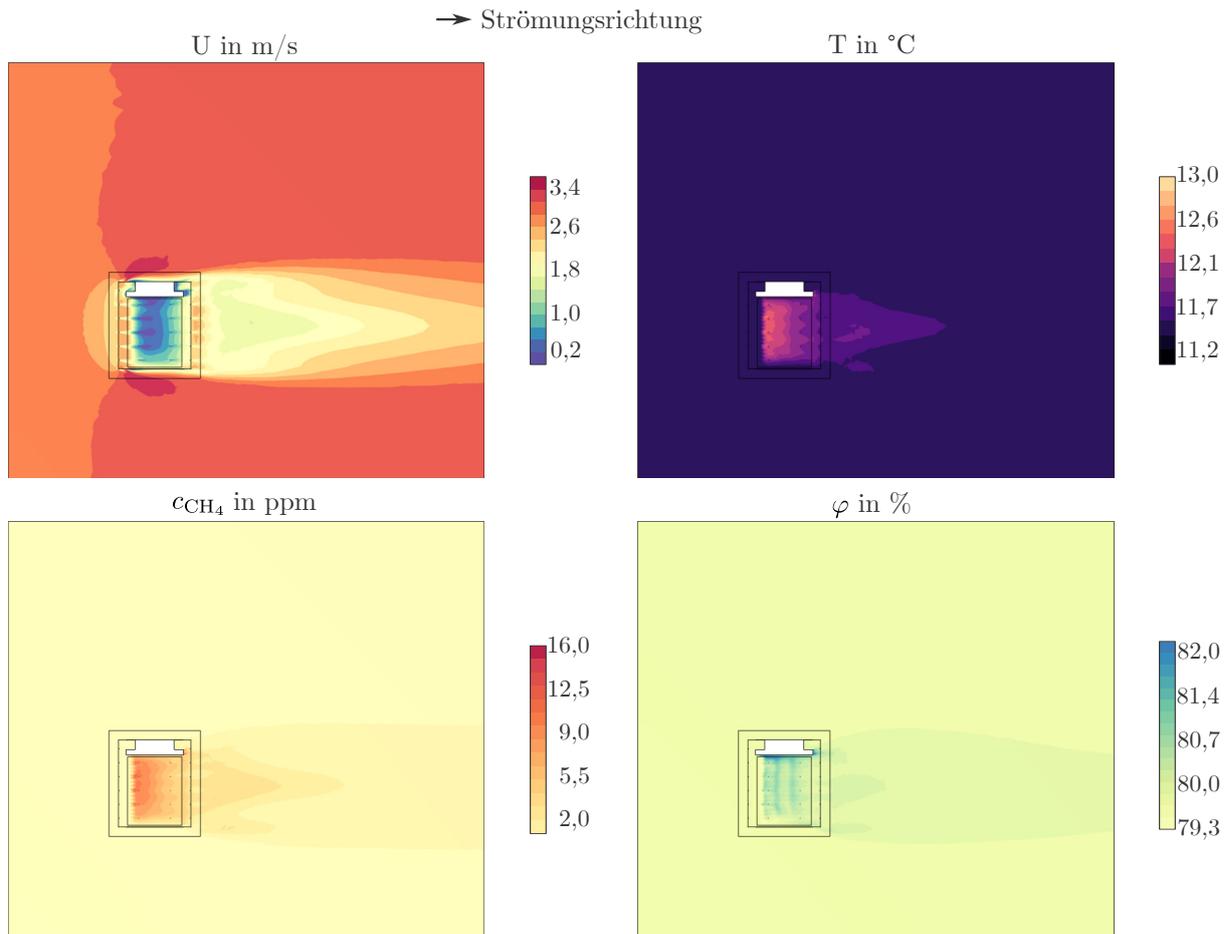


Abbildung D.55: CFD-Ergebnisse des Durchlaufes N8 auf der Schnittebene $x-y$, bei $z = 1,7$ m (eigene Darstellung)

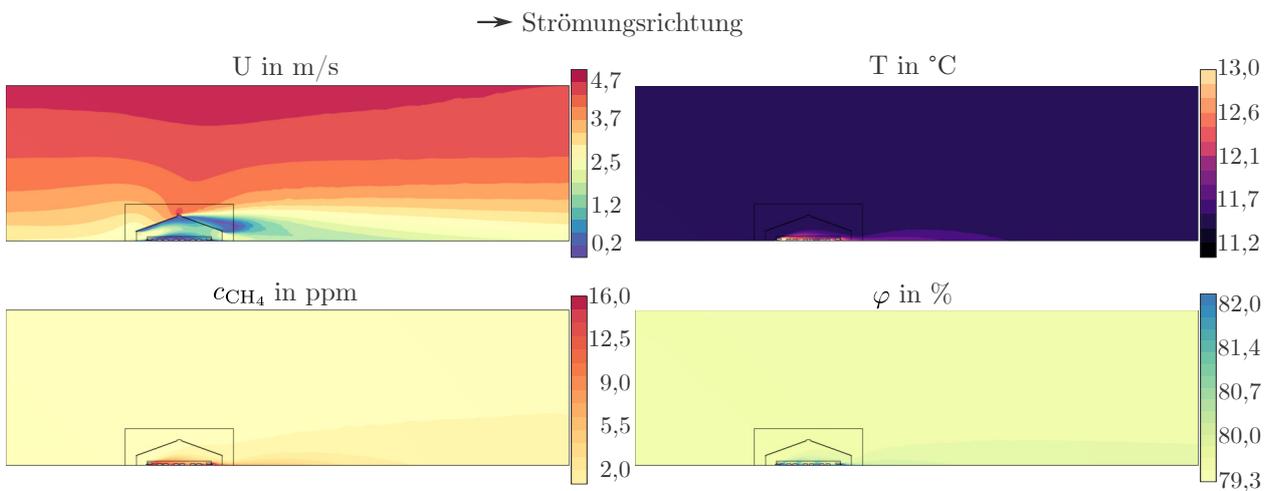


Abbildung D.56: CFD-Ergebnisse des Durchlaufes N8 auf der Schnittebene $x-z$, bei $y = 20,8$ m (eigene Darstellung)

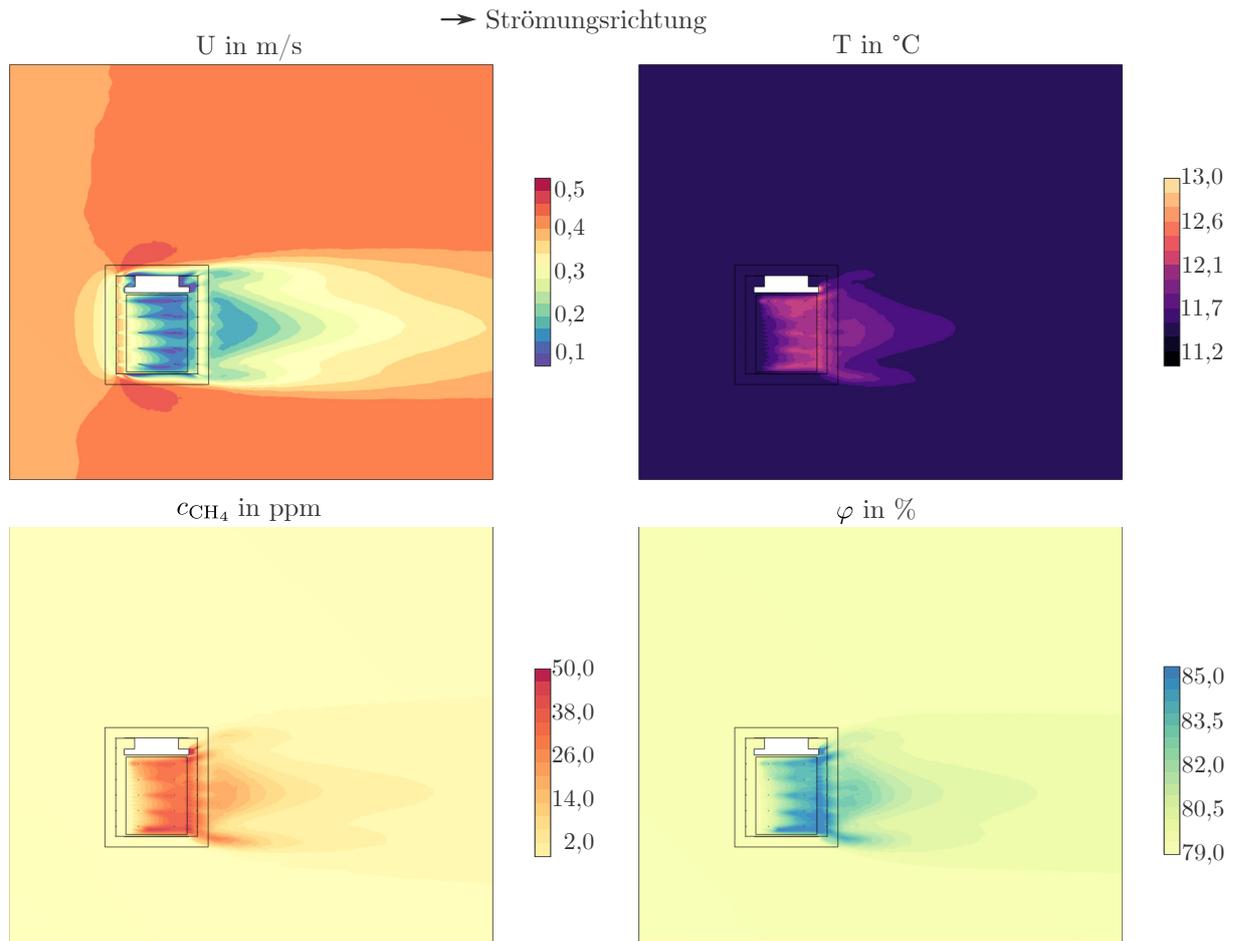


Abbildung D.57: CFD-Ergebnisse des Durchlaufes N9 auf der Schnittebene $x-y$, bei $z = 1,7$ m (eigene Darstellung)

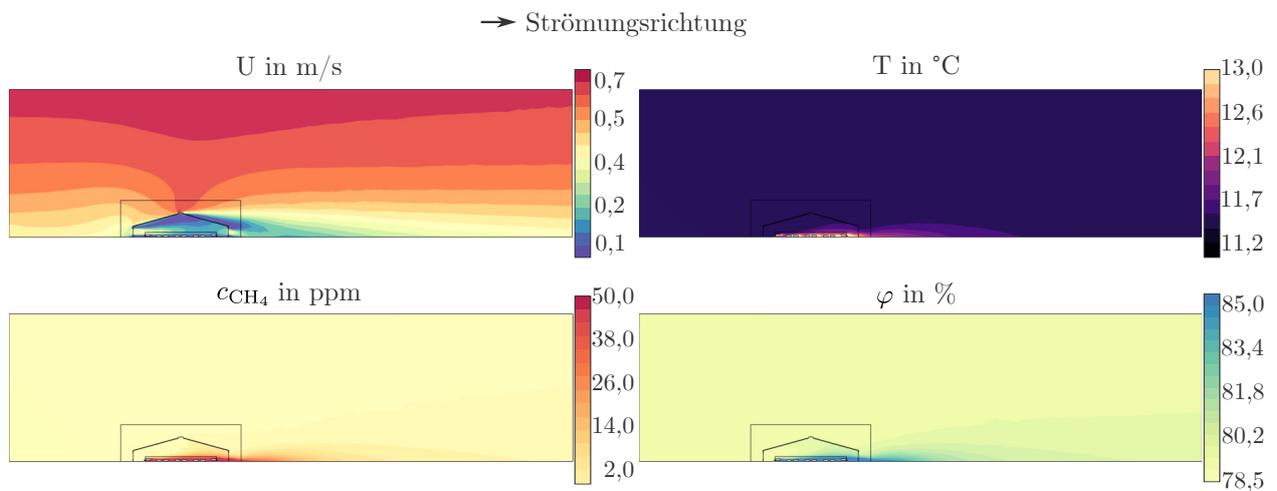


Abbildung D.58: CFD-Ergebnisse des Durchlaufes N9 auf der Schnittebene $x-z$, bei $y = 20,8$ m (eigene Darstellung)

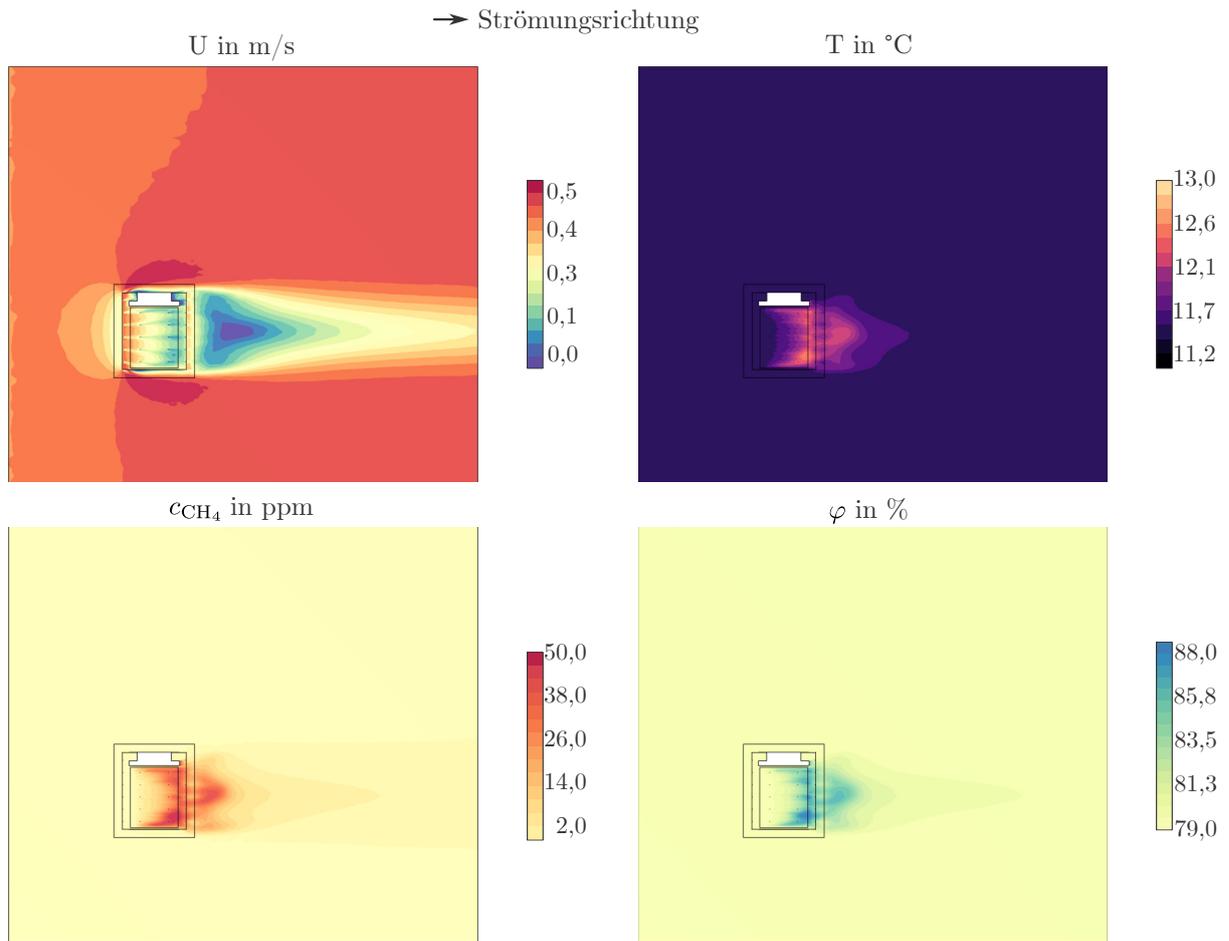


Abbildung D.59: CFD-Ergebnisse des Durchlaufes N10 auf der Schnittebene x - y , bei $z = 1,7$ m (eigene Darstellung)

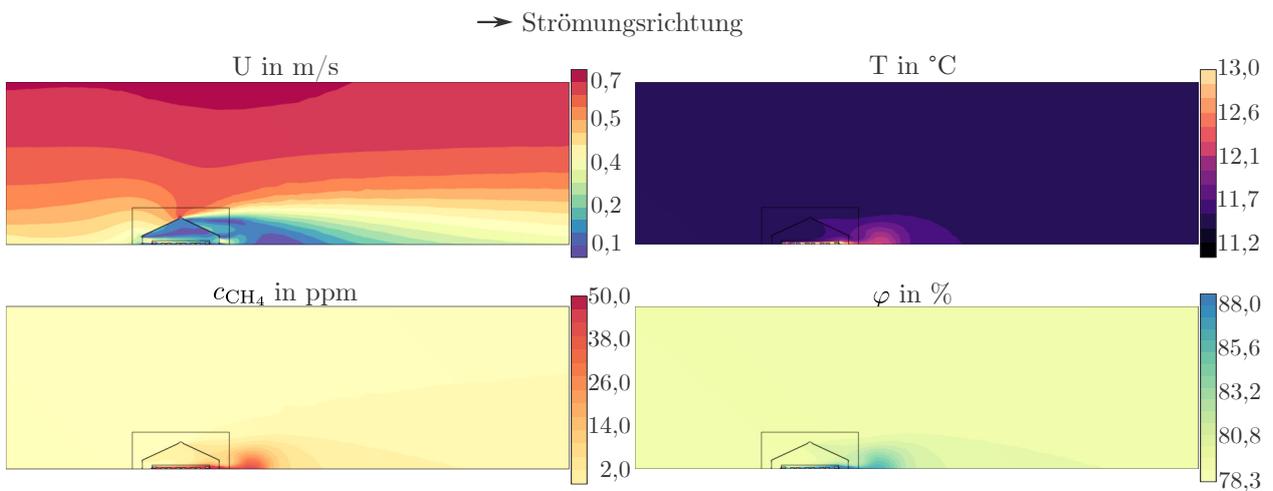


Abbildung D.60: CFD-Ergebnisse des Durchlaufes N10 auf der Schnittebene x - z , bei $y = 20,8$ m (eigene Darstellung)

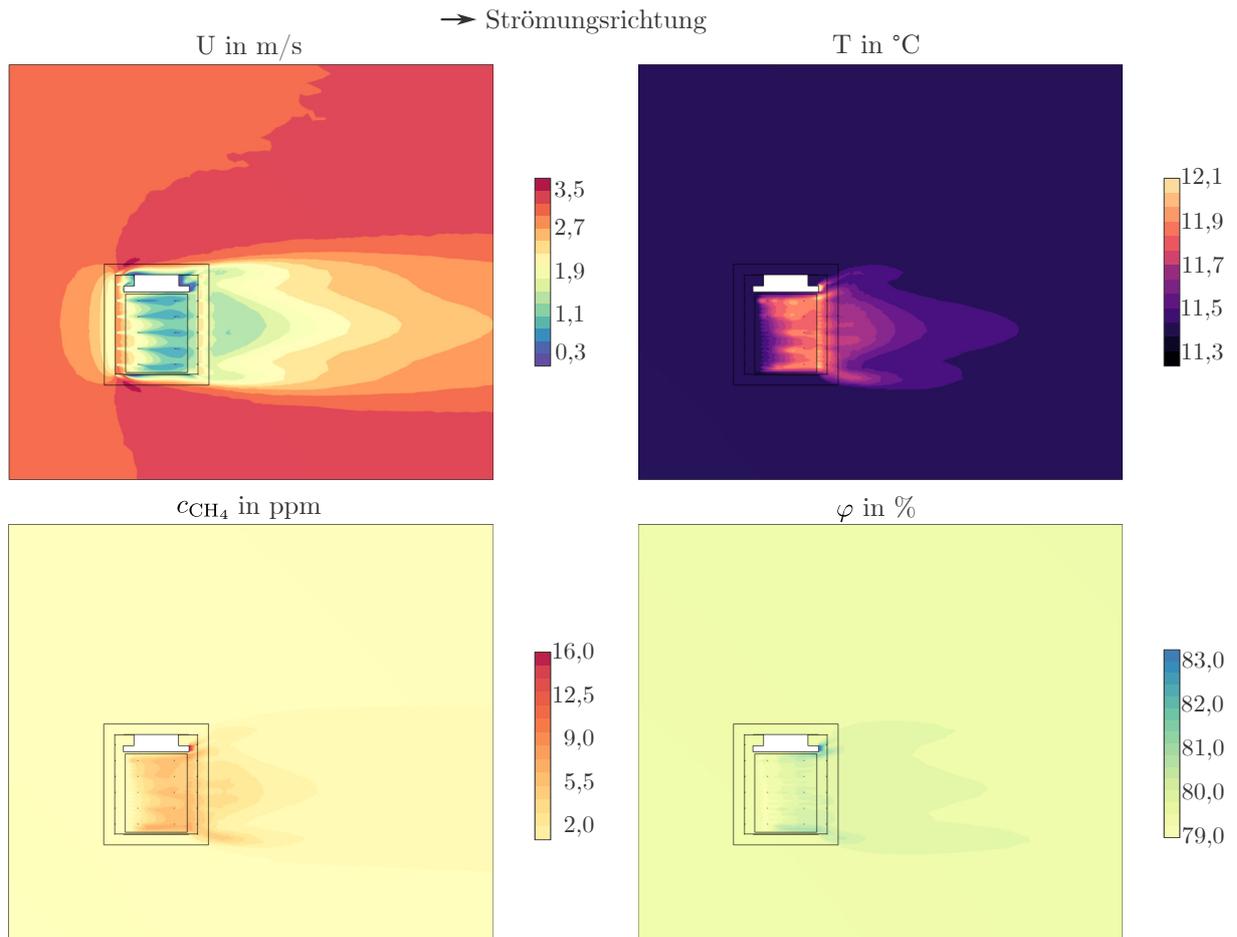


Abbildung D.61: CFD-Ergebnisse des Durchlaufes N11 auf der Schnittebene $x-y$, bei $z = 1,7$ m (eigene Darstellung)

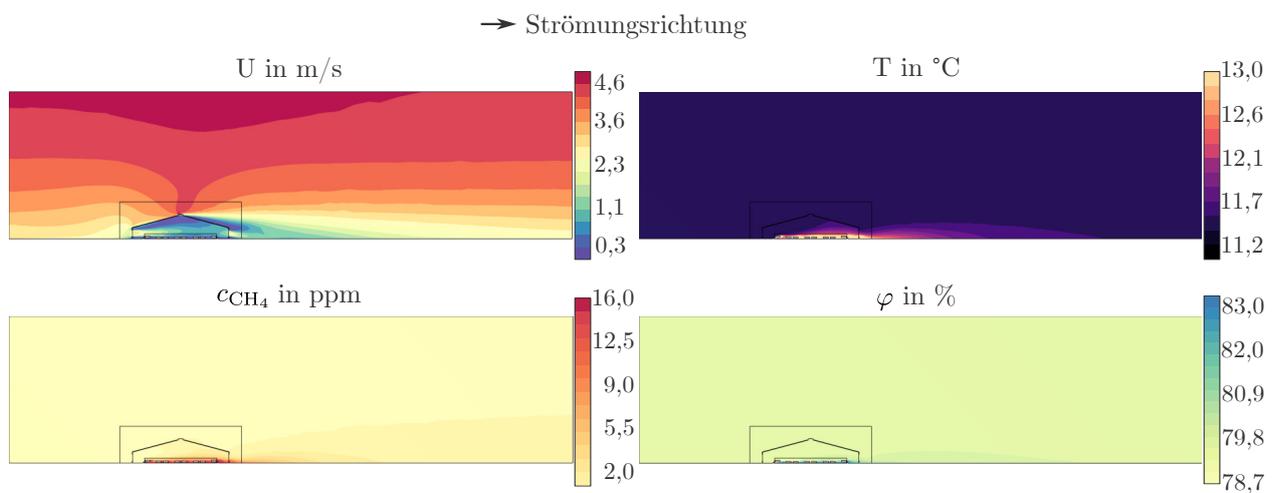


Abbildung D.62: CFD-Ergebnisse des Durchlaufes N11 auf der Schnittebene $x-z$, bei $y = 20,8$ m (eigene Darstellung)

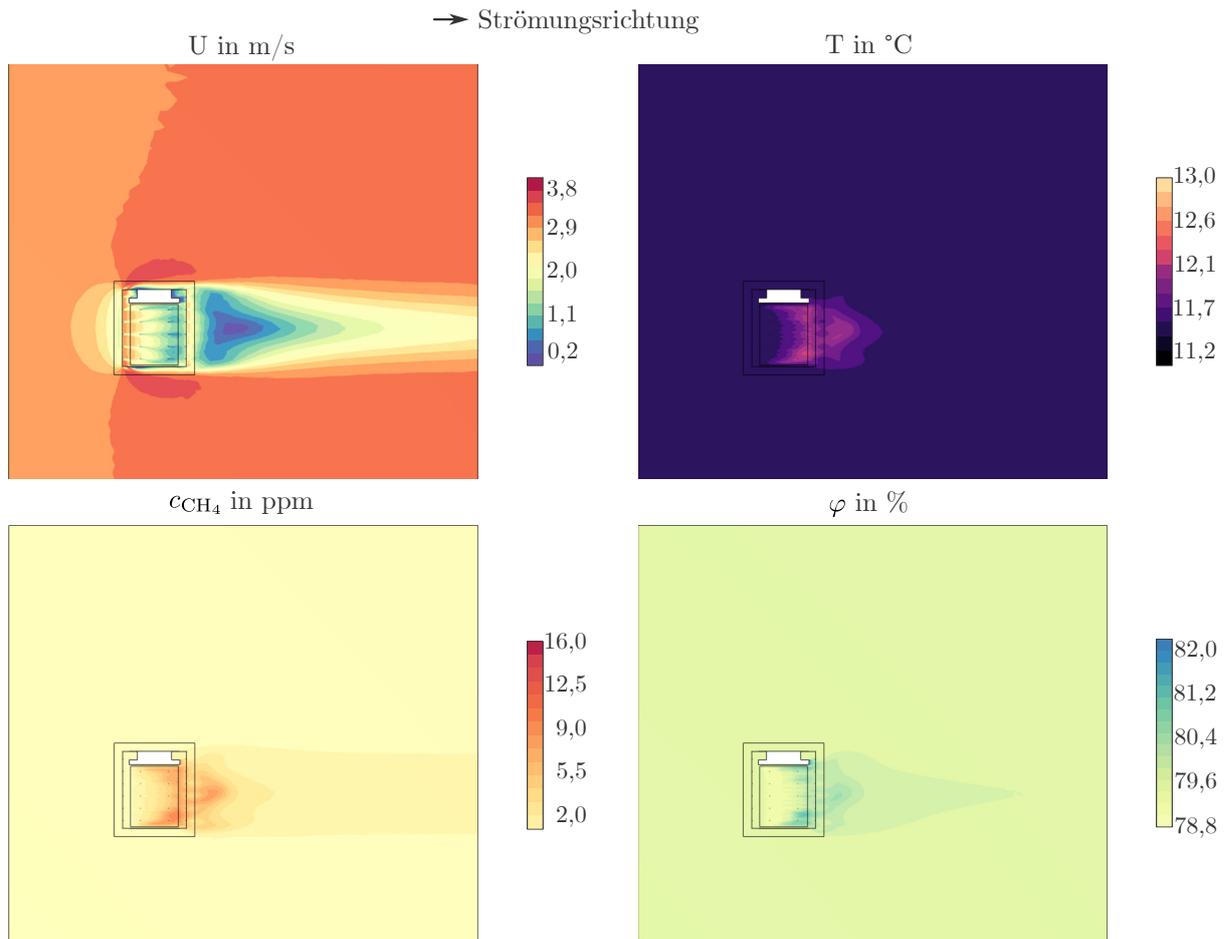


Abbildung D.63: CFD-Ergebnisse des Durchlaufes N12 auf der Schnittebene x - y , bei $z = 1,7$ m (eigene Darstellung)

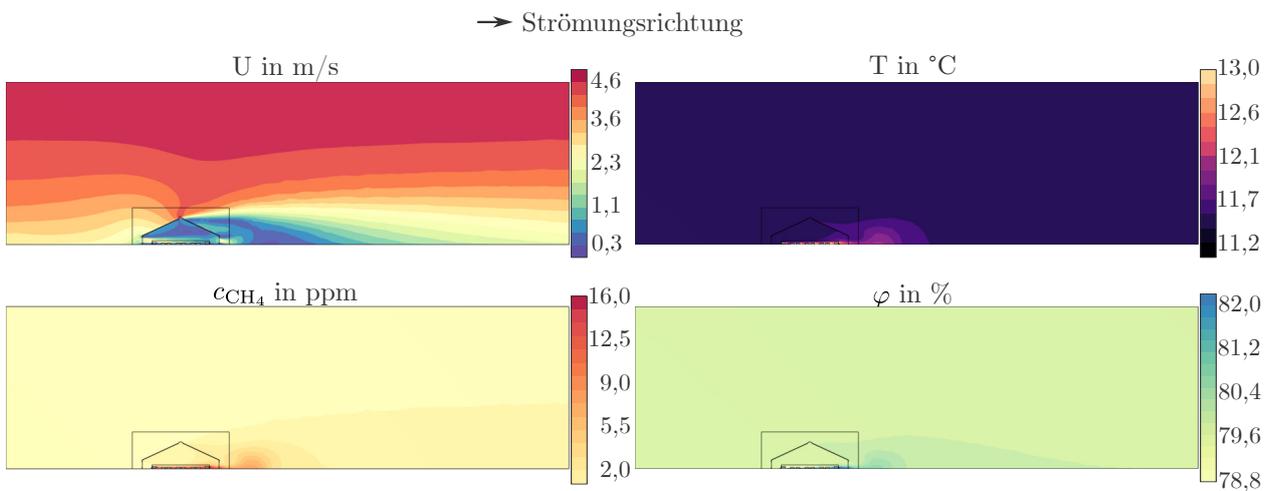


Abbildung D.64: CFD-Ergebnisse des Durchlaufes N12 auf der Schnittebene x - z , bei $y = 20,8$ m (eigene Darstellung)

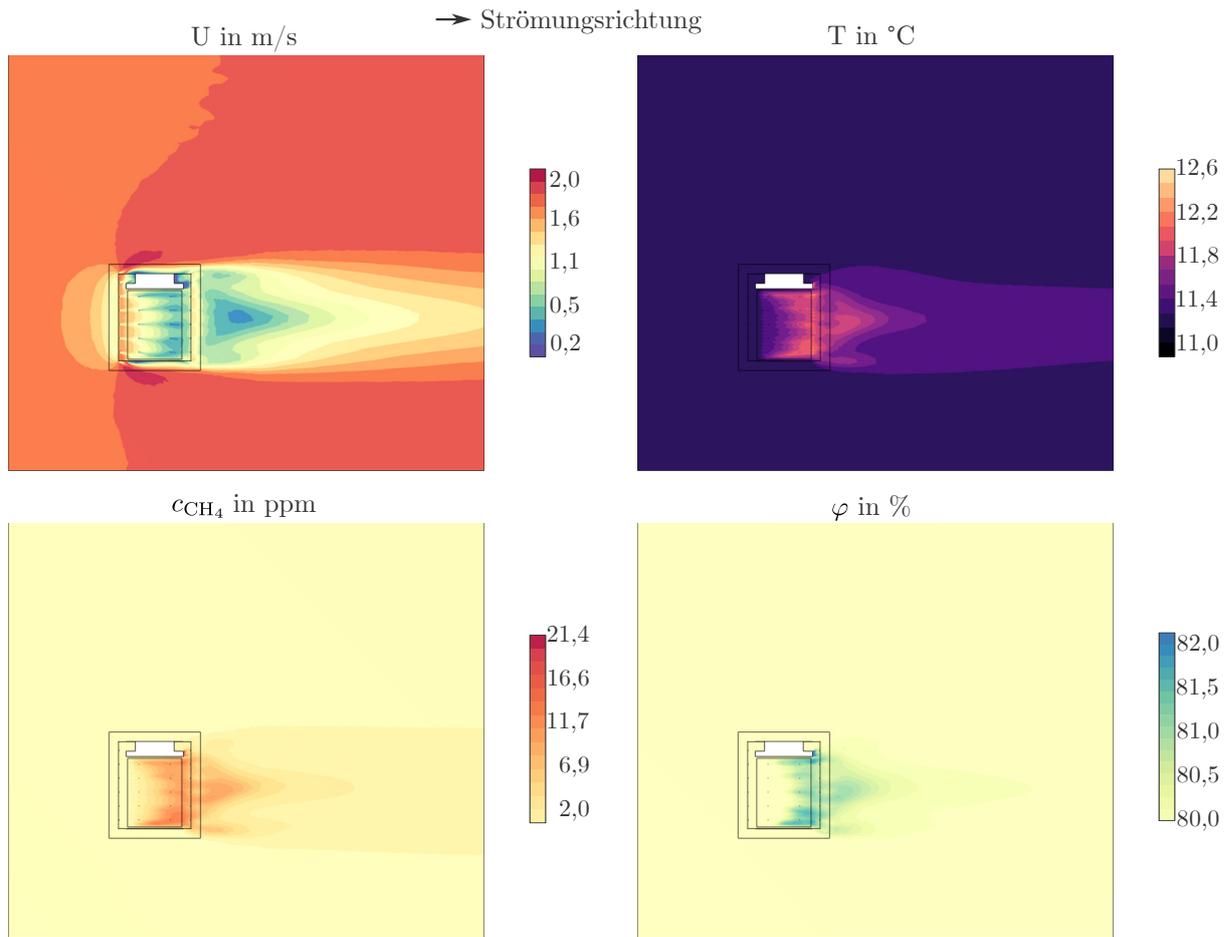


Abbildung D.65: CFD-Ergebnisse des Durchlaufes N13 auf der Schnittebene $x-y$, bei $z = 1,7$ m (eigene Darstellung)

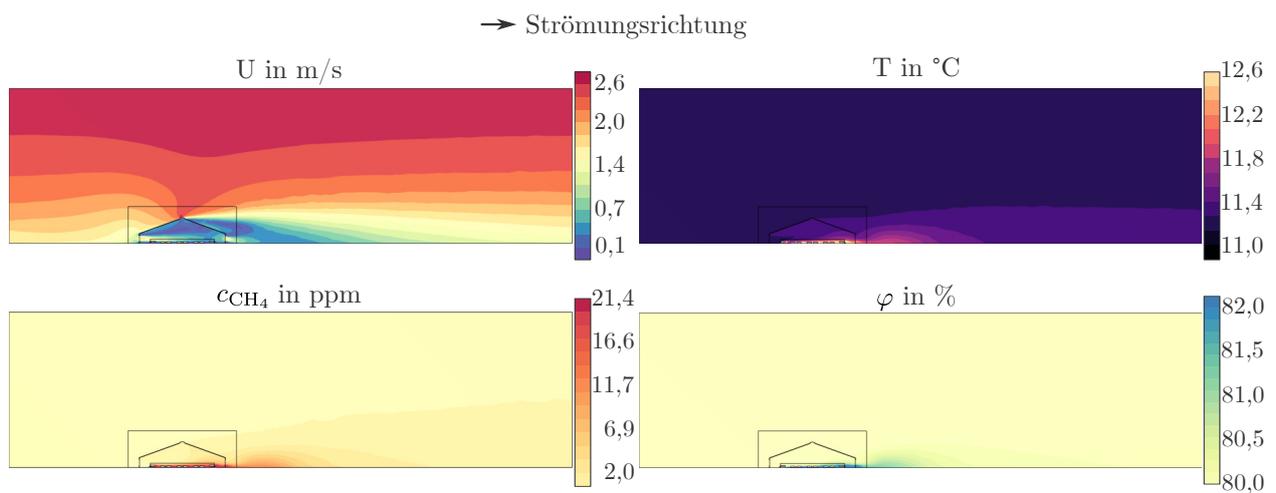


Abbildung D.66: CFD-Ergebnisse des Durchlaufes N13 auf der Schnittebene $x-z$, bei $y = 20,8$ m (eigene Darstellung)

Programmcode

E.1 IPCC Berechnung des Methans

Der folgende Python-Programmcode implementiert die in Kapitel B **TOP3** festgehaltenen Parameter. Die restlichen Werte und Berechnungen entstammen der Richtlinie des IPCC (EGGLESTON et al., 2006).

```
1 # ~~~~~
2 # Metadaten und Parameter
3 # ~~~~~
4 nt      = 120      # Anzahl Vieh
5 y_m    = 6.5      # Methankonversationsfaktor des Milchviehs
6 cfi    = 0.386    # Erhaltungsenergiefaktor (Tabelle s. 10.4)
7 bw     = 670     # Durchschnittliche Lebendmasse in kg
8 weight = nt*670   # Masse der Herde
9
10 # ~~~~~
11 # Berechnung der Energiemengen
12 # ~~~~~
13 # Erhaltungsenergie
14 ne_m = lambda cfi, weight: cfi * (weight)**0.75
15 calc_ne_m = ne_m(cfi, weight)
16
17 # Bewegungsenergie
18 ca = 0.00 # s. Tabelle 10.5
19 ne_a = lambda ca, ne_m: ca*ne_m
20 calc_ne_a = ne_a(ca, calc_ne_m)
```

```
21
22 # Wachstumsenergie
23 mw = bw
24 c = 0.8
25 wg = (1/3)*(50/365)
26 ne_g = lambda bw, c, mw, wg: 22.02*(bw/(c*wg))**(0.75)*wg**(1.097)
27 calc_ne_g = ne_g(bw, c, mw, wg)
28
29 # Laktationsenergie
30 milk = 30.0*120
31 fat = 4.2
32 ne_1 = lambda milk, fat: milk*(1.47 + 0.40*fat)
33 calc_ne_1 = ne_1(milk, fat)
34
35 # Energie der Schwangerschaft
36 c_pregnancy = 0.10 # s. Tabelle 10.7
37 c_preg_population = 0.93
38 ne_p = lambda c_preg_population, c_pregnancy, ne_m: \
39     c_preg_population*c_pregnancy*ne_m
40 calc_ne_p = ne_p(c_preg_population, c_pregnancy, calc_ne_m)
41
42 # Energiequotient
43 de = 65 # Verdauliche Energie in % der Bruttoenergiemenge s. Tab. 10.2
44 rem = lambda de: (1.123-(4.092e-3*de)+(1.126e-5 * de**2)-(25.4/de))
45 calc_rem = rem(de)
46
47 reg = lambda de: (1.164-(5.160e-3*de)+(1.308e-5 * de**2)-(37.4/de))
48 calc_reg = reg(de)
49
50 # Bruttoenergiemenge berechnen
51 ge_liket1 = 242
52 ge = lambda ne_m, ne_a, ne_1, ne_work, ne_p, ne_g, ne_wool, rem, reg, de: \
53     (((ne_m+ne_a+ne_1+ne_work+ne_p)/rem)+((ne_g+ne_wool)/reg))/(de
54     /100)
55 ge_calc = ge(calc_ne_m, calc_ne_a, calc_ne_1, 0, \
56     calc_ne_p, calc_ne_g, 0, calc_rem, calc_reg, de)
57 ge_single_cow = ge_calc/nt
58
59 # Emissionsfaktor berechnen (s. Gleichung 10.21 der IPCC-Richtlinie)
60 formula_ef_enteric_t2 = lambda ge, y_m: ((ge*(y_m/100)*365)/55.65)
61 ef_enteric_t2 = formula_ef_enteric_t2(ge_single_cow, y_m)
62
63 # Emissionen aus der enterischen Fermentation
64 emissions_manure_t2 = lambda ef, nt: ef*(nt/10e6)
65 calc_emissions_enteric_t2 = emissions_manure_t2(ef_enteric_t2, nt)
66
67 # Ausscheidungsrate berechnen, mit Standardwerten nach IPCC
68 vs = lambda ge, de, ue, assh: (ge*(1-(de/100))+(ue*ge))*((1-ash)/(18.45))
69 ue = 0.04
```

```

69 assh = 0.08
70 calc_vs = vs(ge_single_cow, de, ue, assh)
71
72 # Berechnen der Emissionen aus dem Lagersystem des Dungs
73 mcf = 19
74 b0_table_10A_4 = 0.24 # Westeuropa
75 f_ef_manure_t2 = lambda vs, bo, mcf, ms: (vs*365)*(bo*0.67*(mcf/100)*ms)
76 ms = 1
77 ef_manure_t2 = f_ef_manure_t2(calc_vs, b0_table_10A_4, mcf, ms)
78
79 # Umrechnen der Einheiten und Ausgabe
80 sec_per_year = 365*24*60*60
81 enteric_ch4_source = (ef_enteric_t2/sec_per_year)*nt
82 manure_ch4_source = (ef_manure_t2/sec_per_year)*nt
83 print(f'Enterische CH4 Emission in kg/s: {enteric_ch4_source:.2e}')
84 print(f'Dung bedingte CH4 Emission in kg/s: {manure_ch4_source:.2e}')

```

E.2 Programmcode PINN

```

1 # ~~~~~
2 # Import der Bibliotheken
3 # ~~~~~
4
5 from os import listdir
6 from os.path import isfile, join
7 import random
8 import numpy as np
9 import matplotlib.pyplot as plt
10 import matplotlib
11 from matplotlib import font_manager
12 from matplotlib import ticker
13 import deepxde as dde
14 from deepxde.backend import tf
15 import pandas as pd
16 from scipy.interpolate import griddata, splprep, splev
17
18 # Prüfen der Version und GPU-Unterstützung
19 print(tf.__version__)
20 print(tf.config.list_physical_devices('GPU'))
21
22
23
24 # ~~~~~
25 # Konstanten und Variablen anlegen
26 # ~~~~~
27
28 DEBUG = True
29

```

```

30 BUILDINGS = {'N1': {'H': 9.938, 'B': 39.1}, 'N2': {'H': 9.938, 'B':
31 39.1}, 'N3': {'H': 13.816, 'B': 39.1},\
          'N4': {'H': 13.816, 'B': 39.1}, 'N5': {'H': 11.815, 'B':
32 39.1}, 'N6': {'H': 11.815, 'B': 39.1},\
          'N7': {'H': 11.815, 'B': 39.1}, 'N8': {'H': 11.815, 'B':
33 39.1}, 'N9': {'H': 9.938, 'B': 39.1},\
          'N10': {'H': 13.816, 'B': 39.1}, 'N11': {'H': 9.938, 'B':
34 39.1}, 'N12': {'H': 13.816, 'B': 39.1},\
          'N13': {'H': 11.815, 'B': 39.1}}
35
36 # Trainingsparameter
37 NUM_BOUNDARY = 5200 # In Summe
38 NUM_SAMPLING = 400 # je Ebene --> 5200 In Summe
39 NUM_PDE = 400 # je Ebene --> 5200 In Summe
40 ACTIVATION = "tanh"
41 INITIALIZER = "Glorot uniform"
42 LOSS_WEIGHTS = [1, 1, 1, 1, 1, 1, 100, 100, 100]
43
44 # Netztopologie
45 INPUT_DIM = 5 # x, z, W, beta, v_m(10)
46 OUTPUT_DIM = 5 # u, w, p, u'_p, w'_p
47 HIDDEN_LAYER = 5 # Fünf verdeckte Schichten
48 NEURONS = 128 # 128 Neuronen je verdeckter Schicht
49 LAYER_SIZE = [INPUT_DIM] + [NEURONS] * HIDDEN_LAYER + [OUTPUT_DIM]
50
51 # Fluideigenschaften des künstlichen Fluids
52 params = {
53     'mu': 7.28821e-6,
54     'rho': 1000
55 }
56 params['nu'] = params['mu']/params['rho']
57
58 # Skalierungsvektor erstellen
59 SCALING_DIM = 100
60 SCALING_U = 3.5
61 SCALING_P = 1e2
62 div_arr = [SCALING_DIM, SCALING_DIM, 1, 1, SCALING_U, SCALING_U, SCALING_U
63 , SCALING_P]
64
65 # Dachneigungen in rad bereitstellen
66 b_15, b_20, b_25 = np.deg2rad(15), np.deg2rad(20), np.deg2rad(25)
67
68 # Faktoreinstellung der Versuchsdurchläufe skalieren
69 HYPERCUBE_COLS = { 'N1': [0.3, b_15, 2.0/SCALING_U], 'N2': [0.9, b_15,
70 2.0/SCALING_U], 'N3': [0.3, b_25, 2.0/SCALING_U],\
          'N4': [0.9, b_25, 2.0/SCALING_U], 'N5': [0.3, b_20,
          0.5/SCALING_U], 'N6': [0.9, b_20, 0.5/SCALING_U],\
          'N7': [0.3, b_20, 3.5/SCALING_U], 'N8': [0.9, b_20,
          3.5/SCALING_U], 'N9': [0.6, b_15, 0.5/SCALING_U],\

```

```

71         'N10': [0.6, b_25, 0.5/SCALING_U], 'N11': [0.6, b_15,
72         3.5/SCALING_U], 'N12': [0.6, b_25, 3.5/SCALING_U],\
73         'N13': [0.6, b_20, 2.0/SCALING_U],}
74 # Pfadangaben: Rechennetz und Daten aus der CFD sind einzelne Dateien
75 path = './data/cfd_results/'
76 LOADPATHS_CFD_DATA = [f'{path}{f}' for f in listdir(path) if isfile(
77     join(path, f))]
78 for f in LOADPATHS_CFD_DATA:
79     print(f)
80 path = './data/meshes/'
81 meshes = [f for f in listdir(path) if isfile(join(path, f))]
82
83 LOADPATHS_MESH_BOUNDARY = []
84 LOADPATHS_MESH_INNER = []
85 for m in meshes:
86     if 'boundary' in m:
87         LOADPATHS_MESH_BOUNDARY.append(f'{path}{m}')
88     elif 'inner' in m:
89         LOADPATHS_MESH_INNER.append(f'{path}{m}')
90
91 MODEL_PATH = './models/model'
92
93
94
95 # ~~~~~
96 # Hilfsfunktionen bereitstellen
97 # ~~~~~
98
99 def filter_arr(arr, index=0, limits=0, tol=1e-5):
100     """
101     Filtert ein Datenarray gemäß Limits
102     limits: float, int etc. oder [min, max]
103     """
104     if type(limits) is int or type(limits) is float:
105         return arr[np.where(np.isclose(arr[:, index], limits, atol=tol))]
106     elif len(limits) == 2:
107         return arr[np.where((arr[:, index] > limits[0]) & (arr[:, index] <
108             limits[1]))]
109
110 def drop_duplicates(arr1, arr2, debug=False):
111     """
112     Die Funktion stapelt die Matrizen und entfernt alle Mehrfach-Einträge
113     arr1: Erstes Array
114     arr2: Zweites Array
115     debug: Bool --> Zusatzausgaben
116     """
117     combined_data = np.vstack((arr1, arr2))

```

```
117     filtered_data = pd.DataFrame(combined_data.T).T.drop_duplicates(keep=
118     False).to_numpy()
119
120     if debug:
121         print(f'Drop duplicates from:')
122         print(f'arr1 with shape {arr1.shape} and arr2 with shape {arr2.
123         shape}')
124         print(f'Results in shape: abs({arr1.shape[0]} - {arr2.shape[0]}) =
125         {filtered_data.shape[0]}')
126     return filtered_data
127
128 def select_random_rows(arr, num_rows):
129     """
130     Die Funktion gibt eine randomisierte Auswahl
131     an Zeilen einer Matrix zurück
132     """
133     indices = list(range(len(arr)))
134     random.shuffle(indices)
135     selected_indices = indices[:num_rows]
136     selected_rows = [arr[i] for i in selected_indices]
137     return selected_rows, selected_indices
138
139 # ~~~~~
140 # Laden und aufbereiten der Daten
141 # ~~~~~
142 matplotlib.rcParams['font.size'] = 8
143 matplotlib.rcParams['figure.dpi'] = 96
144
145 CFD_data      = []
146 boundary_mesh = []
147 volume_mesh   = []
148 inlet         = []
149 outlet        = []
150 top           = []
151
152 for i in range(0, len(LOADPATHS_CFD_DATA)):
153     # Zeichenkette des jeweiligen Durchlaufs lesen (z. B. 'N1')
154     filename = LOADPATHS_CFD_DATA[i].split('/')[ -1 ].split('_')[ 0 ]
155
156     # Laden der Daten
157     CFD_data.append(np.divide(np.load(LOADPATHS_CFD_DATA[i], allow_pickle=
158     True), div_arr)) # x, y, u, v, p
159     boundary_mesh.append(np.divide(np.load(LOADPATHS_MESH_BOUNDARY[i],
160     allow_pickle=True), np.tile(SCALING_DIM, 2)))
161     volume_mesh.append(np.divide(np.load(LOADPATHS_MESH_INNER[i],
162     allow_pickle=True), np.tile(SCALING_DIM, 2)))
```

```

160 # Entferne äußere Knoten aus dem Rechenetz, indem die Netze
zusammengesetzt und Doppeleinträge entfernt werden
161 volume_mesh[-1] = drop_duplicates(boundary_mesh[-1], volume_mesh[-1])
162 plt.scatter(volume_mesh[-1][:,0], volume_mesh[-1][:,1], s=0.5, color='
0.8', label='volume')
163
164 # Separieren der Gleitwände und des Einströmrand
165 x_in = (-BUILDINGS[filename]['B']/2-5*BUILDINGS[filename]['H'])/
SCALING_DIM
166 x_out = (BUILDINGS[filename]['B']*(3/2)+10*BUILDINGS[filename]['H'])/
SCALING_DIM
167 y_top = (BUILDINGS[filename]['H']+5*BUILDINGS[filename]['H'])/
SCALING_DIM
168 inlet.append(filter_arr(boundary_mesh[-1], limits=x_in, tol=1e-3))
169 outlet.append(filter_arr(boundary_mesh[-1], limits=x_out, tol=1e-3))
170 top.append(filter_arr(boundary_mesh[-1], index=1, limits=y_top, tol=1e
-3))
171 boundary_mesh[-1] = drop_duplicates(boundary_mesh[-1], np.vstack((
inlet[-1], outlet[-1], top[-1])))
172
173 # Ergänze Hypercube-Koordinaten für Randbedingungen
174 w, beta, v_b = HYPERCUBE_COLS[filename]
175 boundary_mesh[-1] = np.insert(boundary_mesh[-1], 2, w, axis=1)
176 boundary_mesh[-1] = np.insert(boundary_mesh[-1], 3, beta, axis=1)
177 boundary_mesh[-1] = np.insert(boundary_mesh[-1], 4, v_b, axis=1)
178 volume_mesh[-1] = np.insert(volume_mesh[-1], 2, w, axis=1)
179 volume_mesh[-1] = np.insert(volume_mesh[-1], 3, beta, axis=1)
180 volume_mesh[-1] = np.insert(volume_mesh[-1], 4, v_b, axis=1)
181 top[-1] = np.insert(top[-1], 2, w, axis=1)
182 top[-1] = np.insert(top[-1], 3, beta, axis=1)
183 top[-1] = np.insert(top[-1], 4, v_b, axis=1)
184
185 plt.scatter(inlet[-1][:,0], inlet[-1][:,1], s=1, label='inlet')
186 plt.scatter(outlet[-1][:,0], outlet[-1][:,1], s=1, label='outlet')
187 plt.scatter(top[-1][:,0], top[-1][:,1], s=1, label='top')
188 plt.scatter(boundary_mesh[-1][:,0], boundary_mesh[-1][:,1], s=1, label
='boundaries')
189 plt.legend(bbox_to_anchor=(1.1, 1.00))
190 plt.axis('image')
191 plt.show()
192
193 boundary_mesh = np.vstack(boundary_mesh)
194 top = np.vstack(top)
195
196
197 # ~~~~~
198 # Randomisiertes Datenset für das Training
199 # ~~~~~
200 temp_data_CFD = []
201 temp_data_PDE = []

```

```

202
203 for plane in range(0, len(CFD_data)):
204     # Aus den CFD-Daten -> Punktrandbedingung
205     arr, _ = select_random_rows(CFD_data[plane], NUM_SAMPLING)
206     temp_data_CFD.append(np.array(arr))
207
208     # Aus den Gitterpunkten -> PDE Residuum und anfügen Hypercube-
    Koordinaten
209     arr, _ = select_random_rows(volume_mesh[plane], NUM_PDE)
210     temp_data_PDE.append(np.array(arr))
211
212     if DEBUG:
213         plt.scatter(temp_data_CFD[-1][:,0], temp_data_CFD[-1][:,1], s=1,
    color='0.8')
214         plt.axis('image')
215         plt.title('CFD Punkte')
216         plt.show()
217
218         plt.scatter(temp_data_PDE[-1][:,0], temp_data_PDE[-1][:,1], s=1,
    color='0.8')
219         plt.axis('image')
220         plt.title('PDE Punkte')
221         plt.show()
222
223 training_data_CFD = np.vstack(temp_data_CFD)
224 training_data_CFD = training_data_CFD[~np.isnan(training_data_CFD[:,5])] #
    entferne NaN-Werte
225 training_data_PDE = np.vstack(temp_data_PDE)
226
227
228
229 # ~~~~~
230 # Partielle Differentialgleichungen für
231 # zusammengesetzte Verlustfunktion anlegen
232 # ~~~~~
233 def pdepr(x, u):
234     u_vel, v_vel, p, u_t, v_t = u[:, 0:1], u[:, 1:2], u[:, 2:3], u[:,
    3:4], u[:, 4:5]
235
236     # Approximiere Ableitungen via 'Backpropagation'
237     du_x = dde.grad.jacobian(u, x, i=0, j=0)
238     du_y = dde.grad.jacobian(u, x, i=0, j=1)
239     du_xx = dde.grad.hessian(u, x, component=0, i=0, j=0)
240     du_yy = dde.grad.hessian(u, x, component=0, i=1, j=1)
241
242     dv_x = dde.grad.jacobian(u, x, i=1, j=0)
243     dv_y = dde.grad.jacobian(u, x, i=1, j=1)
244     dv_xx = dde.grad.hessian(u, x, component=1, i=0, j=0)
245     dv_yy = dde.grad.hessian(u, x, component=1, i=1, j=1)
246

```

```

247     dp_x = dde.grad.jacobian(u, x, i=2, j=0)
248     dp_y = dde.grad.jacobian(u, x, i=2, j=1)
249
250     dut_x = dde.grad.jacobian(u, x, i=3, j=0)
251     dut_y = dde.grad.jacobian(u, x, i=3, j=1)
252     dvt_x = dde.grad.jacobian(u, x, i=4, j=0)
253     dvt_y = dde.grad.jacobian(u, x, i=4, j=1)
254
255     # Berechne Verlustfunktionen
256     f_u = (u_vel*du_x + v_vel*du_y) + (u_t*dut_x + v_t*dut_y) + 1/1000*
dp_x - (7.28821e-6/1000)*(du_xx + du_yy)
257     f_v = (u_vel*dv_x + v_vel*dv_y) + (u_t*dvt_x + v_t*dvt_y) + 1/1000*
dp_y - (7.28821e-6/1000)*(dv_xx + dv_yy)
258     cont = du_x + dv_y
259
260     return [f_u, f_v, cont]
261
262
263
264 # ~~~~~
265 # Hyperwürfel anlegen
266 # ~~~~~
267 # x, z, W, beta, v_m(10)
268 geom = dde.geometry.geometry_nd.Hypercube(xmin=[x_in, 0, 0.3, np.deg2rad
(15), 0.5/3.5], xmax=[x_out, y_top, 0.9, np.deg2rad(25), 3.5/3.5])
269
270
271
272 # ~~~~~
273 # Randbedingungen erstellen
274 # ~~~~~
275 boundary_sample, _ = select_random_rows(boundary_mesh, NUM_BOUNDARY)
276 boundary_sample = np.array(boundary_sample)
277
278 # Gleitwand oben
279 bc_upper_v = dde.PointSetBC(top, np.vstack(np.zeros(top.shape[0])),
component=1)
280
281 # Reibungsbehaftete Wände
282 bc_wall_u = dde.PointSetBC(boundary_sample, np.vstack(np.zeros(
boundary_sample.shape[0])), component=0)
283 bc_wall_v = dde.PointSetBC(boundary_sample, np.vstack(np.zeros(
boundary_sample.shape[0])), component=1)
284
285 # Verwenden der Simulationswerte als Punktrandbedingung
286 bc_mess_u = dde.PointSetBC(training_data_CFD[:, 0:5], np.vstack(
training_data_CFD[:, 5]), component=0)
287 bc_mess_v = dde.PointSetBC(training_data_CFD[:, 0:5], np.vstack(
training_data_CFD[:, 6]), component=1)

```

```

288 bc_mess_p = dde.PointSetBC(training_data_CFD[:, 0:5], np.vstack(
      training_data_CFD[:, 7]), component=2)
289
290 boundaries = [bc_upper_v, bc_wall_u, bc_wall_v, bc_mess_u, bc_mess_v,
      bc_mess_p]
291
292
293
294 # ~~~~~
295 # Initialisieren
296 # ~~~~~
297 # Keine Punkte in der Domäne, da diese mittels punktweiser RB und
      Ankerpunkten erstellt wurden
298 data = dde.data.PDE(geom, pdepr, boundaries, num_domain=0, num_boundary=
      NUM_BOUNDARY, anchors=training_data_PDE, num_test=1024)
299 net = dde.maps.FNN(LAYER_SIZE, ACTIVATION, INITIALIZER)
300
301
302
303 # ~~~~~
304 # Trainieren und sichern der Metriken
305 # ~~~~~
306 model = dde.Model(data, net)
307 model.compile("adam", lr=1e-3, loss_weights=LOSS_WEIGHTS)
308 losshistory, train_state = model.train(epochs=10000)
309
310 model.compile("adam", lr=1e-4, loss_weights=LOSS_WEIGHTS)
311 losshistory, train_state = model.train(epochs=20000)
312
313 model.compile("L-BFGS-B", loss_weights=LOSS_WEIGHTS)
314 losshistory, train_state = model.train(model_save_path=MODEL_PATH)
315
316 dde.saveplot(losshistory, train_state, issave=True, isplot=False)
317
318
319
320 # ~~~~~
321 # Plotten des Ergebnisses
322 # ~~~~~
323 # Hilfsfunktion für das Konturdiagramm
324 def array_to_meshgrid(arr, resolution_per_axis = 100):
325     x, y, u = arr.T
326     xi = np.linspace(np.min(x), np.max(x), resolution_per_axis)
327     yi = np.linspace(np.min(y), np.max(y), resolution_per_axis)
328     ui = griddata((x, y), u, (xi[None, :], yi[:, None]), method='linear')
329     X, Y = np.meshgrid(xi, yi)
330     return X, Y, ui
331
332 def contourplot(coords, ref_data, pred_data, crop=None, filepath=False,
      plot=True):

```

```

333     # Setze die Größe des Konturdiagramms und der Farbskala je nach
Ausschnitt
334     if crop != None:
335         x1, x2, y1, y2 = crop[0], crop[1], crop[2], crop[3]
336         dx, dy = np.abs(x1-x2), np.abs(y1-y2)
337         cropped_ratio = dx/dy
338         height = int(14/cropped_ratio)
339         figsize = (16, height)
340         clb_frac = 0.08*(1/cropped_ratio)
341     else:
342         clb_frac = 0.02
343         figsize = (16,14)
344
345     labels = [['u in m/s', '$\tilde{u}$ in m/s', '|u - $\tilde{u}$| in m
/s'],\
346              ['v in m/s', '$\tilde{v}$ in m/s', '|v - $\tilde{v}$| in m
/s'],\
347              ['p in Pa', '$\tilde{p}$ in Pa', '|p - $\tilde{p}$| in
Pa']]
348
349     # Limits und Anzahl der Level für u, v und p: [cfd, pinn, residual]
350     min_u, max_u = np.round(np.min(ref_data[:,0]),2), np.round(np.max(
ref_data[:,0]),2)
351     perc_err_u = 0.2*2.75
352     min_v, max_v = np.round(np.min(ref_data[:,1]),2), np.round(np.max(
ref_data[:,1]),2)
353     perc_err_v = 0.2*2.75
354     min_p, max_p = np.round(np.min(ref_data[:,2]),3), np.round(np.max(
ref_data[:,2]),3)
355     perc_err_p = 0.2*(max_p-min_p)
356     levels = [[np.linspace(min_u, max_u, 20), np.linspace(min_u, max_u,
20), np.linspace(0, perc_err_u, 20)],\
357              [np.linspace(min_v, max_v, 20), np.linspace(min_v, max_v,
20), np.linspace(0, perc_err_v, 20)],\
358              [np.linspace(min_p, max_p, 20), np.linspace(min_p, max_p,
20), np.linspace(0, perc_err_p, 20)]]
359
360     # Farbgebung
361     cmaps = np.tile(['Spectral_r', 'Spectral_r', 'Spectral_r'],(3,1))
362     cb_loc = 'bottom'
363
364     # Interpolationsauflösung
365     r = 500
366
367     # Setze Ergebnisse als Gitter zusammen
368     num_data = len(ref_data[:,1])
369     Z = [[], [], []]
370     for i, _ in enumerate(Z):
371         a = np.hstack((coords, ref_data[:,i].reshape((num_data, 1))))
372         x, y, u_ref = array_to_meshgrid(a, resolution_per_axis = r)

```

```
373
374     a = np.hstack((coords, pred_data[:,i].reshape((num_data, 1))))
375     _, _, u_pred = array_to_meshgrid(a, resolution_per_axis = r)
376
377     res = np.abs(u_ref - u_pred)
378
379     Z[i] = [u_ref, u_pred, res]
380
381     # Erstelle Subplot mit 3 Spalten und 3 Zeilen
382     fig, axs = plt.subplots(3, 3, constrained_layout=True, sharex='col',
383                             sharey='row', dpi=300)
384
385     # Iteriere über alle Zeilen
386     for i in range(0, 3):
387         # Iteriere über alle Spalten
388         for j in range(0, 3):
389             # Lege Konturdiagramm an
390             cm = axs[i, j].contourf(x, y, Z[i][j], cmap=cmaps[i][j],
391                                     levels=levels[i][j], extend='both')
392             # Modifiziere Farbskala
393             clb = plt.colorbar(cm, ax=axs[i, j], fraction=clb_frac, pad
394                             =0.02, location=cb_loc)
395             clb.set_ticks(np.linspace(np.min(levels[i][j]), np.max(levels[
396 i][j]), 3))
397             if j == 2:
398                 clb.ax.yaxis.set_major_formatter(ticker.FormatStrFormatter
399 ('%.2e'))
400             else:
401                 clb.ax.yaxis.set_major_formatter(ticker.FormatStrFormatter
402 ('%.2f'))
403
404             # Setze Beschriftung
405             axs[i, j].set_title(labels[i][j])
406             if crop != None:
407                 axs[i, j].set_xlim(x1, x2)
408                 axs[i, j].set_ylim(y1, y2)
409                 axs[i, j].set_aspect('equal')
410             else:
411                 axs[i, j].axis('image')
412
413             if filepath:
414                 plt.savefig(filepath, dpi=300)
415             if plot:
416                 plt.show()
417
418 matplotlib.rcParams['font.size'] = 11
419 matplotlib.rcParams['figure.dpi'] = 300
420
421 # Entnehme randomisierte Punkte aus CFD-Daten nach Index
422 test_ind = 2
423 test_arr, _ = select_random_rows(CFD_data[test_ind], 8000)
```

```

417 test_arr = np.array(test_arr)
418 pinn_data = model.predict(test_arr[:,0:5])
419
420 # Reskalieren der Ergebnisse
421 pinn_div_arr = np.array(div_arr)[5:]
422 test_arr = np.divide(test_arr, 1/np.array(div_arr))
423 pinn_data = np.divide(pinn_data[:, 0:3], 1/pinn_div_arr) # nur u, v, p
424
425 filepath = 'PINN_Konturdiagramm_Referenzebene.svg'
426
427 contourplot(test_arr[:, 0:2], test_arr[:, 5:], pinn_data, crop=[-50, 50,
    0.0, 50], filepath=filepath)

```

E.3 Programmcode Webapplikation

```

1 import os
2 import jwt
3 import datetime
4 import plotly
5 import plotly.express as px
6 import plotly.graph_objects as grob
7 import numpy as np
8 import pandas as pd
9 import json
10 from bson.objectid import ObjectId
11 from functools import wraps
12 from flask import Flask, render_template, request, jsonify, redirect
13 from flask_mongoengine import MongoEngine
14 from pymongo import MongoClient
15 from flask_security import Security, MongoEngineUserDatastore, UserMixin,
    RoleMixin, auth_required, hash_password, current_user
16
17 # neural nets
18 import deepxde as dde
19 from deepxde.backend import tf
20
21 # Eigene ausgelagerte Funktionen
22 from api.pinn import *
23 from api.prediction import *
24
25
26
27 # ~~~~~
28 # Setup Flask
29 # ~~~~~
30 # Initialize Flask app
31 app = Flask(__name__)
32 app.config['DEBUG'] = True
33

```

```
34 # Generiere einen sicheren key via secrets.token_urlsafe()
35 app.config['SECRET_KEY'] = os.environ.get("SECRET_KEY", '
    qiguWW2_rJhiHZJgHfvcb-hAp8IACVFZ8FVS_ANOMiQ')
36 # Bcrypt ist der Standard Hashing-Algorithmus für SECURITY_PASSWORD_HASH,
    wofür ein Salt benötigt wird
37 # Generiere ein sicheres Salt via : secrets.SystemRandom().getrandbits
    (128)
38 app.config['SECURITY_PASSWORD_SALT'] = os.environ.get("
    SECURITY_PASSWORD_SALT", '150346766491128376770556107980472356765')
39
40 # Wenn Registrierung erlaubt sein soll
41 app.config['SECURITY_REGISTERABLE'] = False
42 # app.config['SECURITY_REGISTER_URL'] = '/create_account' # definiere
    eigene Registrierungsseite
43
44 # Eigenes Design für die Login-Seite
45 app.config['SECURITY_LOGIN_USER_TEMPLATE'] = 'security/login_user.html'
46
47 # MongoDB konfigurieren
48 app.config['MONGODB_DB'] = 'account_management'
49 app.config['MONGODB_HOST'] = 'localhost'
50 app.config['MONGODB_PORT'] = 27017
51 app.config["MONGO_URI"] = f"mongodb://{app.config['MONGODB_HOST']}:{app.
    config['MONGODB_PORT']}/digital_twin"
52
53
54
55 # ~~~~~
56 # Initialisiere 'model' vor Nutzung
57 # (Für mehrere Nutzer via Session realisieren)
58 # ~~~~~
59 @app.before_first_request
60 def load_model_to_app():
61     app.model = load_pinn('46439')
62
63
64
65 # ~~~~~
66 # Stelle https Verbindung für Online Dienste sicher
67 # ~~~~~
68 @app.before_request
69 def before_request():
70     scheme = request.headers.get('X-Forwarded-Proto')
71     if scheme and scheme == 'http' and request.url.startswith('http://'):
72         url = request.url.replace('http://', 'https://', 1)
73         code = 301
74         return redirect(url, code=code)
75
76
77
```

```

78 # ~~~~~
79 # Definiere Datenstruktur für Datenbank-Objekte
80 # ~~~~~
81 # Erstelle Verbindungsobjekt zur Datenbank
82 db = MongoEngine(app)
83
84 class Role(db.Document, RoleMixin):
85     name = db.StringField(max_length=80, unique=True)
86     description = db.StringField(max_length=255)
87     permissions = db.StringField(max_length=255)
88
89 class User(db.Document, UserMixin):
90     email = db.StringField(max_length=255, unique=True)
91     password = db.StringField(max_length=255)
92     active = db.BooleanField(default=True)
93     fs_uniquifier = db.StringField(max_length=64, unique=True)
94     confirmed_at = db.DateTimeField()
95     roles = db.ListField(db.ReferenceField(Role), default=[])
96
97 class Token(db.Document):
98     owner_id = db.StringField(max_length=255, unique=True)
99     token = db.StringField(max_length=255)
100
101 # Konfiguriere Flask-Security für die Nutzerdatenbank
102 user_datastore = MongoEngineUserDatastore(db, User, Role)
103 app.security = Security(app, user_datastore)
104
105 # Verbinde zur Datenbank der digitalen Zwillinge via pymongo für Zugriffe
    im JSON Format
106 client = MongoClient(app.config["MONGO_URI"])
107 db_digital_twin = client.digital_twin
108 db_account_management = client.account_management
109
110
111
112 # ~~~~~
113 # Erstelle 'Decorator' für API Zugriff via JSON Web Token
114 # ~~~~~
115 def token_required(func):
116     @wraps(func)
117     def decorated(*args, **kwargs):
118         token = request.args.get('token')
119         if not token:
120             return jsonify({'message': 'Missing Token'}), 403
121         try:
122             data = jwt.decode(token, app.config['SECRET_KEY'], algorithms
    =["HS256"])
123         except:
124             return jsonify({'message': 'Invalid Token'}), 403
125     return func(*args, **kwargs)

```

```
126     return decorated
127
128
129
130 # ~~~~~
131 # Erstelle Testnutzer, falls nicht vorhanden
132 # Der Code und Nutzer muss vor Release entfernt werden!
133 # ~~~~~
134 with app.app_context():
135     if not app.security.datastore.find_user(email="test@me.com"):
136         app.security.datastore.create_user(email="test@me.com", password=
            hash_password("password"))
137
138
139
140 # ~~~~~
141 # Index Seite
142 # ~~~~~
143 @app.route("/", methods=['GET'])
144 @auth_required()
145 def home():
146     return render_template('index.html')
147
148
149
150 # ~~~~~
151 # Info Seite
152 # ~~~~~
153 @app.route('/info', methods=['GET'])
154 @auth_required()
155 def info():
156     return render_template('info.html')
157
158
159
160 # ~~~~~
161 # Simulation Seite
162 # ~~~~~
163 @app.route('/pinn', methods=['GET', 'POST'])
164 @auth_required()
165 def pinn():
166     # Konturdiagramm via Plotly: https://plotly.com/python/contour-plots/
167     # Leerer Plot bei Aufruf der Seite
168     contour_plot = grob.Figure(data = grob.Contour())
169     graphJSON = json.dumps(contour_plot, cls=plotly.utils.
        PlotlyJSONEncoder)
170
171     model = app.model # Bei mehreren Nutzern via Sessiondaten realisieren
172
173     if request.method=='POST':
```

```

174     request_data = request.get_json(force=True)
175
176     x_range, y_range, X, Y = limits2grid([-1, 1], [0, 1], 300)
177     pinn_coords = grid2coords(X, Y)
178
179     # Ergänze Koordinaten aus Frontend inkl. Skalierung für PINN
180     W_vec = np.vstack(np.tile(float(request_data['W'])/100,
pinn_coords.shape[0]))
181     beta_vec = np.vstack(np.tile(np.deg2rad(float(request_data['beta'
182 ])), pinn_coords.shape[0]))
182     v_m_vec = np.vstack(np.tile(float(request_data['v_m'])/3.5,
pinn_coords.shape[0]))
183     pinn_coords = np.hstack((pinn_coords, W_vec, beta_vec, v_m_vec))
184
185     pred_data = model.predict(pinn_coords)
186     pred_data = pred_data[:, 0]*3.5
187
188     Z = vec2grid(X, Y, pred_data)
189     min_pred = np.min(pred_data)
190     max_pred = np.max(pred_data)
191     d_pred = max_pred-min_pred
192     contour_plot = grob.Figure(data = grob.Contour(x=x_range, y=
y_range, z=Z, colorscale='spectral_r', \
193                                     colorbar=dict(
thickness=20, thicknessmode='pixels', len=0.9, lenmode='fraction',
outlinewidth=1, title='Geschw. in m/s'), \
194                                     contours=dict(start=
min_pred, end=max_pred, size=d_pred/20,), line_width=0,))
195     contour_plot = draw_barn(np.max(W_vec), np.max(beta_vec),
contour_plot)
196     graphJSON = json.dumps(contour_plot, cls=plotly.utils.
PlotlyJSONEncoder)
197     return jsonify(graphJSON)
198 else:
199     return render_template('pinn.html', graphJSON=graphJSON)
200
201
202
203 # ~~~~~
204 # Datenbank
205 # ~~~~~
206 @app.route('/data', methods=['GET', 'POST'])
207 @auth_required()
208 def data():
209     # Lade alle Sensoren und Daten und überführe in Arrays für das
Frontend
210     sensor_arr = []
211     quantities_arr = []
212     db_sensors = db_digital_twin.sensors.find({'owner_id': current_user.id
})

```

```
213     for item in list(db_sensors):
214         sensor_arr.append(item["sensor"])
215         data_columns = json.loads(item["data"])["columns"]
216         data_columns.remove("datetime")
217         quantities_arr.append(data_columns)
218
219     # Plotten
220     if request.method=='POST':
221         request_data = request.get_json(force=True)
222         data = db_digital_twin.sensors.find_one({'owner_id': current_user.
id, 'sensor': request_data['sensor']})
223         df = pd.read_json(data['data'], orient="split")
224
225         # Prüfen ob Mittelung der Daten sinnvoll
226         if len(df['datetime'].values) > 600:
227             moving_average = 300
228         else:
229             moving_average = 1
230
231         fig = px.line(df, x=df['datetime'], y=df[request_data["quantity"
]].rolling(moving_average).mean(), labels={'x': 'Zeit', 'y':
request_data["quantity"]})
232         fig['data'][0]['line']['color']='#476db0'
233         graphJSON = json.dumps(fig, cls=plotly.utils.PlotlyJSONEncoder)
234         return jsonify(graphJSON)
235     else:
236         graphJSON = json.dumps(px.line(), cls=plotly.utils.
PlotlyJSONEncoder)
237         return render_template('data.html', graphJSON=graphJSON, sensors=
sensor_arr, quantities=quantities_arr)
238
239
240
241 # ~~~~~
242 # API - WebApp
243 # ~~~~~
244 @app.route('/API', methods=['GET', 'POST'])
245 @auth_required()
246 def api():
247     db_token = db_account_management.token.find_one({'owner_id':
current_user.id})
248     if request.method=='POST' or db_token==None:
249         encoded_token = jwt.encode({
250             'user': str(current_user.id),
251             'exp' : datetime.datetime.utcnow() + datetime.timedelta(
days=30)
252         },
253         app.config['SECRET_KEY'],
254         algorithm="HS256")
255
```

```

256     if db_token != None:
257         db_account_management.token.update_one({'owner_id':
current_user.id}, {'$set': {'token': encoded_token}})
258     else:
259         token = {'owner_id': current_user.id, 'token': encoded_token}
260         db_account_management.token.insert_one(token)
261         db_token = db_account_management.token.find_one({'owner_id':
current_user.id})
262         data = jwt.decode(db_token['token'], app.config['SECRET_KEY'],
algorithms=["HS256"])
263         exp_date = datetime.datetime.fromtimestamp(data['exp'])
264         status = True
265         redirect(url_for('api'))
266
267     try:
268         data = jwt.decode(db_token['token'], app.config['SECRET_KEY'],
algorithms=["HS256"])
269         exp_date = datetime.datetime.fromtimestamp(data['exp'])
270         status = True
271     except:
272         exp_date = ''
273         status = False
274     return render_template('api.html', token=db_token['token'], status=
status, date=exp_date)
275
276
277
278 # ~~~~~
279 # GET - Regression nach Plackett-Burmann
280 # ~~~~~
281 @app.route('/API/get/regression_pb', methods=['GET', 'POST'])
282 @token_required
283 def api_get_pb():
284     token = request.args.get('token')
285     token_user = db_account_management.token.find_one({'token': str(token)
})['owner_id']
286
287     # Nehme Daten entgegen oder gebe Fehlernachricht zurück
288     api_get_json = request.get_json()
289     if not api_get_json:
290         return jsonify({"message": "No input data provided"}), 400
291
292     try:
293         return jsonify({'THI [-]': calc_thi_pb(api_get_json), 'R [-]':
calc_R_pb(api_get_json), 'CH4 [kg/(h*Kuh)]': calc_ch4_pb(api_get_json)
})
294     except Exception as err:
295         return jsonify({'message': f'error during regression function {err
}'})
296

```

```
297
298
299 # ~~~~~
300 # GET - Regression nach Box-Behnken
301 # ~~~~~
302 @app.route('/API/get/regression_bb', methods=['GET', 'POST'])
303 @token_required
304 def api_get_bb():
305     token = request.args.get('token')
306     token_user = db_account_management.token.find_one({'token': str(token)
307     })['owner_id']
308
309     # Nehme Daten entgegen oder gebe Fehlernachricht zurück
310     api_get_json = request.get_json()
311     if not api_get_json:
312         return jsonify({"message": "No input data provided"}), 400
313
314     try:
315         return jsonify({'THI [-]': calc_thi_bb(api_get_json), 'R [-]':
316         calc_R_bb(api_get_json), 'CH4 [kg/(h*Kuh)]': calc_ch4_bb(api_get_json)
317         })
318     except Exception as err:
319         return jsonify({'message': f'error during regression function {err
320         }'})
321
322 # ~~~~~
323 # GET - Sensordaten
324 # ~~~~~
325 @app.route('/API/get/sensordata', methods=['GET', 'POST'])
326 @token_required
327 def api_get_sensordata():
328     token = request.args.get('token')
329     token_user = db_account_management.token.find_one({'token': str(token)
330     })['owner_id']
331
332     # Nehme Daten entgegen oder gebe Fehlernachricht zurück
333     api_post_json = request.get_json()
334     if not api_post_json:
335         return jsonify({"message": "No input data provided"}), 400
336
337     # Falls Sensor in DB --> lade Daten
338     if db_digital_twin.sensors.find_one({'owner_id': token_user, '_id':
339     ObjectId(api_post_json['_id'])}):
340         # Lade Daten
341         data = db_digital_twin.sensors.find_one({'owner_id': token_user, '
342         _id': ObjectId(api_post_json['_id'])})
343         df = pd.read_json(data['data'], orient="split")
```

```

339     daterange = pd.read_json(api_post_json['daterange'], orient="split
")
340     daterange = [daterange['datetime'].iloc[0], daterange['datetime'].
iloc[1]]
341
342     # Filtere nach Datumsvorgabe
343     df = df[(df['datetime'] > daterange[0]) & (df['datetime'] <
daterange[1])]
344     return jsonify({"message": "OK", "status": 200, "data": df.to_json
(orient="split")})
345     else:
346         return jsonify({"message": "sensor not found"}), 404
347
348
349
350 # ~~~~~
351 # POST
352 # ~~~~~
353 @app.route('/API/post', methods=['GET', 'POST'])
354 @token_required
355 def api_post():
356     token = request.args.get('token')
357     token_user = db_account_management.token.find_one({'token': str(token)
})['owner_id']
358
359     # Nehme Daten entgegen oder gebe Fehlernachricht zurück
360     api_post_json = request.get_json()
361     if not api_post_json:
362         return jsonify({"message": "No input data provided"}), 400
363
364     # Falls Sensor nicht in DB --> lege Sensor an und füge Daten ein
365     if not db_digital_twin.sensors.find_one({'owner_id': token_user, '
sensor': api_post_json['sensor']}):
366         new_data = {'owner_id': token_user, 'sensor': api_post_json['
sensor'], 'data': api_post_json['data']}
367         inserted_id = db_digital_twin.sensors.insert_one(new_data).
inserted_id
368         return jsonify({'message': f'new data inserted with sensor id: {
inserted_id}'}), 201
369     else:
370         return jsonify({'message': f'sensor already inserted -> use patch'
})
371
372
373
374 # ~~~~~
375 # PATCH
376 # ~~~~~
377 @app.route('/API/patch', methods=['GET', 'POST'])
378 @token_required

```

```

379 def api_patch():
380     token = request.args.get('token')
381     token_user = db_account_management.token.find_one({'token': str(token)
382     })['owner_id']
383
384     # Nehme Daten entgegen oder gebe Fehlernachricht zurück
385     api_post_json = request.get_json()
386     if not api_post_json:
387         return jsonify({"message": "No input data provided"}), 400
388
389     # Falls Sensor in DB --> erneuere Daten
390     if db_digital_twin.sensors.find_one({'owner_id': token_user, '_id':
391     ObjectId(api_post_json['_id'])}):
392         # Lade alte Daten
393         old_data = db_digital_twin.sensors.find_one({'owner_id':
394         token_user, '_id': ObjectId(api_post_json['_id'])})
395         df_old = pd.read_json(old_data['data'], orient="split")
396         df_new = pd.read_json(api_post_json['data'], orient="split")
397
398         # Füge neue Daten an und speichere in der Datenbank
399         try:
400             df = pd.concat([df_old, df_new], axis=0)
401             updates = {'$set': {'data': df.to_json(orient="split")}}
402             db_digital_twin.sensors.update_one({'owner_id': token_user, '_
403             id': ObjectId(api_post_json['_id'])}, updates)
404             return jsonify({'message': 'dataset has been updated'}), 200
405         except Exception as err:
406             print(err)
407             return jsonify({'message': 'during patch process the
408             following error occured {err}'})
409         else:
410             return jsonify({'message': f'sensor not found -> use post'})
411
412 # ~~~~~
413 # DELETE
414 # ~~~~~
415 @app.route('/API/delete', methods=['GET', 'POST'])
416 @token_required
417 def api_delete():
418     token = request.args.get('token')
419     token_user = db_account_management.token.find_one({'token': str(token)
420     })['owner_id']
421
422     # Nehme Daten entgegen oder gebe Fehlernachricht zurück
423     api_post_json = request.get_json()
424     if not api_post_json:
425         return jsonify({"message": "No input data provided"}), 400

```

```

423     try:
424         db_digital_twin.sensors.delete_one({'owner_id': token_user, '_id':
         ObjectId(api_post_json['_id'])})
425         return jsonify({'message': f'dataset with _id {api_post_json["_id
         "]} has been deleted from database'})
426     except Exception as err:
427         return jsonify({'message': f'during delete the following error
         occured {err}'})
428
429
430
431 # ~~~~~
432 # Main
433 # ~~~~~
434 if __name__ == '__main__':
435     app.run(debug=True)

```

E.4 Beispiele zur Programmierschnittstelle

```

1 # ~~~~~
2 # Importieren der Bibliotheken
3 # ~~~~~
4 import jwt
5 import requests
6 import datetime
7 import pandas as pd
8
9 # ~~~~~
10 # Anlegen der Variablen
11 # und Konstanten
12 # ~~~~~
13 adress="http://192.168.0.124:5000/API/get/"
14 _id = "646a0ed11d1de25197f6d2b7"
15 api_key = "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.
         eyJ1c2Vybm91d285IjoiIiwiaWF0IjoiYjoxZjoxNjg3MDA5MTgyfQ.
         iBxq91QDbBqc3B0WxC_qFYmX00P-3uBge2e-xHAMxPQ"
16
17
18 # ~~~~~
19 # Versenden der Anfrage von Sensor-
20 # Daten und Ausgabe der Antwort
21 # ~~~~~
22 daterange = {'datetime': [pd.Timestamp(2023, 1, 31, 14, 0, 0), pd.
         Timestamp(2023, 1, 31, 14, 14, 0)]}
23 df = pd.DataFrame(daterange)
24
25 r = requests.post(adress+'sensordata?token='+api_key, json={'_id': _id, "
         daterange": df.to_json(orient="split")})
26 print(r.json()['message'])

```



```
16 # Verbinden zum OPC UA Server
17 # mittels asyncua Bibliothek
18 # ~~~~~
19 dt = 10 # Intervall der Abfrage
20 _id = '648b2f812584dddd1c56e5f0'
21
22 async with Client(url='opc.tcp://localhost:4840/freeopcua/server/') as
    client:
23     while True:
24         # OPC UA Abfrage der Daten vom Knoten '2'
25         node = client.get_node('ns=2;i=2')
26         value = await node.read_value()
27
28         # Aufruf der Patch-Funktion via REST-Befehl
29         df_new = pd.DataFrame({'datetime': [datetime.datetime.now()], '
Windrichtung in Grad': [value]})
30         r = requests.post(adress+'patch?token='+api_key, json={'_id': _id,
"data": df_new.to_json(orient="split")})
31         print(r.text)
32
33         # Abwarten des Messintervalls
34         await asyncio.sleep(dt)
```

Abbildungsverzeichnis

1.1	Ein typischer frei belüfteter Stall mit außen liegender Futterplatte in der Vollansicht (oben) und Schnittdarstellung (unten) sowie gekennzeichneten und detaillierten Lüftungselementen (eigene Darstellung)	2
1.2	Einordnung der Literatur zum digitalen Zwilling in der Landwirtschaft gemäß den von PURCELL und NEUBAUER (2023) publizierten Daten (eigene Darstellung)	3
1.3	Aufbau der Arbeit (eigene Darstellung)	6
2.1	Die Rolle des digitalen Zwillings über den Produktlebenszyklus in Anlehnung an VERDOUW et al. (2021)	8
2.2	Darstellung der unterschiedlichen Konzepte digitaler Zwillinge nach VERDOUW et al. (2021) (aus dem engl.)	9
2.3	Visualisierung des Gaußschen Divergenztheorems in der Anwendung auf ein Vektorfeld \mathbf{A} . Die Akkumulation im Volumen entspricht dem Fluss über die Oberflächen des Volumens (eigene Darstellung)	13
2.4	Darstellung eines diskretisierten Rechengebiets Ω , mit der Gebietsgrenze $\partial\Omega$ und der Darstellung einer Diskretisierung durch die FEM (links) und FVM (rechts), in Anlehnung an FONTES (2018) (eigene Darstellung)	15
2.5	Illustration der Anwendung von (linearen) Ansatzfunktionen in der FEM (links) und eines Diskretisierungsschemas erster Ordnung für die FVM (rechts) (eigene Darstellung)	16
2.6	Zerlegung einer Zielgröße in zeitliche Mittelung plus turbulente Fluktuation (eigene Darstellung)	16

2.7	Prozessdarstellung der Verifizierung und Validierung im Kontext computergestützter Simulationen. Eigene Darstellung, in Anlehnung an SCS (1979) aus dem engl.	19
2.8	Schematische Darstellung des Diskretisierungsfehlers (links) und der Richardson-Extrapolation (rechts) (eigene Darstellung)	21
2.9	Darstellung der grafischen Aufbereitung einer Sensitivitätsanalyse des Turbulenzmodells (eigene Darstellung)	23
2.10	Darstellung des Faktorraumes eines vollfaktoriellen (links), teilfaktoriellen (mitig) und Wirkungsflächen-Versuchsplans nach Box-Behnken (rechts)	24
2.11	Vereinfachte Darstellung eines biologischen Neurons links und die Darstellung der künstlichen Adaption rechts (eigene Darstellung)	26
2.12	Aufbau eines vollständig verknüpften künstlichen neuronalen Netzes (eigene Darstellung)	27
2.13	Gradientenbasierte Optimierung der Parameter Θ einer Funktion $f(\Theta)$. Eigene Darstellung, unter Verwendung von MATPLOTLIB (HUNTER, 2007)	28
2.14	Schema der <i>backpropagation</i> am Beispiel eines neuronalen Netzes mit einer verdeckten Schicht und je einem Neuron pro Schicht sowie Aktualisierung der Parameter mittels Gradientenabstieg (eigene Darstellung)	30
3.1	Karte der Umgebung des Versuchsstalls ($51^\circ 50' 3,57''N$ $6^\circ 34' 6,4092''O$) mit Kompass und Markierungen relevanter Objekte (eigene Darstellung)	42
3.2	Aufbau und Luftaufnahme des Versuchsstalls, mit freundlicher Genehmigung der Firma WELSING MILCHVIEH (eigene Darstellung)	42
3.3	Fotos des Versuchsstalls, mit freundlicher Genehmigung der Firma WELSING MILCHVIEH	43
3.4	Aufbau des 3D-USA und Darstellung der Messstrecken (eigene Darstellung)	44
3.5	Markierung 1 zeigt den Schaltplan eines TGS 2600 MOX-Sensors und der Auswerteeinheit, 2 stellt den Aufbau der Kalibrierungsmessungen und 3 das Diagramm der Messwerte einer Kalibrierungsmessung bei atmosphärischer Konzentration dar (eigene Darstellung)	46
3.6	Übersicht der Positionen aller Sensoren am und im Stall (eigene Darstellung)	49
3.7	Boxplot der Geschwindigkeitskomponenten u , v und w im lokalen Koordinatensystem des Stalls für alle Sensorpositionen und Messkampagnen (eigene Darstellung)	51
3.8	Windrosen der Messstation des DWD und am Versuchsstall über alle Messkampagnen (eigene Darstellung)	52
3.9	Boxplot der virtuellen Lufttemperatur gemäß den 3D-USA für alle Sensorpositionen und Messkampagnen (eigene Darstellung)	53
3.10	Auswertung der Methansensoren in einem linksseitig dargestellten Liniendiagramm und rechtsseitigen Boxplot (eigene Darstellung)	55

4.1	Ein frei belüfteter Kuhstall und das Nachbargebäude mit markierten geometrischen Parametern (eigene Darstellung)	58
4.2	Gruppierung beispielhafter Konstruktionselemente und deren Verknüpfung gemäß <i>Resilient Modeling</i> Strategie (eigene Darstellung)	60
4.3	Darstellung des Strömungsraumes zur CFD-Simulation mit Gebäuden gemäß TOMINAGA et al. (2008) und Anwendung der Richtlinie auf die Kernelemente des CAD-Modells bei mittlerer Einstellung der geometrischen Parameter (eigene Darstellung)	61
4.4	Teilweise Schnittdarstellung des Stalls mit den Abmessungen der parametrisierten Geometrien und Funktionsräume (eigene Darstellung)	62
4.5	Zeichnung des Stalls in Schnitt- und Volldarstellungen, inkl. der Detailbetrachtung der Skizzen für die Extrusion und Musterung der Stützen im CAD-Programm (eigene Darstellung)	63
4.6	CAD-Geometrien des Stalls, inkl. Detailbetrachtungen der Skizzen zur Extrusion und Musterung der Kühe (eigene Darstellung)	64
4.7	Schnitt- und Volldarstellung des Stalls, inkl. Detailbetrachtung der implementierten Lüftungselemente und deren Musterung (eigene Darstellung)	65
4.8	Grobe Diskretisierung des Strömungsraumes und der Stallgeometrie gemäß FRANKE und BAKLANOV (2007). Ansicht der $z-x$ und $z-y$ Schnittebenen bei $y > 22,0$ m und $x > 1,0$ m (eigene Darstellung)	66
4.9	Mittlere Diskretisierung des Strömungsraumes und der Stallgeometrie gemäß FRANKE und BAKLANOV (2007). Ansicht der $z-x$ und $z-y$ Schnittebenen bei $y > 22,0$ m und $x > 1,0$ m (eigene Darstellung)	67
4.10	Feine Diskretisierung des Strömungsraumes und der Stallgeometrie gemäß FRANKE und BAKLANOV (2007). Ansicht der $z-x$ und $z-y$ Schnittebenen bei $y > 22,0$ m und $x > 1,0$ m (eigene Darstellung)	67
4.11	Darstellung der im numerischen Simulationsmodell angelegten Randbedingungen bei einer West-Ost-Anströmung des Stalls (eigene Darstellung)	71
4.12	Konvergenzverlauf des iterativen Fehlers (links) und Beobachtung der maximalen Werte verschiedener Quantitäten im Simulationsmodell (rechts) (eigene Darstellung)	73
4.13	Ergebnisse der Auswertung der Strömungsgeschwindigkeiten in Hauptströmungsrichtung für die Parameterkonstellation des Verifizierungsmodells und Kennzeichnung der Position für die Auswertung im Stall (eigene Darstellung)	74
4.14	Geometrische Varianten des CAD-Modells für die Validierung mittels RANS-Simulation (eigene Darstellung)	76
4.15	Sensitivitätsanalyse der geometrischen Varianten des CAD-Modells (eigene Darstellung)	77
4.16	Sensitivitätsanalyse der Windgeschwindigkeit (eigene Darstellung)	78
4.17	Sensitivitätsanalyse der Windrichtung α (eigene Darstellung)	78
4.18	Sensitivitätsanalyse des Turbulenzmodells (eigene Darstellung)	79

4.19	Ausschnitt der Ergebnisse des Geschwindigkeitsbetrages, der Lufttemperatur, Luftfeuchte und des Methangehaltes auf der Schnittebene $x-y$ bei $z = 1,7$ m (eigene Darstellung)	80
4.20	Ausschnitt der Ergebnisse des Geschwindigkeitsbetrages, der Lufttemperatur, Luftfeuchte und des Methangehaltes auf der Schnittebene $z-x$ bei $y = 29,6$ m (eigene Darstellung)	81
4.21	Darstellung des y^+ Wertes für die Validierungssimulation in der $x-y$ -Ebene bei einer Schnittdarstellung auf einer Höhe von $z = 3,5$ m (eigene Darstellung)	82
4.22	Wahrscheinlichkeitsnetz des Plackett-Burman-Versuchsplans hinsichtlich des THI ausgewertet in MINITAB (Version 19) (eigene Darstellung)	85
4.23	Wahrscheinlichkeitsnetz des <i>Screening</i> -Versuchsplans hinsichtlich des emittierten CH_4 ausgewertet in MINITAB (Version 19) (eigene Darstellung) . . .	86
4.24	Wahrscheinlichkeitsnetz der Luftwechselrate R_V ausgewertet in MINITAB (Version 19) (eigene Darstellung)	87
4.25	Wahrscheinlichkeitsnetz der Ergebnisse des Box-Behnken-Versuchsplans gemäß Auswertung des Temperatur-Luftfeuchte-Indexes in MINITAB (Version 19) (eigene Darstellung)	89
4.26	Antwortflächen der Faktoren des Box-Behnken-Versuchsplans gemäß Auswertung nach Tierwohl-Index THI (eigene Darstellung)	90
4.27	Wahrscheinlichkeitsnetz des Box-Behnken-Versuchsplans hinsichtlich des emittierten CH_4 ausgewertet in MINITAB (Version 19) (eigene Darstellung) . . .	91
4.28	Antwortflächen der Faktoren des Box-Behnken-Versuchsplans gemäß Auswertung nach Emissionsfluss \dot{m}_{CH_4} (eigene Darstellung)	91
4.29	Wahrscheinlichkeitsnetz der Ergebnisse des Box-Behnken-Versuchsplans gemäß Auswertung der Luftwechselrate in MINITAB (Version 19) (eigene Darstellung)	92
4.30	Antwortflächen der Faktoren des Box-Behnken-Versuchsplans gemäß Auswertung nach Luftwechselrate R_V (eigene Darstellung)	92
5.1	Darstellung des verallgemeinerten Trainingsalgorithmus für PINNs (eigene Darstellung)	98
5.2	Strömungsmerkmale an einer rückwärtsgewandten Stufe, in Anlehnung an PIOCH, HARMENING, MÜLLER et al. (2023)	99
5.3	Randbedingungen für das Training eines PINN an einer rückwärtsgewandten Stufe, in Anlehnung an PIOCH, HARMENING, MÜLLER et al. (2023)	99
5.4	Vorhersage des PINN im Strömungsfeld für die rückwärtsgewandte Stufe mit einem Mischlängenmodell (links) und der Vergleich mit DNS-Daten (rechts) (eigene Darstellung)	100
5.5	Vorhersage des PINN im Strömungsfeld für die rückwärtsgewandte Stufe mit einem $k-\omega$ -Modell (links) und der Vergleich mit DNS-Daten (rechts) (eigene Darstellung)	101

5.6	Vorhersage des PINN zur rückwärts-gewandten Stufe mit dem ν_t -Modell (links) und der Vergleich mit DNS-Daten (rechts) (eigene Darstellung)	102
5.7	Vorhersage des PINN im Strömungsfeld für die rückwärts-gewandte Stufe mit einem pseudo-Reynolds-Spannungs-Turbulenzmodell (links) und der Vergleich mit DNS-Daten (rechts) (eigene Darstellung)	103
5.8	Grafische Darstellung eines Hyperwürfels bei der Umströmung eines Gebäudes mit variablem Dachneigungswinkel β (links), Koordinatendatei und Ergebnisdatei auf der Schnittebene bei $\beta = 20^\circ$ (mittig) und das zugehörige neuronale Netz, bei Verwendung eines Turbulenzmodells mit pseudo-Reynolds-Spannungen (rechts) (eigene Darstellung)	104
5.9	Boxplot der Betragswerte des relativen Fehlers von PINNs bei variiertem Reynolds-Zahl über die Viskosität (links) und erzwungener Konvektion (rechts), in Anlehnung an HARMENING, PIOCH, FUHRIG et al. (2023)	106
5.10	Boxplot der Auswertung des relativen Fehlers zwischen der CFD-Referenz und dem PINN für die Umströmung eines Tragflügelprofils bei variiertem Größe der Datensätze für das Training, in Anlehnung an HARMENING, PIOCH, FUHRIG et al. (2023)	107
5.11	Boxplot der Auswertung des relativen Fehlers zwischen der CFD-Referenz und dem PINN für ein umströmtes Tragflügelprofil bei variiertem <i>Sampling</i> -Methode, in Anlehnung an HARMENING, PIOCH, FUHRIG et al. (2023)	108
5.12	Echtzeitanforderung für die Vorhersage der vom Wind aufgeprägten Strömungen durch einen frei belüfteten Stall (eigene Darstellung)	108
5.13	Hyperwürfel und Trainingskoordinaten des PINN auf einer Schnittebene bei $\beta = 20^\circ$ (eigene Darstellung)	109
5.14	<i>Sampling</i> der Datenpunkte für das Training des PINN am Beispiel des Box-Behnken-Versuchsdurchlaufes N8 (eigene Darstellung)	110
5.15	Ablaufdiagramm für das PINN, vom Datenexport über die Angabe der Gleichungen, bis hin zur Visualisierung und Speicherung der Daten des Trainingsprozesses (eigene Darstellung)	112
5.16	Gemittelte Verlustmetrik und Standardabweichung des Trainingsprozesses für ein PINN zur Vorhersage der Strömungen am Stall (eigene Darstellung)	114
5.17	Konturdiagramme der CFD-Simulation, (reskalierten) PINN-Vorhersage und dem sich ergebenden absoluten Fehler für $W = 60\%$, $\beta = 25^\circ$ und $v_m(10) = 3,5 \frac{\text{m}}{\text{s}}$ (eigene Darstellung)	114
5.18	Konturdiagramme der CFD-Simulation, (reskalierten) PINN-Vorhersage und dem sich ergebenden absoluten Fehler für $W = 75\%$, $\beta = 22,5^\circ$ und $v_m(10) = 2,75 \frac{\text{m}}{\text{s}}$ (eigene Darstellung)	115
5.19	Konturdiagramme der CFD-Simulation, (reskalierten) PINN-Vorhersage und dem sich ergebenden absoluten Fehler für $W = 15\%$, $\beta = 27,5^\circ$ und $v_m(10) = 4,25 \frac{\text{m}}{\text{s}}$ (eigene Darstellung)	116

6.1	Kommunikationshierarchie in Anlehnung an DIN EN 62264-1:2014-07 (2014) (eigene Darstellung)	120
6.2	OPC UA Kommunikationsstruktur, in Anlehnung an CAVALIERI und CHIAC- CHIO (2013) (eigene Darstellung)	122
6.3	REST Kommunikationsstruktur (eigene Darstellung)	123
6.4	Visualisierung des entwickelten Kommunikationsschemas für digitale Zwillinge (eigene Darstellung)	127
6.5	Dateistruktur der Webapplikation (eigene Darstellung)	128
6.6	<i>Login</i> -Seite der entwickelten Webapplikation (eigene Darstellung)	128
6.7	Darstellung der Seite der Webapplikation für die PINN-Simulation beim Zu- griff über einen Computer (eigene Darstellung)	129
6.8	Datenbankseite der entwickelten Webapplikation (eigene Darstellung)	129
6.9	API-Seite der entwickelten Webapplikation (eigene Darstellung)	130
6.10	Zugriff auf die Webapplikation über GOOGLE CHROME von einem Mobiltele- fon (links) und einem <i>Tablet</i> (rechts) (eigene Darstellung)	130
8.1	Visuelle Zusammenfassung der Arbeit (eigene Darstellung)	144

Tabellenverzeichnis

2.1	Auflösung von Versuchsplänen nach SIEBERTZ, BEBBER und HOCHKIRCHEN (2017)	25
2.2	Einordnung der Literatur zu numerischen Modellen von Stallungen	33
2.3	Einordnung der Literatur Physik-informierter neuronaler Netze mit Anwendung auf die Navier-Stokes-Gleichungen	35
2.4	Kategorisierte Literatur zu digitalen Zwillingen in der Landwirtschaft	38
3.1	Spezifikationen des 3D-USA (Modell 4.3830.22.300) der Firma ADOLF THIES	44
3.2	Spezifikationen der LUBRATEC SMARTBOX von HUESKER SYNTHETIC	47
3.3	Spezifikationen der Geräte der Wetterstation des DWD mit nationaler Kennung 00554 in Bocholt-Liedern (Wasserwerk)	47
3.4	Spezifikationen der Geräte der lokalen Wetterstation am Versuchsstall	48
3.5	Positionsangaben der Sensoren im lokalen Koordinatensystem des Versuchsstalls	49
3.6	Zeitfolge der Lüftungseinstellungen für die Axiallüfter der ersten Messkampagne	50
3.7	Mittlere Temperaturen der Wetterstationen des DWD und am Versuchsstall für die durchgeführten Messkampagnen	53
3.8	Mittlere Luftfeuchte der Wetterstation und LUBRATEC SMARTBOXEN für die durchgeführten Messkampagnen	54
3.9	Gemittelte Messwerte der Methansensoren für alle Messkampagnen	55
4.1	Geometrische Modellparameter für den Versuchsplan	59
4.2	Netzunabhängigkeitsstudie und Unsicherheitsanalyse der mittleren Temperatur, Methankonzentration und Luftfeuchte. Auswertung entlang der x -Achse von $x_{min} = -20$ m bis $x_{max} = +20$ m bei $y_1 = 16$ m und $y_2 = 24$ m	75
4.3	Validierungsmetriken bei mittlerer Einstellung der Parameter	80

4.4	Placket-Burman-Versuchsplan für 14 Faktoren und die ausgewerteten Ergebnisse des THI, Emissionsstroms und der Luftwechselrate, erstellt mittels MINITAB (Version 19)	84
4.5	Statistischer Box-Behnken-Versuchsplan und die ausgewerteten Ergebnisse für die Faktoren Wickellüftung W , den Dachneigungswinkel β und der Basisgeschwindigkeit $v_m(10)$, erstellt mittels MINITAB (Version 19)	88
5.1	Hyperparameter des PINN	111
5.2	Rechenzeit und Standardabweichung des PINN für unterschiedlich große Datensätze und zwei Hardwarekonfigurationen	113
5.3	Validierungsmetriken für den Abgleich des PINN mit den Daten der CFD-Simulation für den Versuchsdurchlauf zwölf des Box-Behnken-Versuchsplans	115
5.4	Validierungsmetriken für den Abgleich des PINN mit den Daten der CFD-Simulation bei einer Interpolation im Hyperwürfel	116
5.5	Validierungsmetriken für den Abgleich des PINN mit den Daten der CFD-Simulation für eine Extrapolation aus dem Hyperwürfel	117

Literatur

- ABBASI, Rabiya, Pablo MARTINEZ und Rafiq AHMAD (2022). „The digitization of agricultural industry – a systematic literature review on agriculture 4.0“. In: *Smart Agricultural Technology* 2, S. 100042. DOI: 10.1016/j.atech.2022.100042.
- ALLEN, B. Danette (2021). *Digital twins and living models at NASA*. <https://ntrs.nasa.gov/citations/20210023699>. (Abgerufen am 01.02.2023).
- ALVES, Rafael Gomes et al. (2019). „A digital twin for smart farming“. In: *Proceedings of the 2019 IEEE Global Humanitarian Technology Conference (GHTC)*. IEEE. DOI: 10.1109/ghtc46095.2019.9033075.
- ANGIN, Pelin et al. (2020). „AgriLoRa: A Digital Twin Framework for Smart Agriculture“. In: *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.* 11, S. 77–96. DOI: 10.22667/JOWUA.2020.12.31.077.
- ARTHURS, Christopher J. und Andrew P. KING (2021). „Active training of physics-informed neural networks to aggregate and interpolate parametric solutions to the Navier-Stokes equations“. In: *Journal of Computational Physics* 438, S. 110364. DOI: 10.1016/j.jcp.2021.110364.
- ARZANI, Amirhossein, Jian-Xun WANG und Roshan M. D’SOUZA (2021). *Uncovering near-wall blood flow from sparse data with physics-informed neural networks*. Preprint. DOI: 10.48550/ARXIV.2104.08249.
- ASME (2009). *ASME V&V 20-2009: Standard for verification and validation in computational fluid dynamics and heat transfer*. Reaffirmed 2016. Bd. 20-2009. ASME V&V. ASME Press. ISBN: 0791832090.
- BJERG, Bjarne und Mathias ANDERSEN (2010). „Numerical simulation of a pit exhausts system for reduction of ammonia emission from a naturally ventilated cattle building“. In: *Proceedings of the XVIIth World Congress of the International Commission of Agricultural and Biosystems Engineering (CIGR)*.

- BJERG, Bjarne, Giovanni CASCONE et al. (2013). „Modelling of ammonia emissions from naturally ventilated livestock buildings. Part 3: CFD modelling“. In: *Biosystems Engineering* 116.3, S. 259–275. DOI: 10.1016/j.biosystemseng.2013.06.012.
- BJERG, Bjarne, Guoqiang ZHANG et al. (2011). „Methane emission from naturally ventilated livestock buildings can be determined from gas concentration measurements“. In: *Environmental Monitoring and Assessment* 184.10, S. 5989–6000. DOI: 10.1007/s10661-011-2397-8.
- BLANES-VIDAL, V. et al. (2008). „Application of computational fluid dynamics to the prediction of airflow in a mechanically ventilated commercial poultry building“. In: *Biosystems Engineering* 100.1, S. 105–116. DOI: 10.1016/j.biosystemseng.2008.02.004.
- BLOCKEN, Bert et al. (2011). „Application of computational fluid dynamics in building performance simulation for the outdoor environment: an overview“. In: *Journal of Building Performance Simulation* 4.2, S. 157–184. DOI: 10.1080/19401493.2010.513740.
- BOX, George E. P. (1976). „Science and Statistics“. In: *Journal of the American Statistical Association* 71.356, S. 791–799. DOI: 10.1080/01621459.1976.10480949.
- BRAY, Tim (2014). *The JavaScript Object Notation (JSON) Data Interchange Format*. RFC 7159. DOI: 10.17487/RFC7159.
- BUSTAMANTE, Eliseo et al. (2013). „Exploring Ventilation Efficiency in Poultry Buildings: The Validation of Computational Fluid Dynamics (CFD) in a Cross-Mechanically Ventilated Broiler Farm“. In: *Energies* 6.5, S. 2605–2623. DOI: 10.3390/en6052605.
- CAI, Shengze et al. (2021). „Physics-informed neural networks (PINNs) for fluid mechanics: a review“. In: *Acta Mechanica Sinica* 37.12, S. 1727–1738. DOI: 10.1007/s10409-021-01148-1.
- CAMBA, Jorge D., Manuel CONTERO und Pedro COMPANY (2016). „Parametric CAD modeling: An analysis of strategies for design reusability“. In: *Computer-Aided Design* 74, S. 18–31. DOI: 10.1016/j.cad.2016.01.003.
- CAVALIERI, Salvatore und Ferdinando CHIACCHIO (2013). „Analysis of OPC UA performances“. In: *Computer Standards & Interfaces* 36.1, S. 165–177. DOI: 10.1016/j.csi.2013.06.004.
- CELIK, Ismail B. et al. (2008). „Procedure for Estimation and Reporting of Uncertainty Due to Discretization in CFD Applications“. In: *Journal of Fluids Engineering* 130.7, S. 078001. DOI: 10.1115/1.2960953.
- CHANG, J. C. und S. R. HANNA (2004). „Air quality model performance evaluation“. In: *Meteorology and Atmospheric Physics* 87.1-3. DOI: 10.1007/s00703-003-0070-7.
- CHAUDHARI, M., I. KULKARNI und M. DAMODARAN (2021). „Exploring Physics-Informed Neural Networks for Compressible Flow Prediction“. In: *Lecture Notes in Mechanical Engineering*. Springer Singapore, S. 313–322. DOI: 10.1007/978-981-15-5183-3_34.

-
- CHAUX, Jesus David, David SANCHEZ-LONDONO und Giacomo BARBIERI (2021). „A Digital Twin Architecture to Optimize Productivity within Controlled Environment Agriculture“. In: *Applied Sciences* 11.19, S. 8875. DOI: 10.3390/app11198875.
- COLEMAN, Hugh W. und Fred STERN (1997). „Uncertainties and CFD Code Validation“. In: *Journal of Fluids Engineering* 119.4, S. 795–803. DOI: 10.1115/1.2819500.
- DAMASCENO, Flávio A. et al. (2020). „3D CFD Analysis of Natural Ventilation in Reduced Scale Model of Compost Bedded Pack Barn for Dairy Cows“. In: *Applied Sciences* 10.22, S. 8112. DOI: 10.3390/app10228112.
- DIN1991-1-4/NA:2010-12 (2010). *Nationaler Anhang - National festgelegte Parameter - Eurocode 1: Einwirkungen auf Tragwerke - Teil 1-4: Allgemeine Einwirkungen - Windlasten*. Norm. DOI: 10.31030/1723628.
- DIN EN 62264-1:2014-07 (2014). *DIN EN 62264-1:2014-07, Integration von Unternehmensführungs- und Leitsystemen- Teil 1: Modelle und Terminologie (IEC_62264-1:2013); Deutsche Fassung EN_62264-1:2013*. Norm. DOI: 10.31030/2156368.
- DOUMBIA, E. Moustapha, David JANKE, Qianying YI, Thomas AMON et al. (2021). „CFD modelling of an animal occupied zone using an anisotropic porous medium model with velocity depended resistance parameters“. In: *Computers and Electronics in Agriculture* 181, S. 105950. DOI: 10.1016/j.compag.2020.105950.
- DOUMBIA, E. Moustapha, David JANKE, Qianying YI, Alexander PRINZ et al. (2021). „A Parametric Model for Local Air Exchange Rate of Naturally Ventilated Barns“. In: *Agronomy* 11.8, S. 1585. DOI: 10.3390/agronomy11081585.
- DREWRY, Jessica L., Christopher Y. CHOI et al. (2018). „Computational model of methane and ammonia emissions from dairy barns: Development and validation“. In: *Computers and Electronics in Agriculture* 149, S. 80–89. DOI: 10.1016/j.compag.2017.07.012.
- DREWRY, Jessica L., Mario R. MONDACA et al. (2018). „A Computational Fluid Dynamics Model of Biological Heat and Gas Generation in a Dairy Holding Area“. In: *Transactions of the ASABE* 61.2, S. 449–460. DOI: 10.13031/trans.12394.
- EÇA, L. und M. HOEKSTRA (2002). „An Evaluation of Verification Procedures for CFD Algorithms“. In: *Proceedings of the Twenty-Fourth Symposium on Naval Hydrodynamics*, S. 568–587. DOI: 10.17226/10834.
- EGGLESTON, H. S. et al. (2006). *2006 IPCC Guidelines for National Greenhouse Gas Inventories*.
- EIVAZI, Hamidreza, Mojtaba TAHANI et al. (2022). „Physics-informed neural networks for solving Reynolds-averaged Navier–Stokes equations“. In: *Physics of Fluids* 34.7, S. 075117. DOI: 10.1063/5.0095270.

- EIVAZI, Hamidreza und Ricardo VINUESA (2022). *Physics-informed deep-learning applications to experimental fluid mechanics*. Preprint. DOI: 10.48550/ARXIV.2203.15402.
- ERDÉLYI, Viktor und László JÁNOSI (2019). „Digital Twin and Shadow in Smart Pork Fatteners“. In: *International Journal of Engineering and Management Sciences* 4.1, S. 515–520. DOI: 10.21791/ijems.2019.1.63..
- EUGSTER, Werner und George W. KLING (2012). „Performance of a low-cost methane sensor for ambient concentration measurements in preliminary studies“. In: *Atmospheric Measurement Techniques* 5.8, S. 1925–1934. DOI: 10.5194/amt-5-1925-2012.
- EUGSTER, Werner, James LAUNDRE et al. (2020). „Long-term reliability of the Figaro TGS 2600 solid-state methane sensor under low-Arctic conditions at Toolik Lake, Alaska“. In: *Atmospheric Measurement Techniques* 13.5, S. 2681–2695. DOI: 10.5194/amt-13-2681-2020.
- FAGUNDES, B. et al. (2020). „Comparison of airflow homogeneity in Compost Dairy Barns with different ventilation systems using the CFD model“. In: *Agronomy Research*. DOI: 10.15159/AR.20.106.
- FERZIGER, Joel H., Milovan PERIĆ und Robert L. STREET (2020). *Numerische Strömungsmechanik*. 2. Aufl. Springer Berlin Heidelberg. DOI: 10.1007/978-3-662-46544-8.
- FIEDLER, Merike et al. (2013). „Air velocity measurements using ultrasonic anemometers in the animal zone of a naturally ventilated dairy barn“. In: *Biosystems Engineering* 116.3, S. 276–285. DOI: 10.1016/j.biosystemseng.2012.10.006.
- FONTES, Ed (2018). *COMSOL Blog – FEM vs. FVM*. <https://www.comsol.de/blogs/fem-vs-fvm/>. (Abgerufen am 20.09.2022).
- FRANKE, Jörg (2006). „Recommendations of the COST action C14 on the use of CFD in predicting pedestrian wind environment“. In: *Proceedings of the fourth international symposium on computational wind engineering*, S. 529–532.
- FRANKE, Jörg und Alexander BAKLANOV (2007). *Best Practice Guideline for the CFD Simulation of Flows in the Urban Environment: COST Action 732 Quality Assurance and Improvement of Microscale Meteorological Models*. ISBN: 3-00-018312-4.
- GEBHARD, Richard (2013). „A resilient modeling strategy“. In: *Technical Presentation, Solid Edge University*.
- GEBREMEDHIN, Kifle G. (1987). „Effect of animal orientation with respect to wind direction on convective heat loss“. In: *Agricultural and Forest Meteorology* 40.2, S. 199–206. DOI: 10.1016/0168-1923(87)90007-4.
- GEBREMEDHIN, Kifle G. und Binxin WU (2002). „Simulation of sensible and latent heat losses from wet-skin surface and fur layer“. In: *Journal of Thermal Biology* 27.4, S. 291–297. DOI: 10.1016/s0306-4565(01)00091-2.

-
- GLAESSGEN, Edward und David STARGEL (2012). „The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles“. In: *Proceedings of the 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*. American Institute of Aeronautics und Astronautics. DOI: 10.2514/6.2012-1818.
- GRIEVES, Michael W. (2005). „Product lifecycle management: the new paradigm for enterprises“. In: *International Journal of Product Development* 2.1/2, S. 71. DOI: 10.1504/ijpd.2005.006669.
- GROHER, T., K. HEITKÄMPER und C. UMSTÄTTER (2020). „Digital technology adoption in livestock production with a special focus on ruminant farming“. In: *Animal* 14.11, S. 2404–2413. DOI: 10.1017/s1751731120001391.
- HAN, Xue et al. (2022). „AI Based Digital Twin Model for Cattle Caring“. In: *Sensors* 22.19, S. 7118. DOI: 10.3390/s22197118.
- HARMENING, Jan Hauke, Fabian PIOCH, Lennart FUHRIG et al. (2023). *Data-Assisted Training of a Physics-Informed Neural Network to Predict the Reynolds-Averaged Turbulent Flow Field around a Stalled Airfoil under Variable Angles of Attack*. Preprint. DOI: 10.20944/preprints202304.1244.v1.
- HARMENING, Jan Hauke, Fabian PIOCH und Dieter SCHRAMM (2022). „Physics Informed Neural Networks as Multidimensional Surrogate Models of CFD Simulations“. In: *Proceedings of the NAFEMS conference on machine learning und artificial intelligence in CFD and structural analysis*.
- HASANUZZAMAN, Gazi et al. (2023). „Enhancement of PIV measurements via physics-informed neural networks“. In: *Measurement Science and Technology* 34.4, S. 044002. DOI: 10.1088/1361-6501/aca9eb.
- HEIDEL, Roland et al. (2017). *Basiswissen RAMI4.0*. Hrsg. von DIN e. V. 1. Aufl. VDE Verlag. ISBN: 978-3-8007-4247-9.
- HEIDENREICH, Thomas et al. (2008). „Quantifizierung von Emissionen in der Rinderhaltung“. In: *Sächsisches Landesamt für Umwelt, Landwirtschaft und Geologie, Schriftenreihe* 33.
- HENNIGH, Oliver et al. (2021). „NVIDIA SimNet™: An AI-Accelerated Multi-Physics Simulation Framework“. In: *Computational Science – ICCS 2021*. Springer International Publishing, S. 447–461. DOI: 10.1007/978-3-030-77977-1_36.
- HUNTER, John D. (2007). „Matplotlib: A 2D Graphics Environment“. In: *Computing in Science & Engineering* 9.3, S. 90–95. DOI: 10.1109/mcse.2007.55.
- JANKE, David, Alfonso CAIAZZO et al. (2020). „On the feasibility of using open source solvers for the simulation of a turbulent air flow in a dairy barn“. In: *Computers and Electronics in Agriculture* 175, S. 105546. DOI: 10.1016/j.compag.2020.105546.

- JANKE, David, Dylia WILLINK et al. (2020). „Calculation of ventilation rates and ammonia emissions: Comparison of sampling strategies for a naturally ventilated dairy barn“. In: *Biosystems Engineering* 198, S. 15–30. DOI: 10.1016/j.biosystemseng.2020.07.011.
- JANS-SINGH, Melanie et al. (2020). „Digital twin of an urban-integrated hydroponic farm“. In: *Data-Centric Engineering* 1. DOI: 10.1017/dce.2020.21.
- JIN, Xiaowei et al. (2021). „NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations“. In: *Journal of Computational Physics* 426, S. 109951. DOI: 10.1016/j.jcp.2020.109951.
- JO, Seng-Kyoun, Dae-Heon PARK, Hyeon PARK und Se-Han KIM (2018). „Smart Livestock Farms Using Digital Twin: Feasibility Study“. In: *Proceedings of the 2018 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE. DOI: 10.1109/ictc.2018.8539516.
- JO, Seng-Kyoun, Dae-Heon PARK, Hyeon PARK, Younghoon KWAK et al. (2019). „Energy Planning of Pigsty Using Digital Twin“. In: *Proceedings of the 2019 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE. DOI: 10.1109/ictc46691.2019.8940032.
- JONES, David et al. (2020). „Characterising the Digital Twin: A systematic literature review“. In: *CIRP Journal of Manufacturing Science and Technology* 29, S. 36–52. DOI: 10.1016/j.cirpj.2020.02.002.
- KAYS, William M. (1994). „Turbulent Prandtl Number—Where Are We?“ In: *Journal of Heat Transfer* 116.2, S. 284–295. DOI: 10.1115/1.2911398.
- KINGMA, Diederik P. und Jimmy BA (2015). „Adam: A method for stochastic optimization“. In: *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. DOI: 10.48550/ARXIV.1412.6980.
- KNIBBE, Willem Jan et al. (2022). „Digital twins in the green life sciences“. In: *NJAS: Impact in Agricultural and Life Sciences* 94.1, S. 249–279. DOI: 10.1080/27685241.2022.2150571.
- KRISHNAPRIYAN, Aditi et al. (2021). „Characterizing possible failure modes in physics-informed neural networks“. In: *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS 2021)*. Bd. 34. Curran Associates, S. 26548–26560.
- KRITZINGER, Werner et al. (2018). „Digital Twin in manufacturing: A categorical literature review and classification“. In: *IFAC-PapersOnLine* 51.11, S. 1016–1022. DOI: 10.1016/j.ifacol.2018.08.474.
- LAGARIS, Isaac E., Aristidis LIKAS und Dimitrios I. FOTIADIS (1998). „Artificial neural networks for solving ordinary and partial differential equations“. In: *IEEE Transactions on Neural Networks* 9.5, S. 987–1000. DOI: 10.1109/72.712178.

-
- LANTSOVA, Evgenia, Valerii VTORYI und Sergei VTORYI (2015). „Investigation of water evaporation from cattle manure“. In: *Proceedings of the 14th International Scientific Conference Engineering for Rural Development*.
- LAUBSCHER, Ryno und Pieter ROUSSEAU (2021). *Application of mixed-variable physics-informed neural networks to solve normalised momentum and energy transport equations for 2D internal convective flow*. Preprint. DOI: 10.48550/ARXIV.2105.10889.
- LAUNDER, B. E. und D. B. SPALDING (1974). „The numerical computation of turbulent flows“. In: *Computer Methods in Applied Mechanics and Engineering* 3.2, S. 269–289. DOI: 10.1016/0045-7825(74)90029-2.
- LE, Hung, Parviz MOIN und John KIM (1997). „Direct numerical simulation of turbulent flow over a backward-facing step“. In: *Journal of Fluid Mechanics* 330, S. 349–374. DOI: 10.1017/s0022112096003941.
- LECHELER, Stefan (2018). *Numerische Strömungsberechnung*. Springer Fachmedien Wiesbaden. ISBN: 978-3-658-19191-7. DOI: 10.1007/978-3-658-19192-4.
- LEE, In-Bok et al. (2013). „The past, present and future of CFD for agro-environmental applications“. In: *Computers and Electronics in Agriculture* 93, S. 168–183. DOI: 10.1016/j.compag.2012.09.006.
- LENGERS, Bernd, Inga SCHIEFLER und Wolfgang BÜSCHER (2013). „A comparison of emission calculations using different modeled indicators with 1-year online measurements“. In: *Environmental Monitoring and Assessment* 185.12, S. 9751–9762. DOI: 10.1007/s10661-013-3288-y.
- LI, Shirong und Xinlong FENG (2022). „Dynamic Weight Strategy of Physics-Informed Neural Networks for the 2D Navier–Stokes Equations“. In: *Entropy* 24.9, S. 1254. DOI: 10.3390/e24091254.
- LIDE, David R., Hrsg. (2008). *CRC handbook of chemistry and physics: A Ready-Reference of Chemical and Physical Data*. 89. Aufl. CRC Press.
- LIU, Dong C. und Jorge NOCEDAL (1989). „On the limited memory BFGS method for large scale optimization“. In: *Mathematical Programming* 45.1-3, S. 503–528. DOI: 10.1007/bf01589116.
- LU, Lu et al. (2021). „DeepXDE: A Deep Learning Library for Solving Differential Equations“. In: *SIAM Review* 63.1, S. 208–228. DOI: 10.1137/19m1274067.
- MA, Hao et al. (2022). „Physics-Driven Learning of the Steady Navier-Stokes Equations using Deep Convolutional Neural Networks“. In: *Communications in Computational Physics* 32.3, S. 715–736. DOI: 10.4208/cicp.oa-2021-0146.
- MCCULLOCH, Warren S. und Walter PITTS (1943). „A logical calculus of the ideas immanent in nervous activity“. In: *The Bulletin of Mathematical Biophysics* 5.4, S. 115–133. DOI: 10.1007/bf02478259.

- MENEZES, Alfred J., Scott A. VANSTONE und Paul C. Van OORSCHOT (1996). *Handbook of Applied Cryptography*. 1st. CRC Press. ISBN: 978-0-8493-8523-0.
- MONDACA, Mario R. und Christopher Y. CHOI (2016). „An Evaluation of Simplifying Assumptions in Dairy Cow Computational Fluid Dynamics Models“. In: *Transactions of the ASABE* 59.6, S. 1575–1584. DOI: 10.13031/trans.59.11908.
- MONDACA, Mario R., Christopher Y. CHOI und Nigel B. COOK (2019). „Understanding microenvironments within tunnel-ventilated dairy cow freestall facilities: Examination using computational fluid dynamics and experimental validation“. In: *Biosystems Engineering* 183, S. 70–84. DOI: 10.1016/j.biosystemseng.2019.04.014.
- MONDACA, Mario R., Fernando ROJANO et al. (2013). „A Conjugate Heat and Mass Transfer Model to Evaluate the Efficiency of Conductive Cooling for Dairy Cattle“. In: *Transactions of the ASABE*, S. 1471–1482. DOI: 10.13031/trans.56.10178.
- MONTEIRO, José et al. (2018). „Towards Sustainable Digital Twins for Vertical Farming“. In: *Proceedings of the 2018 Thirteenth International Conference on Digital Information Management (ICDIM)*. IEEE. DOI: 10.1109/icdim.2018.8847169.
- MONTEIRO, José et al. (2022). „A scalable digital twin for vertical farming“. In: *Journal of Ambient Intelligence and Humanized Computing*. DOI: 10.1007/s12652-022-04106-2.
- MOSHREFZADEH, Mandana, Thomas MACHL und David GACKSTETTER (2020). *Towards a Distributed Digital Twin of the Agricultural Landscape*. Wichmann Verlag. DOI: 10.14627/537690019.
- NASIRAHMADI, Abozar und Oliver HENSEL (2022). „Toward the Next Generation of Digitalization in Agriculture Based on Digital Twin Paradigm“. In: *Sensors* 22.2, S. 498. DOI: 10.3390/s22020498.
- NEETHIRAJAN, Suresh und Bas KEMP (2021). „Digital Twins in Livestock Farming“. In: *Animals* 11.4, S. 1008. DOI: 10.3390/ani11041008.
- NEGRI, Elisa, Luca FUMAGALLI und Marco MACCHI (2017). „A Review of the Roles of Digital Twin in CPS-based Production Systems“. In: *Procedia Manufacturing* 11, S. 939–948. DOI: 10.1016/j.promfg.2017.07.198.
- NESTEROV, Yurii (1983). „A method for unconstrained convex minimization problem with the rate of convergence $o(1/k^2)$ “. In: *Soviet Mathematics Doklady (übersetzt)*. Bd. 27, S. 372–376.
- NISWAR, Muhammad et al. (2018). „IoT-based Water Quality Monitoring System for Soft-Shell Crab Farming“. In: *Proceedings of the 2018 IEEE International Conference on Internet of Things and Intelligence System (IOTAIS)*. IEEE. DOI: 10.1109/iotais.2018.8600828.
- NORTON, Tomás, Jim GRANT et al. (2009). „Assessing the ventilation effectiveness of naturally ventilated livestock buildings under wind dominated conditions using computa-

-
- tional fluid dynamics“. In: *Biosystems Engineering* 103.1, S. 78–99. DOI: 10.1016/j.biosystemseng.2009.02.007.
- NORTON, Tomás, Jim GRANT et al. (2010). „Improving the representation of thermal boundary conditions of livestock during CFD modelling of the indoor environment“. In: *Computers and Electronics in Agriculture* 73.1, S. 17–36. DOI: 10.1016/j.compag.2010.04.002.
- NORTON, Tomás, Da-Wen SUN et al. (2007). „Applications of computational fluid dynamics (CFD) in the modelling and design of ventilation systems in the agricultural industry: A review“. In: *Bioresource Technology* 98.12, S. 2386–2414. DOI: 10.1016/j.biortech.2006.11.025.
- OBERKAMPF, William und Timothy TRUCANO (2000). „Validation methodology in computational fluid dynamics“. In: *Fluids 2000 Conference and Exhibit*. American Institute of Aeronautics und Astronautics. DOI: 10.2514/6.2000-2549.
- PARGMANN, Hergen, Dorte EUHAUSEN und Robin FABER (2018). „Intelligent big data processing for wind farm monitoring and analysis based on cloud-technologies and digital twins: A quantitative approach“. In: *Proceedings of the 2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*. IEEE. DOI: 10.1109/icccbda.2018.8386518.
- PATRYL, L. und D. GALERIU (2011). „Statistical Performances measures – models comparison“. In: *Proceedings of the International Atomic Energy Agency IAEA EMRAS II Meeting*. energie atomique — energies alternatives.
- PINKUS, Allan (1999). „Approximation theory of the MLP model in neural networks“. In: *Acta Numerica* 8, S. 143–195. DOI: 10.1017/s0962492900002919.
- PIOCH, Fabian, Jan Hauke HARMENING, Andreas Maximilian MÜLLER et al. (2023). „Turbulence Modeling for Physics-Informed Neural Networks: Comparison of Different RANS Models for the Backward-Facing Step Flow“. In: *Fluids* 8.2, S. 43. DOI: 10.3390/fluids8020043.
- PIOCH, Fabian, Jan Hauke HARMENING und Dieter SCHRAMM (2021). „Internet of Things Communication Protocols for Digital Twins in Dairy Barns: Communication Scheme and Feasibility Study“. In: *Proceedings of the International Agriculture Innovation Conference 2021*. Hrsg. von INTERNATIONAL ASSOCIATION FOR AGRICULTURAL SUSTAINABILITY, S. 115–125.
- PLACKETT, R. L. und J. P. BURMAN (1946). „THE DESIGN OF OPTIMUM MULTIFACTORIAL EXPERIMENTS“. In: *Biometrika* 33.4, S. 305–325. DOI: 10.1093/biomet/33.4.305.
- PSICHOGIOS, Dimitris C. und Lyle H. UNGAR (1992). „A hybrid neural network-first principles approach to process modeling“. In: *AIChE Journal* 38.10, S. 1499–1511. DOI: 10.1002/aic.690381003.

- PURCELL, Warren und Thomas NEUBAUER (2023). „Digital Twins in Agriculture: A State-of-the-art review“. In: *Smart Agricultural Technology* 3, S. 100094. DOI: 10.1016/j.atech.2022.100094.
- PYLIANIDIS, Christos, Sjoukje OSINGA und Ioannis N. ATHANASIADIS (2021). „Introducing digital twins to agriculture“. In: *Computers and Electronics in Agriculture* 184, S. 105942. DOI: 10.1016/j.compag.2020.105942.
- QIAN, Ning (1999). „On the momentum term in gradient descent learning algorithms“. In: *Neural Networks* 12.1, S. 145–151. DOI: 10.1016/s0893-6080(98)00116-6.
- RAISSI, Maziar, Paris PERDIKARIS und George Em KARNIADAKIS (2019). „Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations“. In: *Journal of Computational Physics* 378, S. 686–707. DOI: 10.1016/j.jcp.2018.10.045.
- RAISSI, Maziar, Alireza YAZDANI und George Em KARNIADAKIS (2020). „Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations“. In: *Science* 367.6481, S. 1026–1030. DOI: 10.1126/science.aaw4741.
- REY, Günter Daniel und Karl F. WENDER (2018). *Neuronale Netze*. 3. Aufl. Hogrefe. DOI: 10.1024/85796-000.
- RIEG, Frank, Reinhard HACKENSCHMIDT und Bettina ALBER-LAUKANT (2012). *Finite Elemente Analyse für Ingenieure*. 1. Aufl. Hanser. ISBN: 978-3-446-45639-6.
- ROACHE, Patrick J. (1998). *Verification and validation in computational science and engineering*. Hermosa Publ. ISBN: 978-0913478080.
- RONG, Li, Peter V. NIELSEN, Bjarne BJERG et al. (2016). „Summary of best guidelines and validation of CFD modeling in livestock buildings to ensure prediction quality“. In: *Computers and Electronics in Agriculture* 121, S. 180–190. DOI: 10.1016/j.compag.2015.12.005.
- RONG, Li, Peter V. NIELSEN und Guoqiang ZHANG (2010). „Experimental and Numerical Study on Effects of Airflow and Aqueous Ammonium Solution Temperature on Ammonia Mass Transfer Coefficient“. In: *Journal of the Air & Waste Management Association* 60.4, S. 419–428. DOI: 10.3155/1047-3289.60.4.419.
- RYCK, Tim De, Ameya D. JAGTAP und Siddhartha MISHRA (2023). „Error estimates for physics-informed neural networks approximating the Navier–Stokes equations“. In: *IMA Journal of Numerical Analysis*. DOI: 10.1093/imanum/drac085.
- SAAD, Youcef und Martin H. SCHULTZ (1986). „GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems“. In: *SIAM Journal on Scientific and Statistical Computing* 7.3, S. 856–869. DOI: 10.1137/0907058.
- SAHA, Chayan Kumer, Merike FIEDLER, Christian AMMON et al. (2014). „UNCERTAINTY IN CALCULATING AIR EXCHANGE RATE OF NATURALLY VENTILATED DAIRY

-
- BUILDING BASED ON POINT CONCENTRATIONS“. In: *Environmental Engineering & Management Journal (EEMJ)* 13.9.
- SAHA, Chayan Kumer, Merike FIEDLER, Thomas AMON et al. (2014). „Assessing effects of different opening combinations on airflow pattern and air exchange rate of a naturally ventilated dairy building – A CFD approach“. In: *Proceedings of the International Conference of Agricultural Engineering AgEng*, S. 1–4.
- SAHA, Chayan Kumer, Qianying YI et al. (2020). „Opening Size Effects on Airflow Pattern and Airflow Rate of a Naturally Ventilated Dairy Building—A CFD Study“. In: *Applied Sciences* 10.17, S. 6054. DOI: 10.3390/app10176054.
- SAPOUNAS, A. A., H. J. C. van DOOREN und M. C. J. SMITS (2013). „Natural ventilation of commercial dairy cow houses: Simulating the effect of roof shape using CFD“. In: *Acta Hort.* 1008, S. 221–228.
- SCHMEING, Lucas (2021). „Parametrische CAD-Modellierung eines Nutztierstalls für automatisierte CFD-Berechnungen“. Masterarbeit (unveröffentlicht). Westfälische Hochschule Gelsenkirchen, Bocholt, Recklinghausen.
- SCS (1979). „Terminology for model credibility“. In: *SIMULATION* 32.3, S. 103–104. DOI: 10.1177/003754977903200304.
- SEKAR, Vinothkumar et al. (2022). *Accurate near wall steady flow field prediction using Physics Informed Neural Network (PINN)*. Preprint. DOI: 10.48550/ARXIV.2204.03352.
- SHAFTO, Mike et al. (2010). *DRAFT Modeling, Simulation, Information Technology & Processing Roadmap Technology Area 11*. https://www.nasa.gov/pdf/501321main_TA11-MSITP-DRAFT-Nov2010-A1.pdf. (Abgerufen am 01.02.2023).
- SHAFTO, Mike et al. (2012). *Modeling, Simulation, Information Technology & Processing Roadmap Technology Area 11*. https://www.nasa.gov/sites/default/files/501321main_TA11-ID_rev4_NRC-wTASR.pdf. (Abgerufen am 01.02.2023).
- SIEBERTZ, Karl, David van BEBBER und Thomas HOCHKIRCHEN (2017). *Statistische Versuchsplanung*. 2. Aufl. Springer Berlin Heidelberg. DOI: 10.1007/978-3-662-55743-3.
- SLOB, Naftali und William HURST (2022). „Digital Twins and Industry 4.0 Technologies for Agricultural Greenhouses“. In: *Smart Cities* 5.3, S. 1179–1192. DOI: 10.3390/smartcities5030059.
- SREEDEVI, T. R. und M. B. Santosh KUMAR (2020). „Digital Twin in Smart Farming: A Categorical Literature Review and Exploring Possibilities in Hydroponics“. In: *Proceedings of the 2020 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA)*. IEEE. DOI: 10.1109/accthpa49271.2020.9213235.
- STANLEY, Kieran M. et al. (2018). „Greenhouse gas measurements from a UK network of tall towers: technical description and first results“. In: *Atmospheric Measurement Techniques* 11.3, S. 1437–1458. DOI: 10.5194/amt-11-1437-2018.

- STATISTISCHES BUNDESAMT DEUTSCHLAND (2020). *Stallhaltung, Weidehaltung*. <https://www.destatis.de/DE/Themen/Branchen-Unternehmen/Landwirtschaft-Forstwirtschaft-Fischerei/Produktionsmethoden/Publikationen/Downloads-Produktionsmethoden/stallhaltung-weidehaltung-tb-5411404209004.html>. (Abgerufen am 17.02.2023).
- STERN, Fred et al. (2001). „Comprehensive Approach to Verification and Validation of CFD Simulations—Part 1: Methodology and Procedures“. In: *Journal of Fluids Engineering* 123.4, S. 793–802. DOI: 10.1115/1.1412235.
- SUN, Luning et al. (2020). „Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data“. In: *Computer Methods in Applied Mechanics and Engineering* 361, S. 112732. DOI: 10.1016/j.cma.2019.112732.
- SUTTON, Richard S. (1986). „Two problems with backpropagation and other steepest-descent learning procedures for networks“. In: *Proceedings of the Eighth Annual Conference of the Cognitive Science Society*, S. 823–831.
- TERGAST, Hauke und Heiko HANSEN (2021). *Steckbriefe zur Tierhaltung in Deutschland: Milchkuhe*. Hrsg. von Thünen-Institut für Betriebswirtschaft.
- THOM, E. C. (1959). „The Discomfort Index“. In: *Weatherwise* 12.2, S. 57–61. DOI: 10.1080/00431672.1959.9926960.
- TOMASELLO, Nicoletta et al. (2019). „Development of a CFD Model to Simulate Natural Ventilation in a Semi-Open Free-Stall Barn for Dairy Cows“. In: *Buildings* 9.8, S. 183. DOI: 10.3390/buildings9080183.
- TOMINAGA, Yoshihide et al. (2008). „AIJ guidelines for practical applications of CFD to pedestrian wind environment around buildings“. In: *Journal of Wind Engineering and Industrial Aerodynamics* 96.10-11, S. 1749–1761. DOI: 10.1016/j.jweia.2008.02.058.
- UNITED NATIONS (2019). *World Urbanization Prospects: The 2018 Revision (ST/ESA/SER.A/420)*. United Nations, Department of Economic and Social Affairs, Population Division (UNDESA).
- VAN DRIEST, Edward R. (1956). „On Turbulent Flow Near a Wall“. In: *Journal of the Aeronautical Sciences* 23.11, S. 1007–1011. DOI: 10.2514/8.3713.
- VEGA, Felipe Andrés Obando et al. (2022). „CFD Study of a Tunnel-Ventilated Compost-Bedded Pack Barn Integrating an Evaporative Pad Cooling System“. In: *Animals* 12.14, S. 1776. DOI: 10.3390/ani12141776.
- VERDOUW, Cor et al. (2021). „Digital twins in smart farming“. In: *Agricultural Systems* 189, S. 103046. DOI: 10.1016/j.agsy.2020.103046.
- WANDEL, Nils, Michael WEINMANN und Reinhard KLEIN (2021). „Teaching the incompressible Navier–Stokes equations to fast neural surrogate models in three dimensions“. In: *Physics of Fluids* 33.4, S. 047117. DOI: 10.1063/5.0047428.

-
- WANG, Hongping, Yi LIU und Shizhao WANG (2022). „Dense velocity reconstruction from particle image velocimetry/particle tracking velocimetry using a physics-informed neural network“. In: *Physics of Fluids* 34.1, S. 017116. DOI: 10.1063/5.0078143.
- WANG, Rui et al. (2020). „Towards Physics-informed Deep Learning for Turbulent Flow Prediction“. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM. DOI: 10.1145/3394486.3403198.
- WANG, Xiaoshuai, Guoqiang ZHANG und Christopher Y. CHOI (2018). „Effect of airflow speed and direction on convective heat transfer of standing and reclining cows“. In: *Bio-systems Engineering* 167, S. 87–98. DOI: 10.1016/j.biosystemseng.2017.12.011.
- WILCOX, David C. (1988). „Reassessment of the scale-determining equation for advanced turbulence models“. In: *AIAA Journal* 26.11, S. 1299–1310. DOI: 10.2514/3.10041.
- WILCOX, David C. (2004). *Turbulence modeling for CFD*. 2. Aufl. DCW Industries. ISBN: 1-928729-10-X.
- WILCOX, David C. und I. E. ALBER (1972). „A turbulence model for high speed flows“. In: *Proceedings of the 1972 Heat Transfer and Fluid Mechanics Institute*. Bd. 231. Stanford University Press, S. 252.
- WU, Chenxi et al. (2023). „A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks“. In: *Computer Methods in Applied Mechanics and Engineering* 403, S. 115671. DOI: 10.1016/j.cma.2022.115671.
- XIANG, Zixue et al. (2022). „Self-adaptive loss balanced Physics-informed neural networks“. In: *Neurocomputing* 496, S. 11–34. DOI: 10.1016/j.neucom.2022.05.015.
- XU, Hui, Wei ZHANG und Yong WANG (2021). „Explore missing flow dynamics by physics-informed deep learning: The parameterized governing systems“. In: *Physics of Fluids* 33.9, S. 095116. DOI: 10.1063/5.0062377.
- XU, Shengfeng et al. (2022). „A practical approach to flow field reconstruction with sparse or incomplete data through physics informed neural network“. In: *Acta Mechanica Sinica* 39.3. DOI: 10.1007/s10409-022-22302-x.
- YANG, Hui et al. (2022). „Digital twin key technology on rare earth process“. In: *Scientific Reports* 12.1. DOI: 10.1038/s41598-022-19090-y.
- YAO et al. (2019). „Effect of Mixed-Flow Fans with a Newly Shaped Diffuser on Heat Stress of Dairy Cows Based on CFD“. In: *Energies* 12.22, S. 4315. DOI: 10.3390/en12224315.
- YI, Qianying et al. (2020). „Modelling air change rate of naturally ventilated dairy buildings using response surface methodology and numerical simulation“. In: *Building Simulation* 14.3, S. 827–839. DOI: 10.1007/s12273-020-0697-z.
- YOSHIE, Ryuichiro, Akashi MOCHIDA und Yoshihide TOMINAGA (2006). „CFD prediction of wind environment around a high-rise building located in an urban area“. In: *Proceedings*

- of the Fourth International Symposium on Computational Wind Engineering (CWE2006), S. 16–19.
- YOSHIE, Ryuichiro, Akashi MOCHIDA, Yoshihide TOMINAGA et al. (2007). „Cooperative project for CFD prediction of pedestrian wind environment in the Architectural Institute of Japan“. In: *Journal of Wind Engineering and Industrial Aerodynamics* 95.9-11, S. 1551–1578. DOI: 10.1016/j.jweia.2007.02.023.
- ZHOU, Bo et al. (2019). „Assessment of optimal airflow baffle locations and angles in mechanically-ventilated dairy houses using computational fluid dynamics“. In: *Computers and Electronics in Agriculture* 165, S. 104930. DOI: 10.1016/j.compag.2019.104930.
- ZHU, Qiming, Zeliang LIU und Jinhui YAN (2021). „Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks“. In: *Computational Mechanics* 67.2, S. 619–635. DOI: 10.1007/s00466-020-01952-9.
- ZHU, Yinhao et al. (2019). „Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data“. In: *Journal of Computational Physics* 394, S. 56–81. DOI: 10.1016/j.jcp.2019.05.024.

DuEPublico

Duisburg-Essen Publications online

UNIVERSITÄT
DUISBURG
ESSEN

Offen im Denken

ub

universitäts
bibliothek

Diese Dissertation wird via DuEPublico, dem Dokumenten- und Publikationsserver der Universität Duisburg-Essen, zur Verfügung gestellt und liegt auch als Print-Version vor.

DOI: 10.17185/duepublico/81714

URN: urn:nbn:de:hbz:465-20240314-102616-4



Dieses Werk kann unter einer Creative Commons Namensnennung - Keine Bearbeitungen 4.0 Lizenz (CC BY-ND 4.0) genutzt werden.