



Diplomarbeit

Entwicklung eines Frameworks zur Parameteroptimierung auf Basis von CFD Simulationen am Beispiel des k- ϵ - Turbulenzmodells

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines

Diplom-Ingenieurs

unter der Leitung von

Univ.-Prof. Dr.-Ing. habil. Jens-Uwe Repke

(Fachgebiet Dynamik und Betrieb technischer Anlagen, TU Berlin)

Univ.Prof. Univ.Prof. Dipl.-Ing. Dr.techn. Anton Friedl

(E166 - Inst. f. Verfahrenstechnik, Umwelttechnik und Techn. Biowissenschaften, TU Wien)

Ass.Prof. Dipl.-Ing. Dr.techn. Michael Harasek

(E166 - Inst. f. Verfahrenstechnik, Umwelttechnik und Techn. Biowissenschaften, TU Wien)

eingereicht an der Technischen Universität Wien

Fakultät für Technische Chemie

von

Benjamin Lukitsch

1025868 (066473)

Defreggerstraße 93

1120 Wien

Wien, im Juli 2017

Benjamin Lukitsch

Ich habe zur Kenntnis genommen, dass ich zur Drucklegung meiner Arbeit unter der Bezeichnung

Diplomarbeit

Nur mit Bewilligung der Prüfungskommission berechtigt bin.

Ich erkläre weiteres Eides statt, dass ich meine Diplomarbeit nach den anerkannten Grundsätzen für wissenschaftliche Abhandlungen selbstständig ausgeführt habe und alle verwendeten Hilfsmittel, insbesondere die zugrunde gelegte Literatur, genannt habe.

Weiters erkläre ich, dass ich dieses Diplomarbeitsthema bisher weder im In- noch im Ausland (einer Beurteilerin/einem Beurteiler zur Begutachtung) in irgendeiner Form als Prüfungsarbeit vorgelegt habe und dass diese Arbeit mit der vom Begutachter beurteilten Arbeit übereinstimmt.

Wien, im Juli 2017

Benjamin Lukitsch

Kurzfassung

Ausgangspunkt dieser Masterarbeit sind Untersuchungen des Wärmeübergangs zwischen einem kugelförmigen Partikel und einer ebenen Platte mittels *direkter numerischer Simulation* (DNS). Diese Methode ist im turbulenten Bereich sehr aufwendig. Abhilfe schafft hierbei das k- ϵ -Turbulenzmodell. Die dadurch ermittelten Felder sind jedoch weniger präzise.

Im Zuge dieser Arbeit wurde daher ein Framework entwickelt, welches automatisiert die Strömungsfelder der k- ϵ -Turbulenzmodell-Berechnung mit jenen der DNS abgleicht und somit die k- ϵ -Modellparameter sukzessive optimiert. Der optimierte Parametersatz soll dann die effiziente Gewinnung korrekter Strömungsfelder im turbulenten respektive den Transitionsbereich ermöglichen.

Um physikalische Legitimität und insofern den Gültigkeitsbereich der Parametersätze bewerten zu können, wurde zunächst das theoretische Grundgerüst des k- ϵ -Modells ausgearbeitet und diskutiert. Des Weiteren konnte das praktische Verhalten des k- ϵ -Modells in Hinblick auf die untersuchte Strömung mithilfe einer Sensitivitätsanalyse untersucht werden. Der Vergleich der theoretischen mit den praktischen Erkenntnissen ermöglichte die Identifikation von physikalisch fundierten Parametergrenzen. Außerdem konnte ein geeignetes Optimierungsvorgehen ausgearbeitet werden. Zusätzlich wurden mehrere, leicht abgewandelte Optimierungen unternommen. Die daraus gewonnenen Ergebnisse ermöglichen eine Effizienzsteigerung künftiger Optimierungen. Abschließend wurde die Verbesserung der Strömungsfelder durch die optimierten Parameter an einer leicht abgewandelten Strömung überprüft.

Zusätzlich erlaubt die Implementierung des Frameworks in Zukunft auch eine Optimierung von Prozessparametern.

Abstract

Heat transfer between spherical particles and plane walls is of relevance for many industrial applications. Research done by TU Berlin has focused on the heat transport in a laminar regime. In these cases, heat conduction could be identified as the main heat transport mechanism (Brösigke, Herter, Rädle, & Repke, 2017). The question that then arises is to what extent turbulence influences the quantitative contributions of the different heat transport mechanisms.

To clarify that question, further research has to be conducted on heat transfer between a spherical particle and a plane wall in turbulent regimes. For these cases, the applied method of *direct numerical simulation* (DNS) is computationally expensive. One possibility to reduce the computational cost is the use of *Reynolds-averaged Navier-Stokes* (RANS) equations

These equations contain the Reynolds stress tensor, which can be described by different turbulence models. Within the scope of this research, the k - ε turbulence model was chosen for its robustness. DNS and k - ε turbulence models produce qualitatively similar results. Nevertheless, quantitative differences of up to 50% can be reported. By optimizing the model parameters, a quantitative representation of DNS data could be attained.

Therefore, the aim of this project is to develop an automatically operating interface, which compares the DNS and k - ε turbulence model results and numerically optimizes the model parameters in respect of the velocity field \mathbf{U} , the turbulent kinetic energy k and the dissipation rate ε .

In order to evaluate the physical legitimacy of parameter sets - as well as their application boundaries - the basic, theoretical principles of the k - ε model were compiled and discussed. The findings of this research were compared with sensitivity analyses. These were carried out to gain a practical understanding of the k - ε model with regard to the examined flow. By comparing theoretical and practical correlations, physically justified constraints for the parameter optimization were developed. Additionally, an appropriate optimization procedure was selected. Several modified optimizations have been executed to increase the efficiency of future optimizations. Finally, the improvement of the fields was checked by applying the optimized parameters on a similar flow.

Besides model parameter optimization, the interface will be able to facilitate process parameter optimization for future applications.

Inhaltsverzeichnis

Inhaltsverzeichnis

Nomenklatur	ix
1 Einleitung.....	1
1.1 Motivation und Zielsetzung	1
1.2 Gliederung	2
2 Theorie -Turbulenzmodellierung.....	4
2.1 Turbulente Strömungen.....	4
2.2 Navier-Sokes Gleichung	6
2.3 Direkte numerische Simulation	7
2.3.1 Theorie der Energiekaskade	7
2.3.2 Kolmogorov Skalen	8
2.3.3 Berechnungsaufwand DNS	9
2.4 Gemittelte Navier-Stokes Gleichungen.....	10
2.4.1 Gemittelte Massenerhaltung	12
2.4.2 Gemittelte Impulserhaltung	12
2.4.3 Gemittelte Erhaltungsgleichung eines Skalars.....	13
2.5 k- ϵ -Modell	13
2.5.1 Zugrundeliegende Theorien	13
2.5.2 Transportgleichungen für k und ϵ	17
2.5.3 Diskussion des k- ϵ -Modells	33
3 Theorie - Optimierung	39
3.1 Lokale Optimierung	39
3.1.1 Newtonverfahren.....	39
3.1.2 Gedämpftes Newtonverfahren	39
3.1.3 Quasi-Newtonverfahren.....	40

Inhaltsverzeichnis

3.2	Globalisierungs Strategien	42
3.2.1	Hammersley-Verteilung.....	42
4	Umsetzung.....	44
4.1	Strömungssimulation.....	44
4.1.1	Strömungssimulator - OpenFOAM.....	44
4.1.2	Strömung.....	45
4.1.3	Rechengitter	45
4.1.4	k- ϵ -Simulation.....	46
4.1.5	Sensitivitätsanalyse	50
4.2	Optimierung.....	63
4.2.1	Summe der Fehlerquadrate.....	63
4.2.2	Optimierer - <code>scipy.optimize.minimize()</code>	64
4.3	Framework.....	64
4.3.1	Framework - lokale Optimierer	64
4.3.2	Framework - Hammersley-Verteilung	69
5	Ergebnisse und Diskussion.....	70
5.1	Hammersley-Verteilung.....	70
5.2	Lokale Optimierung.....	73
5.2.1	Reduktion Optimierungsvariablen	73
5.2.2	Zielfunktion.....	74
5.2.3	Rauschen der Zielfunktion	77
5.2.4	Verhalten bei Instabilität.....	83
5.3	Auswahl der Optimierungsvariablen.....	86
5.4	Optimierte Strömungsfelder	90
5.5	Übertragbarkeit.....	93
5.6	Eignung k- ϵ -Model	95

Inhaltsverzeichnis

6	Ausblick	97
6.1	Hammersley-Verteilung	97
6.2	Verhalten bei Instabilität	97
6.3	Auswahl der Optimierungsvariablen	98
6.4	Überprüfung der Übertragbarkeit	98
6.5	Anwendung anderer Turbulenzmodelle	99
7	Konklusion.....	100
8	Literaturverzeichnis	101
	Anhang.....	104
Anhang A	Allgemeiner Aufbau Framework	104
Anhang B	Aufbau Framework lokale Optimierung	107
Anhang C	Aufbau Framework Hammersley-Verteilung	111
Anhang D	OpenFOAM-Postprocessing-Tools	114
Anhang E	Bedienungsanleitung lokale Optimierung	117
Anhang F	Bedienungsanleitung Hammersley-Verteilung.....	118

Abbildungsverzeichnis

Abbildung 2.1	Turbulenter Strahl mit $Re \approx 2300$	4
Abbildung 2.2	Zeitliche Entwicklung der Axialgeschwindigkeit	5
Abbildung 2.3	Schematische Darstellung der Reynolds Decomposition	11
Abbildung 2.4	Schematische Darstellung des Prandtlschen Mischungsweges	16
Abbildung 2.5	Verlauf von $\nu T \varepsilon k^2 = C \mu$ in Abhängigkeit zum dimensionslosen Wandabstand y^+	18
Abbildung 2.6	Abklingende Reynoldssche Spannungen einer turbulenten Gitterströmung.....	24
Abbildung 2.7	Schematische Darstellung der Scherströmung	26
Abbildung 2.8	Reynoldssche Spannungen über dem axialen Abstand zum Gitter	26
Abbildung 2.9	Skizze der Strömung durch den rechteckigen Kanal	28
Abbildung 2.10	Verhältnis von Produktion zu Dissipationsrate	29
Abbildung 2.11	Turbulente kinetische Energie und Reynoldssche Spannungen	29

Abbildungsverzeichnis

Abbildung 2.12 Profile der Anisotropen Reynoldsschen Spannungen	30
Abbildung 2.13 Turbulente Viskosität in Abhängigkeit von y^+ in einer Kanalströmung	35
Abbildung 3.1 Hammersley-Verteilung im zweidimensionalen Raum.....	43
Abbildung 4.1 Rechengitter der k - ε -Simulation.....	46
Abbildung 4.2 Betrag des gemittelten Geschwindigkeitsfeldes	48
Abbildung 4.3 turbulente kinetische Energie	49
Abbildung 4.4 erhöhte Skalierung der turbulenten kinetischen Energie.....	49
Abbildung 4.5 Dissipationsrate	50
Abbildung 4.6 Einfluss von $C\varepsilon 1$ auf das Feld von k	51
Abbildung 4.7 Einfluss von $C\varepsilon 1$ auf das Feld von ε	51
Abbildung 4.8 Einfluss von $C\varepsilon 1$ auf \mathbf{U}	52
Abbildung 4.9 Einfluss von $C\varepsilon 2$ auf das Feld von k	53
Abbildung 4.10 Einfluss von $C\varepsilon 1$ auf das Feld von ε	53
Abbildung 4.11 Einfluss von $C\varepsilon 1$ auf \mathbf{U}	54
Abbildung 4.12 Einfluss von $\sigma\varepsilon$	55
Abbildung 4.13 Einfluss von $\sigma\varepsilon$ auf das Feld von ε	55
Abbildung 4.14 Einfluss von $\sigma\varepsilon$ auf \mathbf{U}	56
Abbildung 4.15 Die drei möglichen Berechnungsausgänge	57
Abbildung 4.16 k - $C\varepsilon i$ -Sensitivitätenfelder	58
Abbildung 4.17 k - σk -Sensitivitätenfelder	59
Abbildung 4.18 k - $C\mu$ -Sensitivitätenfeld	59
Abbildung 4.19 k - $\sigma\varepsilon$ -Sensitivitätenfelder	60
Abbildung 4.20 ε - $C\varepsilon i$ -Sensitivitätenfelder	61
Abbildung 4.21 ε -Sensitivitätenfeld - Detailplot.....	61
Abbildung 4.22 \mathbf{U} - Ci -Sensitivitätenfelder	62
Abbildung 4.23 Schematische Darstellung des Frameworks	65
Abbildung 4.24 UML-Aktivitätsdiagramm des Frameworks.....	66
Abbildung 5.1 SFQs Hammersley-Untersuchung 1.....	71
Abbildung 5.2 SFQs Hammersley-Untersuchung 2.....	71
Abbildung 5.3 Überprüfung Framework.....	74

Abbildungsverzeichnis

Abbildung 5.4 Dreidimensionale Darstellung der Zielfunktion.....	76
Abbildung 5.5 Zweidimensionale Darstellung der Zielfunktion.....	77
Abbildung 5.6 Berechnungsverläufe der Optimierer	78
Abbildung 5.7 Einfluss des Rechenweges auf SFQs.....	80
Abbildung 5.8 Änderung der Residuenverläufe bei wiederholter Berechnung	81
Abbildung 5.9 Verläufe der SFQs für den Standardparametersatz bei Reduktion der Residuen	82
Abbildung 5.10 Verlauf der SFQs bei ungeeigneter Schrittweitenwahl.	83
Abbildung 5.11 Test des Frameworks in Bereichen nahe der Stabilitätsgrenze.....	83
Abbildung 5.12 Verläufe der SFQs bei einer Optimierung	84
Abbildung 5.13 Anpassung von $\sigma\varepsilon$ zur Erweiterung des stabilen Bereiches.....	86
Abbildung 5.14 Optimierung von $C\varepsilon1$ und $C\varepsilon2$, Optimierungsbereich: $Pi, Standard1 \pm 0.9$	87
Abbildung 5.15 Optimierung aller Parameter, Optimierungsbereich: $Pi, Standard1 \pm 0.9$	88
Abbildung 5.16 Optimierung aller Parameter, Optimierungsbereich: $Pi, Standard1 \pm 0.05$	89
Abbildung 5.17 Optimierte Feld von ε	91
Abbildung 5.18 Optimierter Betrag des gemittelten Geschwindigkeitsfeldes	92
Abbildung 5.19 Optimierte Feld der turbulenten kinetischen Energie k	93
Abbildung 5.20 Stichprobenartige Nachrechnung einzelner Punkte hinsichtlich Übertragbarkeit.....	95

Tabellenverzeichnis

Tabelle 2.1 Experimentell generierte Ergebnisse für eine turbulente, homogene Scherströmung	27
Tabelle 2.2 Auszug experimentell bestimmter Daten für AT und BT , sowie berechnete Werte für PrT	33
Tabelle 2.3 Entwicklung der k- ε -Modellparameter.....	38
Tabelle 4.1 Startbedingungen der k- ε -Simulation	47
Tabelle 4.2 Größenordnung der unterschiedlichen Sensitivitäten.	63
Tabelle 5.1 Vom Framework bestimmte Zielfunktionswerte einer Optimierungsiteration.	70
Tabelle 5.2 Parametersätze mit niedrigeren SFQs als der Standardparametersatz.	72
Tabelle 5.3 Vom Framework bestimmte Zielfunktionswerte einer Optimierungsiteration.	79
Tabelle 5.4 Vergleich der optimierten Parameter.....	90
Tabelle 5.5 Vergleich der SFQs bei Standardparametern.....	94

Nomenklatur

Lateinische Großbuchstaben

A_T	Kármán-Konstante des logarithmischen Wandgesetzes der Temperatur
B	Konstante des logarithmischen Wandgesetzes
B_T	Konstante des logarithmischen Wandgesetzes der Temperatur
C	Integrationskonstante
$C_{\varepsilon 1}, C_{\varepsilon 2}$	Konstanten der modellierten Transportgleichung für ε
C_μ	Konstante für die turbulente Viskosität im k- ε -Modell
D_N	Diskrepanz
D/Dt	Substantielle Ableitung ($\partial/\partial t + \mathbf{U} \cdot \nabla$)
$\bar{D}/\bar{D}t$	mittlere Substantielle Ableitung ($\partial/\partial t + \langle \mathbf{U} \rangle \cdot \nabla$)
$E(\mathbf{x}, t)$	kinetische Energie ($1/2 \mathbf{U} \cdot \mathbf{U}$)
$\bar{E}(\mathbf{x}, t)$	kinetische Energie der gemittelten Strömung ($1/2 \langle \mathbf{U} \rangle \cdot \langle \mathbf{U} \rangle$)
\mathbf{H}_k	approximierte Hessematrix
\mathbf{I}	Einheitsmatrix
\mathbf{P}	isotrope Spannungen, Druck
P_i	beliebiger k- ε -Modellparameter
\mathcal{P}	Produktion, Produktionsrate von k
Pr	Prandtlzahl (ν/a)
Pr_T	turbulente Prandtlzahl (ν_T/Γ_T)
Re	Reynoldszahl (ul/ν)
Re_0	Reynoldszahl der größten Wirbel ($u_0 l_0/\nu$)
Re_K	Kugel-Reynoldszahl ($U_{ein} d_K/\nu$)
$S_i(\mathbf{x}, P_i, \Delta P_i)$	Sensitivitätenfeld des Parameters P_i
S_{ij}	Scherraten Tensor ($1/2 (\partial U_i/\partial x_j + \partial U_j/\partial x_i)$)
\bar{S}_{ij}	mittlerer Scherraten Tensor ($1/2 (\partial \langle U_i \rangle/\partial x_j + \partial \langle U_j \rangle/\partial x_i)$)
\mathcal{S}	Scherrate in einfachen Scherströmungen ($\mathcal{S} = \partial \langle U_i \rangle/\partial x_j$)
T	Temperatur
T_τ	viskose Temperatur ($q_w/(\rho c_p u_\tau)$)
T^+	dimensionslose Temperatur im Bereich des logarithmischen Wandgesetzes ($(T_w - T)/T_\tau$)
\mathcal{T}	Rate mit der k auf immer kleinere Wirbel übertragen wird

Nomenklatur

$\mathbf{U}(\mathbf{x}, t)$	Vektorfeld der Geschwindigkeit
$\langle \mathbf{U} \rangle(\mathbf{x}, t)$	Vektorfeld der mittleren Geschwindigkeit
U_{ein}	Eintrittsrandbedingung der Geschwindigkeit

Lateinische Kleinbuchstaben

a	Temperaturleitfähigkeit
a_{eff}	effektive Temperaturleitfähigkeit
a_{ij}	anisotroper Reynoldsscher Spannungstensor $(\langle u_i u_j \rangle - 2/3 k \delta_{ij})$
a_T	turbulente Temperaturleitfähigkeit
c_p	isobare spezifische Wärmekapazität
d_K	Kugel-/Partikeldurchmesser
d_k	Suchrichtung des Optimierers
f	allgemeine Funktion oder im Optimierungskontext Zielfunktion
f_μ	Dämpfungsfunktion
g	spezifische Gravitationskraft
$\mathbf{g}(\mathbf{x})$	nichtlineares Gleichungssystem
h	Maschenweite
k	turbulente kinetische Energie
k_P	turbulente kinetische Energie am Zellmittelpunktes der ersten Zellschicht
l	Untergrenze (engl. lower bound) der Optimierungsvariablen
l^*	Längenskalierung für die turbulente Viskosität
l_0	Längenskalierung der größten Wirbel
l_m	Prandtlscher Mischungsweg
$p(\mathbf{x}, t)$	modifizierter Druck, zusammengesetzt aus isotropen Spannungen \mathbf{P} und statischem Druck ρg
$p'(\mathbf{x}, t)$	fluktuierender modifizierter Druck
q	Wärmestromdichte
q_W	Wärmestromdichte an der Wand
s_{ij}	fluktuierender Scherratenstensor $(1/2 (\partial u_i / \partial x_j + \partial u_j / \partial x_i))$
s_k	Iterationsschritt des Newtonverfahrens $(\alpha_k \cdot d_k)$
t	Zeit
u	Obergrenze (engl. upper bound) der Optimierungsvariablen
$\mathbf{u}(\mathbf{x}, t)$	Vektorfeld der fluktuierenden Geschwindigkeit
u_0	Geschwindigkeitsskalierung der größten Wirbel

Nomenklatur

u_η	Geschwindigkeitsskalierung der Kolmogorov-Wirbel
u_τ	viskose Geschwindigkeitsskalierung ($Re = u_\tau \delta_v / \nu = 1$) des logarithmischen Wandgesetzes
u^+	mittlere Geschwindigkeit des logarithmischen Wandbereiches normiert durch viskose Geschwindigkeitsskalierung u_τ
$\mathbf{v}_k, \mathbf{w}_k$	Vektoren zur Aufdatierung der Hessematrix
\mathbf{x}	kartesische Koordinaten oder im Optimierungskontext Optimierungsvariablen
\mathbf{x}^*	lokales Minimum
y^+	dimensionsloser Wandabstand des logarithmischen Wandgesetzes, normiert durch Rauigkeitslänge δ_v
y_p	Abstand des Zellmittelpunktes der ersten Zellschicht von der Wand

Griechische Großbuchstaben

Γ	molekulare Diffusivität
Γ_{eff}	effektive Diffusivität ($\Gamma + \Gamma_T$)
Γ_T	turbulente Diffusivität
$\Delta C_{\varepsilon i}$	Gradientenschrittweite von $C_{\varepsilon 1}$ oder $C_{\varepsilon 2}$
ΔP_i	Gradientenschrittweite eines beliebigen k- ε -Modellparameters
Ψ	Gravitationspotential

Griechische Kleinbuchstaben

α	Dämpfungsparameter
δ	halbe Höhe des Kanals
δ_{ij}	Kronecker-Delta ($\delta_{i(j=i)} = 1, \delta_{i(j \neq i)} = 0$)
δ_v	Rauigkeitslänge, viskose Geschwindigkeitsskalierung ($Re = u_\tau \delta_v / \nu = 1$) des logarithmischen Wandgesetzes
ε	Dissipationsrate der turbulenten kinetischen Energie k ($2\nu \langle s_{ij} s_{ij} \rangle$)
$\bar{\varepsilon}$	Dissipationsrate der mittleren kinetischen Energie \bar{E} ($2\nu \langle S_{ij} S_{ij} \rangle$)
η	Kolmogorov-Längenskala
κ	Kármán-Konstante des logarithmischen Wandgesetzes
μ	dynamische Viskosität
ν	kinematische Viskosität (μ / ρ)
ν_{eff}	effektive Viskosität ($\nu + \nu_T$)
ν_T	turbulente Viskosität
ρ	Dichte

Nomenklatur

σ_k	turbulente Prandtlzahl der turbulenten kinetischen Energie k
σ_T	turbulente Prandtlzahl der Temperatur T
σ_ε	turbulente Prandtlzahl der Dissipationsrate ε
τ	turbulente Zeitskala (k/ε)
τ_0	Zeitskala der größten Wirbel
τ_{ij}	Spannungstensor
τ_W	Wandschubspannung
τ_η	Kolmogorov-Zeitskala
$\tilde{\tau}$	Schubspannung
$\phi(\boldsymbol{x}, t)$	allgemeines skalares Strömungsfeld
ϕ_p	Hammersleyfolge

Abkürzungen

DNS	direkte numerische Simulation
RANS	Reynolds-averaged Navier-Stokes equations (dt. Reynolds-gemittelte Navier-Stokes Gleichungen)
SFQ	Summe der Fehlerquadrate

1 Einleitung

1.1 Motivation und Zielsetzung

Das theoretische Verständnis des Wärmeübergangs zwischen Partikeln und ebenen Platten kommt oftmals bei der Auslegung von Wirbelschichtsystemen zum Tragen. Diese werden aufgrund ihrer einzigartigen Eigenschaften in vielen industriellen Anwendungen herangezogen. Zu diesen Eigenschaften zählen der exzellente Wärmeübergang, die hohe, für katalytische Reaktionen wesentliche spezifische Oberfläche, sowie die einfache Handhabung und gute Durchmischung der Feststoffpartikel (Tebanian et al., 2015).

Die Bedeutsamkeit von Wirbelschichtsystemen tritt klar hervor, wenn die Stellung von *Fluid Catalytic Cracking* (FCC) in der Raffinerietechnik vergegenwärtigt wird. FCC dient, als eines der ältesten, kommerziell genutzten Wirbelschichtverfahren, der Erhöhung der Benzinausbeute aus Rohöl. Dementsprechend wurden weltweit 400 FCC-Anlagen in Raffinerien integriert (Speight, 2014). In diesen Raffinerien werden etwa 20% des eingehenden Rohöls mittels FCC prozessiert (Treese, Pujadó, & Jones, 2015).

Angesichts der technischen und ökonomischen Relevanz von Wirbelschichtsystemen sind fundamentale Untersuchungen des Wärmeübergangs zwischen einem sphärischen Partikel und einer ebenen Platte von grundlegender Bedeutung. Dieser Aufgabe verschrieb sich die TU Berlin mit einer Forschungsarbeit von Brösigke, Herter, Rädle und Repke (2017). Mittels CFD konnte hierbei für das laminare Regime Wärmeleitung im Gas-Zwischenraum als dominierender Wärmetransportmechanismus identifiziert werden. Die Ergebnisse dieser Forschungsarbeit ließen jedoch die Frage aufkommen, welchen Einfluss aufkommende Turbulenz auf die quantitativen Beiträge der verschiedenen Transportmechanismen hat.

Zur Klärung wurden weitere Untersuchungen des oben beschriebenen Wärmeübergangsproblems für den turbulenten, respektive den Transitionsbereich eingeleitet. Die von Brösigke et al. (2017) angewandte Methode der *direkten numerischen Simulation* (DNS) ist für den turbulenten Bereich sehr rechenaufwendig. Eine in dieser Masterarbeit verfolgte Möglichkeit den Rechenaufwand zu reduzieren, ist die Anwendung von *Reynolds-gemittelten Navier-Stokes-Gleichungen* (RANS).

1.2 Gliederung

Der darin auftretende Reynoldssche-Spannungstensor kann durch diverse Turbulenzmodelle beschrieben werden. Im Rahmen dieser Arbeit wurde dafür das k - ε -Turbulenzmodell aufgrund seiner Robustheit gewählt.

Der erste Vergleich zwischen DNS und k - ε -Turbulenzmodell zeigt eine gute qualitative Übereinstimmung, wobei quantitativ bis zu 50% Unterschied zu beobachten ist. Durch eine Anpassung der Modell-Parameter, mittels numerischer Optimierung, soll auch eine quantitative Repräsentation der DNS ermöglicht werden. Ähnliche Forschungsarbeiten wurden hierbei bereits von Edeling, Cinnella, Dwight und Bilj (2014) sowie Souza, Janiga und Thèvenin (2017) durchgeführt.

Zielsetzung dieser Masterarbeit ist insofern die Entwicklung eines Frameworks, welches automatisiert die Ergebnisse der DNS mit denen des k - ε -Turbulenzmodells vergleicht und die Modellparameter bzgl. des gemittelten Geschwindigkeitsfeldes $\langle \mathbf{U} \rangle$ der turbulenten kinetischen Energie k und der Dissipationsrate ε numerisch optimiert. Die Implementierung des Frameworks soll in Zukunft auch eine Optimierung von Prozessparametern zulassen.

1.2 Gliederung

Diese Arbeit beginnt mit einer Einführung in die grundlegenden Theorien zur Turbulenz. Auf deren Basis werden die Vorteile von RANS-Modellen erklärt. Im Anschluss erfolgt eine detaillierte Erläuterung des k - ε -Turbulenzmodells. Hierzu werden die Modellgleichungen hergeleitet und ihre Herkunft erörtert. Zusätzlich wird die Bestimmung der Modellparameterwerte über simple Benchmark-Strömungen dargelegt.

Darauffolgend wird der theoretische Hintergrund der Optimierung beleuchtet. Hierbei wird das Funktionsprinzip des herangezogenen lokalen Optimierers (*L-BFGS-B*) geschildert und auf die verwendete Globalisierungsstrategie (*Hammersley-Sampling*) eingegangen.

Im Anschluss an den Theorieteil wird die praktische Umsetzung erläutert. Zunächst wird der verwendete Strömungssimulator *OpenFOAM* beschrieben. Danach werden die wesentlichen Erkenntnisse einer im Zuge dieser Arbeit durchgeführten Sensitivitätsanalyse zusammengefasst. Der Einfluss der Parameteränderungen auf die Strömungsfelder wird hierbei erörtert und mit dem Einfluss der Parameter auf die Felder der Benchmark-Strömungen verglichen. Dabei beobachtete Korrelationen werden

1.2 Gliederung

aufgezeigt. Aufbauend auf theoretischen Überlegungen und praktischen Erkenntnissen werden Rückschlüsse auf ein geeignetes Optimierungsvorgehen gezogen. Anschließend wird der Optimierer (*scipy.optimize.minimize()*) vorgestellt. Zuletzt wird das Framework, als Schnittstelle zwischen Strömungssimulator und Optimierer, in seinem prinzipiellen Aufbau beschrieben und die Implementierung im Detail erörtert.

In dem Kapitel Ergebnisse und Diskussion werden anhand der Resultate zweier Hammersley-Verteilungen anwendungsbezogen Stabilitätsgrenzen diskutiert. Daraufgehend wird die eingehende Überprüfung des Frameworks mittels eines Test-Cases beschrieben. Anhand der dabei entstandenen umfangreichen Daten werden Parallelen zwischen der Theorie und den Benchmark-Strömungen gezogen. Des Weiteren werden theoretisch entwickelte und daher physikalisch fundierte Parametergrenzen diskutiert. Außerdem werden die Ergebnisse mehrerer leicht abgewandelter lokaler Optimierungen vorgestellt und diese den Ergebnissen der Sensitivitätsanalyse gegenübergestellt. Die Option einer Reduktion der Optimierungsvariablen (k - ε -Modellparameter) wird angeführt. Abschließend wird auf die Übertragbarkeit der gewonnenen Parameter auf andere ähnliche Strömungen eingegangen. Hierfür wurde die Verbesserung der Strömungsfelder durch die optimierten Parameter an einer leicht abgewandelten Strömung überprüft.

Im Ausblick werden die theoretisch und praktisch gewonnenen Erkenntnisse zusammengetragen. Auf Basis derer wird ein effizientes und konsequentes Optimierungsvorgehen zur Gewinnung von physikalisch legitimen und daher übertragbaren Parametersätzen entworfen.

2 Theorie -Turbulenzmodellierung

Der Theorieteil über Turbulenz und deren Modellierung hält sich an dem Standardwerk *Turbulent Flows* von Pope (2000). Dieses Buch erörtert detailliert das Verhalten turbulenter Strömungen und deren mathematische Beschreibung. Auf Basis dieser Ausführungen, werden darin die gängigen Turbulenzmodelle eingehend beschrieben. Zudem wird konsequent auf weiterführende Literatur verwiesen. An dieser Stelle soll auch der digitale Anhang, die Homepage von Pope (2013) Erwähnung finden. Von dieser können wichtige Herleitungen und die meisten der in Pope (2000) gezeigten Abbildungen entnommen werden.

2.1 Turbulente Strömungen

Turbulente Strömungen treten in vielen unterschiedlichen Ausprägungsformen im Alltag auf. Bei der Beobachtung eines schnellfließenden Baches können jedoch bereits die grundlegenden, charakteristischen Eigenschaften turbulenter Strömungen festgestellt werden. Sie sind instationär, unregelmäßig, scheinbar chaotisch und weisen unterschiedliche Größenskalen auf. Auch in Laborbedingungen sind diese Eigenschaften, zum Beispiel an einem turbulenten Strahl (siehe Abbildung 2.1), erkennbar.



Abbildung 2.1 Turbulenter Strahl mit $Re \approx 2300$ (Van Dyke, 1982)

Sie ermöglichen, im Vergleich zu laminaren Strömungen, einen erhöhten Wärme-, Stoff-, und Impulstransport in turbulenten Strömungen. Letztere sind daher in vielen technischen Anwendungen relevant. Die Bestimmung ihrer Geschwindigkeitsfelder $\mathbf{U}(\mathbf{x}, t)$ ist insofern, für die Auslegung von Apparaturen, von großer Bedeutung.

2.1 Turbulente Strömungen

Die grundlegenden Eigenschaften turbulenter Strömungen sind jedoch eng mit denen der Geschwindigkeitsfelder verknüpft. Die darin enthaltenen Geschwindigkeiten \mathbf{U} variieren stark bei Änderung des Ortes \mathbf{x} und der Zeit t . In Abbildung 2.2 wird die zeitliche Variation der Axialgeschwindigkeit $U(t)$ auf der Symmetrieachse des turbulenten Strahles dargestellt. Ihr Verlauf fluktuiert, keiner erkennbaren Regelmäßigkeit folgend, um einen mittleren Wert.

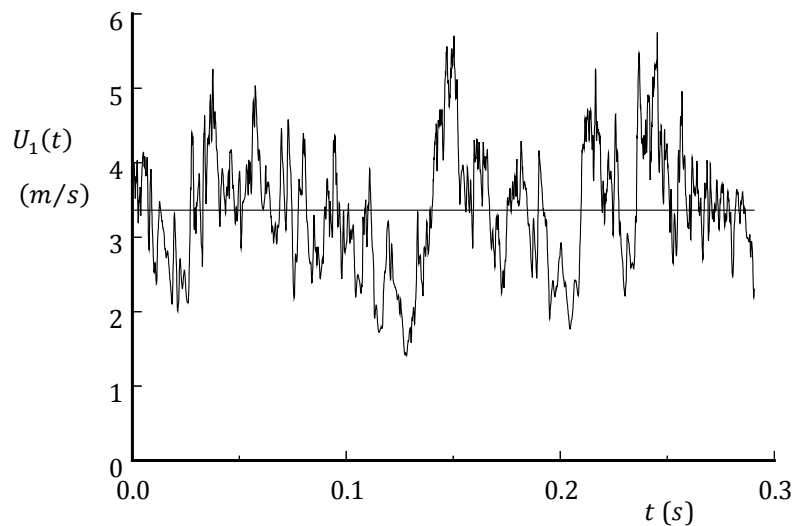


Abbildung 2.2 Zeitliche Entwicklung der Axialgeschwindigkeit $U_1(t)$ auf der Symmetrieachse eines turbulenten Strahles (Tong & Warhaft, 1995).

Durch diese räumlichen und zeitlichen Variationen gestaltet sich die Bestimmung der Geschwindigkeitsfelder als eine große Herausforderung. Sie erfolgt letzten Endes durch die Lösung der den Geschwindigkeitsfeldern zugrundeliegenden Impulserhaltung.

2.2 Navier-Sokes Gleichung

Die Impulserhaltung beschreibt die Beschleunigung eines Fluidelements anhand der angreifenden Oberflächen und Volumenkräfte. Die Oberflächenkräfte können dabei durch den Spannungstensor τ_{ij} abgebildet werden. Die wesentlichste Volumenkraft ist die Schwerkraft mit ihrem Potential $\Psi = \mathbf{g}$. Die Impulsbilanz lässt sich daher in Form des folgenden Kräftegleichgewichts ausformulieren:

$$\rho \frac{DU_j}{Dt} = \frac{\partial \tau_{ij}}{\partial x_i} - \rho \frac{\partial \Psi}{\partial x_j} \quad 2.2.1$$

Der Spannungstensor von Newtonschen Fluiden wird vom Newtonschen Reibspannungsansatz beschrieben.

$$\tau_{ij} = -P\delta_{ij} + \mu S_{ij} = -P\delta_{ij} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad 2.2.2$$

Dieser setzt sich aus den isotropen Spannungen, dem Druck $P\delta_{ij}$ und den anisotropen Spannungen, den Schubspannungen S_{ij} , zusammen. Kronecker-Delta (δ_{ij}) ist dabei wie üblich definiert ($\delta_{i(j=i)} = 1, \delta_{i(j \neq i)} = 0$) und der Spannungstensor S_{ij} symmetrisch $S_{ij} = S_{ji}$.

Für inkompressible Fluide gilt aufgrund der Kontinuitätsgleichung $\partial u_i / \partial x_i = 0$. Die Impulserhaltung für Newtonsche, inkompressible Fluide kann daher wie folgt angeschrieben werden:

$$\rho \frac{DU_j}{Dt} = \mu \frac{\partial^2 u_j}{\partial x_i \partial x_i} - \frac{\partial P}{\partial x_j} - \rho \frac{\partial \Psi}{\partial x_j} \quad 2.2.3$$

Zusammenfassung der isotropen Spannungen P mit dem statischen Druck $\rho \mathbf{g}$ zu dem modifizierten Druck p liefert nach Umformung die *Navier-Stokes-Gleichung*:

$$\frac{DU_j}{Dt} = \nu \nabla^2 u_j - \frac{1}{\rho} \nabla p \quad 2.2.4$$

Sie ist aufgrund der Nichtlinearität des konvektiven Terms $\nu \nabla^2 u_j$ nicht analytisch lösbar. Es müssen insofern numerische Lösungsansätze verfolgt werden.

2.3 Direkte numerische Simulation

In der *direkten numerischen Simulation* (DNS) werden die miteinander verkoppelten Erhaltungsgleichungen für Masse, Impuls und Energie, auch *Navier-Stokes-Gleichungen* genannt, direkt numerisch gelöst. Sie ist die konzeptionell einfachste Methode zur Simulation von Strömungen und ist bezüglich Genauigkeit und Informationsgehalt konkurrenzlos. Für eine physikalisch sinnvolle Lösung müssen aber zumindest die Kolmogorov Skalen räumlich aufgelöst werden. Dies verursacht jedoch erheblichen Rechenaufwand. Für eine Beschreibung und Abschätzung der Kolmogorov Skalen und dem damit einhergehenden Rechenaufwand wird deshalb zunächst die Theorie der Energiekaskade erläutert.

2.3.1 Theorie der Energiekaskade

Die Theorie der Energiekaskade geht von einer voll ausgebildeten, turbulenten Strömung mit hohen Reynoldszahlen $Re \equiv u(l)l/\nu$ aus. Sie setzt sich aus Wirbeln in unterschiedlichen Größen l zusammen. Die Wirbel weisen die charakteristische Geschwindigkeit $u(l)$, Energie in der Größenordnung von $u^2(l)$ und die charakteristische Zeitskala $\tau = l/u(l)$ auf.

Nach der Theorie sind große Wirbel instabil und brechen in immer kleinere Wirbel auf. Dabei wird ihre kinetische Energie auf die nächst kleineren Wirbel übertragen. Dieser Vorgang wiederholt sich, wodurch die Reynoldszahl $Re(l)$ sukzessive sinkt. Am Ende dieser Energieübertragungskaskade stehen demnach die kleinsten Wirbel, in denen viskose Effekte dominieren und die kinetische Energie durch Reibung dissipiert wird (Richardson, 1922).

Die Rate mit der die kinetische Energie auf immer kleinere Wirbel übertragen wird \mathcal{T} , ist nach der Theorie der Energiekaskade konstant.

Da die Dissipation am Ende der Kaskade steht, kann ihre Rate ε durch den ersten Teilschritt, also den Zerfall der größten Wirbel abgeschätzt werden. Deren Energie- und Zeitskala beträgt $u_0^2(l_0)$ bzw. $\tau_0 = l_0/u_0(l_0)$. Die Dissipationsrate kann daher wie folgt angeschrieben werden:

$$\varepsilon = \mathcal{T} = \frac{u_0^2}{\tau_0} = \frac{u_0^3}{l_0} \tag{2.3.1}$$

2.3 Direkte numerische Simulation

2.3.2 Kolmogorov Skalen

Die Größenskala jener Wirbel, in denen die kinetische Energie dissipiert, wurde von Kolmogorov in den 1940er Jahren beschrieben (Kolmogorov, 1991)¹.

Er ging davon aus, dass die größten Wirbel anisotrop sind, wobei deren Strömungsrichtung und Form durch die Randbedingungen und den Verlauf der makroskopischen Strömung bestimmt werden. Im Zuge eines sukzessiven chaotischen Zerfalls in kleinere Wirbel, geht nach Kolmogorov die Information über die Strömungsrichtung und den makroskopischen Randbedingungen verloren. Kleine Wirbel liegen insofern isotrop vor. Es erscheint demnach plausibel davon auszugehen, dass die Ausprägung kleinskaliger Strömungsstrukturen von makroskopischen Bedingungen unbeeinflusst bleibt.

Zwei in dem Zerfallsvorgang dominierende Phänomene sind:

- i. Die dabei erfolgende Übertragung der kinetischen Energie, zwischen unterschiedlich großen Wirbeln (\mathcal{T}).
- ii. Die viskose Dissipation in den kleinsten Wirbeln.

Der Zustand der kleinskaligen Strömungsstrukturen lässt sich daher laut Kolmogorov durch die Energietransportrate \mathcal{T} und die Viskosität ν definieren.

Auf Basis der Theorie der Energiekaskade $\mathcal{T} = \varepsilon$ kann eine Längen-, Geschwindigkeits- und Zeitskalierung dieser kleinskaligen Strukturen mit ε und ν formuliert werden:

$$\eta = (\nu^3/\varepsilon)^{1/4} \tag{2.3.2}$$

$$u_\eta = (\nu\varepsilon)^{1/4} \tag{2.3.3}$$

$$\tau_\eta = (\nu/\varepsilon)^{1/2} \tag{2.3.4}$$

Dass diese *Kolmogorov Skalen* die kleinsten dissipativen Wirbel charakterisieren, zeigt sich zum einen durch die damit gebildete Reynoldszahl:

$$Re = u_\eta \eta / \nu = 1 \tag{2.3.5}$$

Zum anderen kann die Dissipationsrate durch 2.3.4 mit der Zeitskala $\tau_\eta = \eta/u_\eta$ wie folgt angeschrieben werden:

¹ Nachdruck der Royal Society of London

2.3 Direkte numerische Simulation

$$\varepsilon = \nu / \tau_\eta^2 = \nu (u_\eta / \eta)^2 \quad 2.3.6$$

Obiger Ausdruck kann auch durch Division von 2.3.3 mit 2.3.4 und anschließender Umformung erhalten werden. Der Vergleich von Gleichung 2.3.6 mit der Definition der Dissipationsrate

$$\varepsilon \equiv \nu / 2 \left(\partial u_i / \partial x_j + \partial u_j / \partial x_i \right) \left(\partial u_i / \partial x_j + \partial u_j / \partial x_i \right) \quad 2.3.7$$

zeigt, dass die Kolmogorov Skalen eine konsistente Abschätzung $1/\tau_\eta = u_\eta/\eta$ für die Geschwindigkeitsgradienten in den kleinsten dissipativen Wirbeln bieten und daher eben diese skalieren.

Wird die Dissipationsrate mit der aus der Theorie der Energiekaskade gefolgerten Abschätzung $\varepsilon = u_0^3/l_0$ skaliert, können die Verhältnisse der kleinsten zu den größten Skalen als Funktionen der makroskopischen Reynoldszahl $Re_0 = u_0(l_0)/\nu$ dargestellt werden.

$$\eta/l_0 \sim Re_0^{-3/4} \quad 2.3.8$$

$$u_\eta/u_0 \sim Re_0^{-1/4} \quad 2.3.9$$

$$\tau_\eta/\tau_0 \sim Re_0^{-1/2} \quad 2.3.10$$

Mit steigender Turbulenz werden die Abmessungen der dissipativen Wirbel kleiner. Auch das Verhältnis der Zeitskalierungen sinkt. Das ermöglicht eine schnelle Anpassung der kleinen Wirbel an die großen, trägen Wirbel. Damit wird die Annahme von kleinskaligen Strömungsstrukturen, die von der makroskopischen Strömung unabhängig sind, bestärkt. Kolmogorov ging insofern davon aus, dass die kleinskaligen Strukturen in unterschiedlichen Strömungen mit hohen Re_0 zueinander ähnlich sind. Mit der Kolmogorov Skalierung kann insofern ein universelles, dimensionsloses Geschwindigkeitsfeld für die kleinsten dissipativen Wirbel angeschrieben werden.

$$\mathbf{w}(\mathbf{y}) \equiv [\mathbf{U}(\mathbf{x}, t_0) - \mathbf{U}(\mathbf{x}_0, t_0)]/u_\eta \text{ mit } \mathbf{y} \equiv (\mathbf{x} - \mathbf{x}_0)/\eta \quad 2.3.11$$

2.3.3 Berechnungsaufwand DNS

Durch die in Kapitel 2.3.2 angeschriebene Abschätzung der Kolmogorov Skalen mithilfe von Re_0 ist eine grobe Einschätzung des Berechnungsaufwands einer direkten numerischen Simulation möglich. Die Anzahl der notwendigen Gitterpunkte in einer Dimension kann dann durch 2.3.8 ermittelt werden:

2.4 Gemittelte Navier-Stokes Gleichungen

$$N \sim \frac{l_0}{\Delta x} \sim \frac{l_0}{\eta} \sim Re_0^{3/4} \quad 2.3.12$$

Die zu diskretisierende Rechenzeit entspricht üblicherweise der Zeitskala der größten Wirbel $\tau_0 = l_0/u_0$. Die Zeitschrittweite ist aufgrund des Courant-Zahl-Kriteriums nicht frei wählbar:

$$C = \frac{u_0 \Delta t}{\Delta x} \leq 1 \quad 2.3.13$$

Die Anzahl der erforderlichen Zeitschritte lässt sich daher in folgender Weise abschätzen:

$$M \sim \frac{\tau_0}{\Delta t} \sim \frac{l_0}{u_0} \frac{u_0}{\Delta x} \sim \frac{l_0}{\Delta x} \sim N \quad 2.3.14$$

Der gesamte Rechenaufwand entspräche aufgrund der drei diskretisierten Raumrichtungen samt der zeitlichen Auflösung:

$$N^4 \sim Re_0^3 \quad 2.3.15$$

Er steigt also mit zunehmender Turbulenz kubisch an, denn die Kolmogorov Skalen werden mit höheren Re_0 sukzessive kleiner. Stark turbulente Strömungen können daher nicht durch eine DNS berechnet werden. Der Anwendungsbereich der DNS beschränkt sich insofern auf fundamentale, einfache Strömungen mit moderaten Re_0 .

2.4 Gemittelte Navier-Stokes Gleichungen

Um die Simulation von Strömungen mit hohen Re_0 zu ermöglichen, müssen die universellen, turbulenten Strukturen modelliert werden. Reynolds (1895) schlug daher vor, das turbulente Geschwindigkeitsfeld $\mathbf{U}(\mathbf{x}, t)$ in einen gemittelten Anteil $\langle \mathbf{U}(\mathbf{x}, t) \rangle$, zur Beschreibung makroskopischer Strömungsstrukturen, und einen fluktuierenden Anteil $\mathbf{u}(\mathbf{x}, t)$, zur Berücksichtigung der Turbulenz, zu zerlegen (*Reynolds Decomposition*).

Die Reynolds Decomposition lässt sich wie folgt mathematisch formulieren und wird in Abbildung 2.3 illustriert:

$$\mathbf{U}(\mathbf{x}, t) \equiv \langle \mathbf{U}(\mathbf{x}, t) \rangle + \mathbf{u}(\mathbf{x}, t) \quad 2.4.1$$

2.4 Gemittelte Navier-Stokes Gleichungen

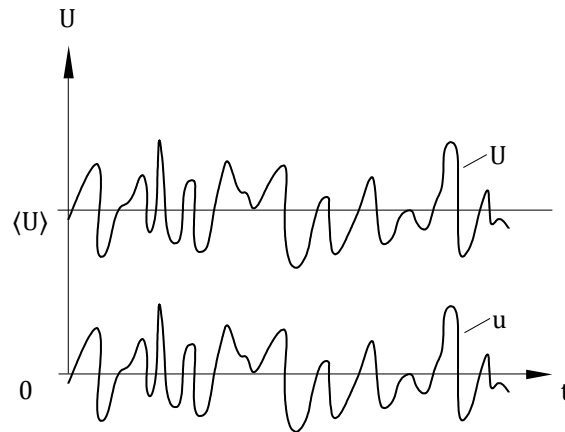


Abbildung 2.3 Schematische Darstellung der Reynolds Decomposition (Paschedag, 2004)

Der reale Geschwindigkeitsverlauf $\mathbf{U}(\mathbf{x}, t)$ weicht vom konstanten Mittelwert $\langle \mathbf{U}(\mathbf{x}, t) \rangle$ mit der Geschwindigkeitsfluktuation $\mathbf{u}(\mathbf{x}, t)$ ab. Das gemittelte Strömungsfeld $\langle \mathbf{U}(\mathbf{x}, t) \rangle$ kann insofern als laminare Unterströmung angesehen werden, deren Berechnung über die numerische Lösung der Navier-Stokes Gleichungen mit verhältnismäßigem Aufwand durchführbar ist. Die Berechnung ist insofern sinnvoll, da der Verlauf von $\langle \mathbf{U}(\mathbf{x}, t) \rangle$, sowie die Größenordnung der charakteristischen turbulenten Größen, für viele technische Anwendungen ausreichend sind.

Um das gemittelte Strömungsfeld $\langle \mathbf{U}(\mathbf{x}, t) \rangle$ zu erhalten, müssen jedoch die zugrundeliegenden Navier-Stokes Gleichungen ebenfalls gemittelt werden - *Reynolds Averaged Navier-Stokes* (RANS) (Reynolds, 1895). Dabei treten in den gemittelten Navier-Stokes-Gleichungen unbekannte Terme auf, welche die Geschwindigkeitsfluktuation $\mathbf{u}(\mathbf{x}, t)$ enthalten. Diese Terme müssen zur Reduktion des Rechenaufwandes durch RANS-Turbulenzmodelle, wie etwa das k- ϵ -Modell, modelliert werden. Im Folgenden werden daher die gemittelten Navier-Stokes- Gleichungen hergeleitet:

Für stationäre Strömungen ist die zeitliche Mittelung einer beliebigen Bilanzgröße $f(\mathbf{x}, t)$ geeignet und nach 2.2.4 definiert:

$$\langle f(\mathbf{x}, t) \rangle = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T f(\mathbf{x}, t) dt = g(\mathbf{x}) \quad 2.4.2$$

Gleichung 2.4.1 in 2.4.2 eingesetzt ergibt:

$$\langle \mathbf{U}(\mathbf{x}, t) \rangle = \langle \langle \mathbf{U}(\mathbf{x}, t) \rangle \rangle + \langle \mathbf{u}(\mathbf{x}, t) \rangle = \langle \mathbf{U}(\mathbf{x}, t) \rangle + 0 \quad 2.4.3$$

Der gemittelte fluktuierende Geschwindigkeitsanteil ist daher Null. Dieser Zusammenhang gilt auch für ein analog aufgeteiltes Skalarfeld:

$$\phi(\mathbf{x}, t) \equiv \langle \phi(\mathbf{x}, t) \rangle + \phi'(\mathbf{x}, t) \quad 2.4.4$$

2.4 Gemittelte Navier-Stokes Gleichungen

wie zum Beispiel das Druckfeld.

$$p(\mathbf{x}, t) \equiv \langle p(\mathbf{x}, t) \rangle + p'(\mathbf{x}, t) \quad 2.4.5$$

2.4.1 Gemittelte Massenerhaltung

Die Kontinuitätsgleichung für inkompressible Fluide kann unter Berücksichtigung von 2.4.1 wie folgt angeschrieben werden:

$$\nabla \mathbf{U} = \nabla(\langle \mathbf{U} \rangle + \mathbf{u}) = 0 \quad 2.4.6$$

Durch die Linearität der Kontinuitätsgleichung ist die Mittelung trivial. Durch die Kommutativität der Mittelung mit der Ableitung (∇) kann Gleichung 2.4.6 formuliert werden:

$$\langle \nabla \mathbf{U} \rangle = \langle \nabla(\langle \mathbf{U} \rangle + \mathbf{u}) \rangle = \nabla(\langle \mathbf{U} \rangle + \langle \mathbf{u} \rangle) = \nabla \langle \mathbf{U} \rangle = 0 \quad 2.4.7$$

Daraus geht unter Berücksichtigung von Gleichung 2.4.6 hervor, dass neben dem mittleren Geschwindigkeitsfeld $\langle \mathbf{U}(\mathbf{x}, t) \rangle$ auch das fluktuierende Geschwindigkeitsfeld $\mathbf{u}(\mathbf{x}, t)$ divergenzfrei ist: $\nabla \cdot \mathbf{u}(\mathbf{x}, t) = 0$

2.4.2 Gemittelte Impulserhaltung

Für inkompressible, Newtonsche Fluide geht die Impulserhaltung in die Navier-Stokes-Gleichung (2.2.4) über. Die Mittelung der rechten Seite ist aufgrund der Linearität der enthaltenen Terme unproblematisch:

$$\left\langle \frac{D U_j}{D t} \right\rangle = \nu \nabla^2 \langle U_j \rangle - \frac{1}{\rho} \frac{\partial \langle p \rangle}{\partial x_j} \quad 2.4.8$$

Die substantielle Ableitung auf der linken Seite, enthält mit dem konvektiven Term einen quadratischen Ausdruck.

$$\left\langle \frac{D U_j}{D t} \right\rangle = \left\langle \frac{\partial U_j}{\partial t} + \frac{\partial}{\partial x_i} (U_i U_j) \right\rangle = \frac{\partial \langle U_j \rangle}{\partial t} + \frac{\partial}{\partial x_i} (\langle U_i U_j \rangle) \quad 2.4.9$$

Der nichtlineare Term lässt sich folgendermaßen aufschlüsseln:

$$\langle U_i U_j \rangle = \langle (\langle U_i \rangle + u_i)(\langle U_j \rangle + u_j) \rangle = \langle U_i \rangle \langle U_j \rangle + u_i \langle U_j \rangle + u_j \langle U_i \rangle + u_i u_j = \langle U_i \rangle \langle U_j \rangle + \langle u_i u_j \rangle \quad 2.4.10$$

Der letzte Ausdruck auf der rechten Seite, die Kovarianz $\langle u_i u_j \rangle$ wird *Reynoldsspannungen* oder *Reynoldsscher Spannungstensor* genannt und ist im Allgemeinen ungleich Null. Wird 2.4.10 herangezogen, kann 2.4.9 wie nachstehend angeschrieben werden.

$$\left\langle \frac{D U_j}{D t} \right\rangle = \frac{\partial \langle U_j \rangle}{\partial t} + \frac{\partial}{\partial x_i} (\langle U_i \rangle \langle U_j \rangle + \langle u_i u_j \rangle) = \frac{\partial \langle U_j \rangle}{\partial t} + \langle U_i \rangle \frac{\partial \langle U_j \rangle}{\partial x_i} + \frac{\partial \langle u_i u_j \rangle}{\partial x_i} = \frac{\bar{D} \langle U_j \rangle}{\bar{D} t} + \frac{\partial \langle u_i u_j \rangle}{\partial x_i} \quad 2.4.11$$

2.5 k-ε-Modell

Wobei die mittlere substantielle Ableitung wie folgt definiert ist:

$$\frac{\bar{D}}{\bar{D}t} \equiv \frac{\partial}{\partial t} + \langle U_i \rangle \frac{\partial}{\partial x_i} \quad 2.4.12$$

Die gemittelte Impulserhaltung, auch *Reynoldsgleichung* genannt, entspricht mit 2.4.8 und 2.4.11:

$$\frac{\bar{D}\langle U_j \rangle}{\bar{D}t} = \nu \nabla^2 \langle U_j \rangle - \frac{\partial \langle u_i u_j \rangle}{\partial x_i} - \frac{1}{\rho} \frac{\partial \langle p \rangle}{\partial x_j} \quad 2.4.13$$

Sie unterscheidet sich von der Form der Navier-Stokes-Gleichung nur durch den Reynoldsschen Spannungstensor. Dieser enthält aufgrund seiner Symmetrie ($\langle u_i u_j \rangle = \langle u_j u_i \rangle$) sechs Unbekannte. Mit den vier Größen $\langle \mathbf{U}(\mathbf{x}, t) \rangle$ und $\langle p(\mathbf{x}, t) \rangle$ liegen insgesamt zehn Unbekannte vor. Diesen stehen mit der Impuls- und Massenerhaltung nur vier Gleichungen gegenüber. Die zehn Unbekannten können daher nicht eindeutig bestimmt werden. Es liegt ein *Schließungsproblem* vor.

2.4.3 Gemittelte Erhaltungsgleichung eines Skalars

Hat ein Skalar keinen Einfluss auf Materialeigenschaften und als Erhaltungsgröße keine Quellen und Senken, so kann die Erhaltungsgleichung wie folgt angeschrieben werden:

$$\frac{D\phi}{Dt} = \frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{U}\phi) = \Gamma \nabla^2 \phi \quad 2.4.14$$

Einsetzen der Reynolds Decomposition 2.4.4 in die Erhaltungsgleichung und Auftrennung des nicht linearen konvektiven Terms nach 2.4.10 liefert nach Mittelung, die Erhaltungsgleichung für das gemittelte Skalar $\langle \phi \rangle$:

$$\rho \frac{\bar{D}\langle \phi \rangle}{\bar{D}t} = \nabla \cdot (\Gamma \nabla \langle \phi \rangle - \langle \mathbf{u}\phi' \rangle) \quad 2.4.15$$

2.5 k-ε-Modell

RANS-Turbulenzmodelle versuchen den Reynoldsschen Spannungstensor, bzw. dessen Transport, zu modellieren und somit das offene Gleichungssystem zu schließen. Eines der bekanntesten dieser Modelle ist das *k-ε-Modell*, dessen Aufbau in den folgenden Kapiteln erklärt werden soll.

2.5.1 Zugrundeliegende Theorien

Das k-ε-Modell ist historisch gewachsen und fußt auf mehreren älteren Theorien und Konzepten. Diese sind für das Verständnis des k-ε-Modells essentiell und sollen deshalb in Kapitel 2.5.1 erläutert werden.

2.5.1.1 Hypothese der Gradienten-Diffusion

Das einfachste Modell zur Beschreibung des Transportes einer turbulenten Größe ist die *Gradienten-Diffusion*. Dabei wird der turbulente Transport einer skalaren Erhaltungsgröße $\langle \mathbf{u}\phi \rangle$ durch Diffusion entgegen dem mittleren Gradienten $\nabla\langle\phi\rangle$ modelliert.

$$\langle \mathbf{u}\phi \rangle = -\Gamma_T \nabla\langle\phi\rangle \quad 2.5.1$$

Die Erhaltungsgleichung 2.4.15 für das gemittelte Skalar $\langle\phi\rangle$ kann also zu folgendem Ausdruck umgeformt werden:

$$\rho \frac{\overline{D}\langle\phi\rangle}{Dt} = \nabla \cdot (\Gamma \nabla\langle\phi\rangle + \Gamma_T(\mathbf{x}, t) \nabla\langle\phi\rangle) = \nabla \cdot (\Gamma + \Gamma_T(\mathbf{x}, t)) \nabla\langle\phi\rangle \quad 2.5.2$$

Die effektive Diffusivität ist dabei als Summe der molekularen und der turbulenten Diffusivität definiert:

$$\Gamma_{\text{eff}}(\mathbf{x}, t) = \Gamma + \Gamma_T(\mathbf{x}, t) \quad 2.5.3$$

Die Gradienten-Diffusion wird im Weiteren benötigt werden, um den turbulenten Transport der turbulenten kinetischen Energie k und der Dissipationsrate ε zu modellieren.

2.5.1.2 Hypothese der turbulenten Viskosität

Die Gradienten-Diffusion modelliert den turbulenten Transport analog zur molekularen Diffusion. Ebenso können die turbulenten Spannungen analog zu den molekularen, viskosen Spannungen definiert werden. Diese Hypothese der turbulenten Viskosität wurde von Boussinesq entwickelt (1897).

Die Reynoldsgleichung kann hierfür wie folgt angeschrieben werden:

$$\rho \frac{\overline{D}\langle u_j \rangle}{Dt} = \frac{\partial}{\partial x_i} \left[\mu \left(\frac{\partial \langle u_i \rangle}{\partial x_j} + \frac{\partial \langle u_j \rangle}{\partial x_i} \right) - \langle p \rangle \delta_{ij} - \rho \langle u_i u_j \rangle \right] \quad 2.5.4$$

Aus der Spur des Reynoldsschen Spannungstensors kann die *turbulente kinetische Energie* k bestimmt werden.

$$k \equiv \frac{1}{2} \langle u_i u_i \rangle \quad 2.5.5$$

Des Weiteren lässt sich der Reynoldssche Spannungstensor in einen isotropen Teil $-2/3 k \delta_{ij}$ und einen anisotropen Teil a_{ij} zerlegen. Der isotrope Teil enthält die turbulenten Normalspannungen, der anisotrope Teil umfasst wiederum die turbulenten Schubspannungen.

$$\langle u_i u_i \rangle \equiv a_{ij} + \frac{2}{3} k \delta_{ij} \quad 2.5.6$$

Wird in die rechte Seite von 2.5.4 Gleichung 2.5.6 eingesetzt, kann 2.5.7 angeschrieben werden:

$$RHS = \frac{\partial}{\partial x_i} \left[\mu \left(\frac{\partial \langle u_i \rangle}{\partial x_j} + \frac{\partial \langle u_j \rangle}{\partial x_i} \right) - \rho a_{ij} \right] - \frac{\partial}{\partial x_j} \left[\langle p \rangle + \frac{2}{3} \rho k \right] \quad 2.5.7$$

Der Transport turbulenter Spannungen wird also durch den anisotropen Teil gewährleistet. Der isotrope Teil kann durch einen modifizierten mittleren Druck $\langle p \rangle + 2/3 \rho k$ aufgenommen werden. Er wird daher in vielen Modellierungen vernachlässigt.

Nach der *Hypothese der turbulenten Viskosität* sind die turbulenten, anisotropen Spannungen analog zu den viskosen anisotropen Spannungen definiert.³

$$-\rho a_{ij} = -\rho \langle u_i u_j \rangle + \frac{2}{3} \rho k \delta_{ij} \equiv \rho \nu_T \left(\frac{\partial \langle u_i \rangle}{\partial x_j} + \frac{\partial \langle u_j \rangle}{\partial x_i} \right) = 2 \rho \nu_T \bar{S}_{ij} \quad 2.5.8$$

Die *turbulente Viskosität* $\nu_T(\mathbf{x}, t)$ ist ein skalarer Proportionalitätsfaktor und im Gegensatz zur Viskosität ν keine Stoffeigenschaft, sondern, als Eigenschaft der Strömung, eine Funktion von Raum \mathbf{x} und Zeit t . Obige Definition in die Reynoldsgleichung eingesetzt ergibt:

$$\frac{\overline{D} \langle u_j \rangle}{Dt} = \frac{\partial}{\partial x_i} \left[\nu_{eff} \left(\frac{\partial \langle u_i \rangle}{\partial x_j} + \frac{\partial \langle u_j \rangle}{\partial x_i} \right) \right] - \frac{1}{\rho} \frac{\partial}{\partial x_j} \left[\langle p \rangle + \frac{2}{3} \rho k \right] \quad 2.5.9$$

Wobei die effektive Viskosität als Summe zwischen der Viskosität und der turbulenten Viskosität definiert ist:

$$\nu_{eff}(\mathbf{x}, t) \equiv \nu + \nu_T(\mathbf{x}, t) \quad 2.5.10$$

2.5.1.3 Prandtlscher Mischungsweg

Durch den oben beschriebenen Modellansatz, wird die turbulente Viskosität als Unbekannte eingeführt. Ihre Beschreibung bzw. Bestimmung beruht auf einem Turbulenzmodell von Prandtl (1931).

² Für δ_{ij} existiert folgende alternative Schreibweise: $\partial x_j / \partial x_i = 1_{j=i}, 0_{j \neq i}$. Durch diese ist nachstehende Formulierung möglich: $\partial \mathbf{x} / \partial x_j = \partial (e_i x_i) / \partial x_j = e_i \partial x_j / \partial x_i = e_i \delta_{ij} = e_j$. Die Indexnotation $e_i \delta_{ij} = e_j$ einhaltend, kann der Term $\partial x_j / \partial x_i \delta_{ij} = \partial / \partial x_j$ angeschrieben werden.

³ Bei inkompressiblen Strömungen besitzen sowohl \bar{S}_{ij} also auch a_{ij} keine Spur.

2.5 k-ε-Modell

Prandtls Überlegung geht von einer zweidimensionalen turbulenten Scherströmung aus ($\langle \mathbf{U} \rangle = (\langle U \rangle, 0, 0)^T$ & $\partial \langle U \rangle / \partial x_i = (0, \partial \langle U \rangle / \partial y > 0, 0)^T$). Diese ist zur Veranschaulichung schematisch in Abbildung 2.4 dargestellt.

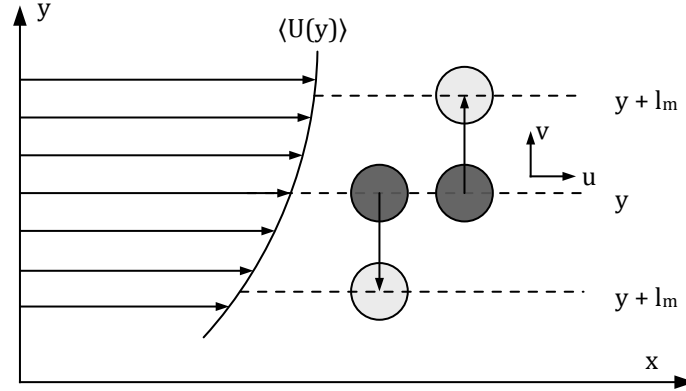


Abbildung 2.4 Schematische Darstellung des Prandtlschen Mischungsweges

Dies entspricht zum Beispiel einer Strömung in Wandnähe. In ihr werden in x -Richtung Turbulenzballen transportiert. Diese turbulenten Wirbel können als substantielle Fluidelemente betrachtet werden. Sie legen im Mittel die Länge l_m in y -Richtung zurück, bevor sie sich mit dem restlichen Fluid vermischen und ihre Identität verlieren. Die Länge l_m wird daher *Mischungsweg* genannt. Durch die Bewegung in y -Richtung ändert sich die Geschwindigkeit in x -Richtung mit der Rate $\partial \langle U \rangle / \partial y$. Diese Geschwindigkeitsänderung wird als x -Komponente der Geschwindigkeitsfluktuation u interpretiert und ist näherungsweise:

$$u \sim l_m \frac{\partial \langle U \rangle}{\partial y} \quad 2.5.11$$

Aufgrund der Kontinuität muss die Änderung der Geschwindigkeit in y -Richtung gegengleich groß sein $u = -v$. Somit erhält man:

$$-\rho \langle u_i u_j \rangle = -\rho \langle uv \rangle = \rho l_m^2 \left| \frac{\partial \langle U \rangle}{\partial y} \right| \frac{\partial \langle U \rangle}{\partial y} \quad 2.5.12$$

Durch Vernachlässigung der isotropen, turbulenten Schubspannungen in 2.5.8, kann die verbleibende Gleichung mit 2.5.12 gleichgesetzt und verglichen werden. Der mittlere Scherraten Tensor $\bar{\mathbf{S}}$ reduziert sich hierbei für die gegebene, einfache Scherströmung zu $\partial \langle U \rangle / \partial y$. Die turbulente Viskosität kann also wie folgt abgeschätzt werden:

$$-\rho \langle uv \rangle = \rho \nu_T \frac{\partial \langle U \rangle}{\partial y} = \rho l_m^2 \left| \frac{\partial \langle U \rangle}{\partial y} \right| \frac{\partial \langle U \rangle}{\partial y} \rightarrow \nu_T = l_m^2 \left| \frac{\partial \langle U \rangle}{\partial y} \right| \approx l^* \cdot u^* \quad 2.5.13$$

2.5 k-ε-Modell

Als Eigenschaft der Strömung und unter Berücksichtigung von Gleichung 2.5.11 ist die in Gleichung 2.5.13 angegebene Skalierung der turbulenten Viskosität $\nu_T \approx l^* \cdot u^*$ naheliegend. Die turbulente Viskosität setzt sich also multiplikativ aus einer Geschwindigkeitsskalierung und einer Längenskalierung zusammen.

Geschwindigkeitsskalierung

Das Produkt aus Scherrate und Mischungslänge (Gleichung 2.5.13) ist als Geschwindigkeitsskalierung ungeeignet. Denn die Scherrate kann an Orten mit hoher Turbulenz, wie beispielsweise an der Mittelachse eines turbulenten Jets, gegen Null gehen. Sowohl Prandtl (1945) als auch Kolmogorov (1942) schlugen deshalb vor, die Referenzgeschwindigkeit u^* durch die turbulente kinetische Energie k abzubilden.

$$u^* = ck^{1/2} \quad 2.5.14$$

Hierbei ist c eine Konstante.

Längenskalierung

Für die Längenskalierung l^* wird die Mischungslänge l_m herangezogen. Im k-ε-Modell wird diese über die Dissipationsrate beschrieben. Die dabei verwendete Skalierung basiert auf dem Konzept der Energiekaskade. Durch diese kann ε abgeschätzt werden als (siehe Kapitel 2.3.1):

$$\varepsilon = \frac{u_0^2}{\tau_0} = \frac{u_0^3}{l_0} \quad 2.5.15$$

Die dabei auftretenden Unbekannten u_0 und l_0 können durch k bzw. l_m modelliert werden.

$$\varepsilon = C_D \frac{k^{3/2}}{l_m} \quad 2.5.16$$

C_D ist hier wiederum eine Modellkonstante. Umformung ergibt für die Mischungslänge:

$$l_m = C_D \frac{k^{3/2}}{\varepsilon} \quad 2.5.17$$

2.5.2 Transportgleichungen für k und ε

Die turbulente Viskosität kann mit k und ε durch das Einsetzen der Gleichungen 2.5.14 und 2.5.17 in Gleichung 2.5.13 skaliert werden:

$$\nu_T = C_\mu \frac{k^2}{\varepsilon} \quad 2.5.18$$

2.5 k-ε-Modell

C_μ ist hierbei die erste der fünf k-ε-Modellkonstanten. In einfachen turbulenten Scherströmungen $((\langle U \rangle, 0, 0)^T, \partial \langle U \rangle / \partial x_i = (0, \partial \langle U \rangle / \partial y, 0)^T)$ reduziert sich die turbulente Viskosität auf $\nu_T = -\langle uv \rangle / (\partial \langle U \rangle / \partial y)$. In diesen simplen Strömungen ist eine Berechnung von C_μ mittels DNS möglich. Abbildung 2.5 zeigt den durch DNS-Ergebnisse erhaltenen Verlauf von C_μ über dem dimensionslosen Wandabstand y^+ .

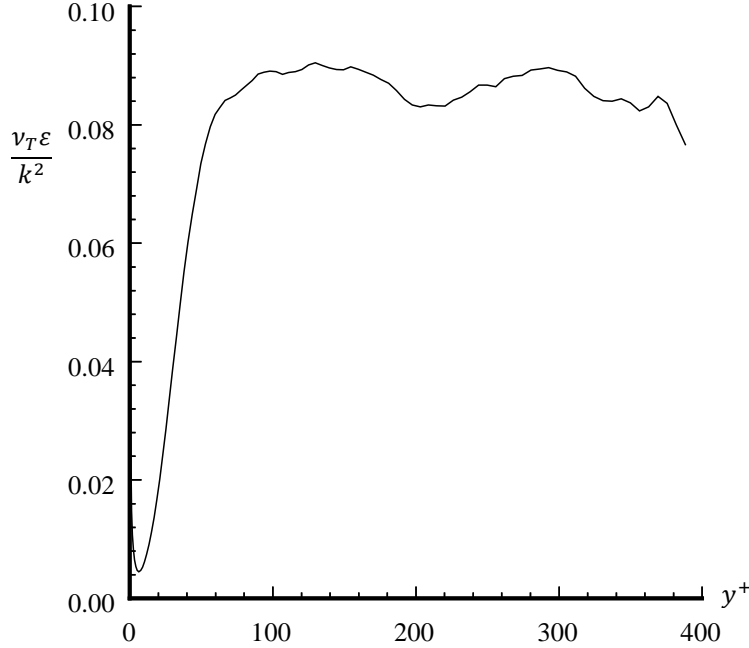


Abbildung 2.5 Verlauf von $\nu_T \varepsilon / k^2 = C_\mu$ in Abhängigkeit zum dimensionslosen Wandabstand y^+ , DNS-Ergebnisse einer Kanalströmung mit $Re = 13750$ (Kim, Moin, & Moser, 1987).

Der Wert für C_μ liegt mit hinreichendem Abstand zur Wand ($y^+ > 50$) bei etwa 0.09. Ist C_μ bekannt, bedarf es, zur Bestimmung der turbulenten Viskosität, Transportgleichungen für die Feldgrößen k und ε .

2.5.2.1 Transportgleichung für k

Das Mittel der gesamten kinetischen Energie $\langle E(\mathbf{x}, t) \rangle$ kann, im Sinne einer Reynolds Decomposition, in die kinetische Energie der gemittelten, makroskopischen Strömung $\bar{E}(\mathbf{x}, t)$ und in die kinetische Energie $k(\mathbf{x}, t)$ der überlagerten turbulenten Strömung zerlegt werden.

$$\langle E(\mathbf{x}, t) \rangle \equiv \left\langle \frac{1}{2} \mathbf{U}(\mathbf{x}, t) \cdot \mathbf{U}(\mathbf{x}, t) \right\rangle = \frac{1}{2} \langle \mathbf{U} \rangle \cdot \langle \mathbf{U} \rangle + \frac{1}{2} \langle \mathbf{u} \cdot \mathbf{u} \rangle \equiv \bar{E}(\mathbf{x}, t) + k(\mathbf{x}, t) \quad 2.5.19$$

Die Transportgleichung für das Mittel der kinetischen Energie $\langle E(\mathbf{x}, t) \rangle$ wird bei inkompressiblen, Newtonschen Fluiden, durch Multiplikation der Navier-Stokes-Gleichung mit der Geschwindigkeit U_j und anschließender Mittelung hergeleitet. Analog

2.5 k-ε-Modell

kann die Transportgleichung der kinetischen Energie der gemittelten Strömung $\bar{E}(\mathbf{x}, t)$, durch das Produkt der Reynoldsgleichung (gemittelte Navier-Stokes-Gleichung) mit der mittleren Geschwindigkeit $\langle U_j \rangle$, entwickelt werden. Subtraktion der Reynoldsgleichung von der Navier-Stokes-Gleichung, anschließende Multiplikation der Differenz mit u_j sowie darauffolgende Mittelung des Produkts ermöglicht, wie im Folgenden bewiesen, die Herleitung der Transportgleichung für die turbulente kinetische Energie:

Die Navier-Stokes-Gleichung kann aufgrund von $\partial U_i / \partial x_i = 0$ alternativ angeschrieben werden:

$$\frac{DU_j}{Dt} = \frac{\partial U_j}{\partial t} + \frac{\partial(U_i U_j)}{\partial x_i} = \nu \nabla^2 U_j - \frac{1}{\rho} \frac{\partial p}{\partial x_j} \quad 2.5.20$$

Die Reynoldsgleichung wird unter Verwendung von Gleichung 2.4.11 auf folgende Form gebracht:

$$\left\langle \frac{DU_j}{Dt} \right\rangle = \frac{\partial \langle U_j \rangle}{\partial t} + \frac{\partial \langle U_i U_j \rangle}{\partial x_i} = \nu \nabla^2 \langle U_j \rangle - \frac{1}{\rho} \frac{\partial \langle p \rangle}{\partial x_j} \quad 2.5.21$$

Durch Subtraktion erhält man:

$$\frac{\partial u_j}{\partial t} + \frac{\partial}{\partial x_i} (U_i U_j - \langle U_i U_j \rangle) = \nu \nabla^2 \langle u_j \rangle - \frac{1}{\rho} \frac{\partial p'}{\partial x_j} \quad 2.5.22$$

Der zweite Term von links lässt sich unter Verwendung von Gleichung 2.4.10 folgendermaßen aufgliedern:

$$\frac{\partial}{\partial x_i} (U_i U_j - \langle U_i U_j \rangle) = \frac{\partial}{\partial x_i} (u_i \langle U_j \rangle + u_j \langle U_i \rangle + u_i u_j - \langle u_i u_j \rangle) = u_i \frac{\partial \langle U_j \rangle}{\partial x_i} - \frac{\partial \langle u_i u_j \rangle}{\partial x_i} + \langle U_i \rangle \frac{\partial u_j}{\partial x_i} + u_i \frac{\partial u_j}{\partial x_i} \quad 2.5.23$$

Multiplikation der Gleichung 2.5.22 mit u_j und nachfolgendes Einsetzen von Gleichung 2.5.23 ergibt:

$$u_j \frac{\partial u_j}{\partial t} + (\langle U_i \rangle + u_i) u_j \frac{\partial u_j}{\partial x_i} = -u_i u_j \frac{\partial \langle U_j \rangle}{\partial x_i} + u_j \frac{\partial \langle u_i u_j \rangle}{\partial x_i} + \nu u_j \nabla^2 \langle u_j \rangle - u_j \frac{1}{\rho} \frac{\partial p'}{\partial x_j} \quad 2.5.24$$

Gleichung 2.5.24 kann unter Verwendung der Produktregel, sowie der Bedingung $\partial U_i / \partial x_i = 0$ auf Gleichung 2.5.25 umgeformt werden:

$$\frac{\partial (u_j u_j / 2)}{\partial t} + \langle U_i \rangle \frac{\partial (u_j u_j / 2)}{\partial x_i} + \frac{\partial (u_i u_j u_j / 2)}{\partial x_i} = -u_i u_j \frac{\partial \langle U_j \rangle}{\partial x_i} + u_j \frac{\partial \langle u_i u_j \rangle}{\partial x_i} + \nu u_j \nabla^2 \langle u_j \rangle - \frac{1}{\rho} \frac{\partial (p' u_j)}{\partial x_j} \quad 2.5.25$$

Durch Mittelung der obigen Gleichung und unter Berücksichtigung von $\langle u_i \rangle = 0$ und $k = 1/2 \langle u_i u_i \rangle$, entspricht die Transportgleichung für k nachstehendem Ausdruck:

$$\frac{\partial k}{\partial t} + \langle U_i \rangle \frac{\partial k}{\partial x_i} + \frac{\partial \langle u_i u_j u_j / 2 \rangle}{\partial x_i} = -\langle u_i u_j \rangle \frac{\partial \langle U_j \rangle}{\partial x_i} + \nu \langle u_j \nabla^2 u_j \rangle - \frac{1}{\rho} \frac{\partial \langle p' u_j \rangle}{\partial x_j} \quad 2.5.26$$

Quelle - Produktion

Der erste Term auf der rechten Seite von 2.5.26 erscheint auch in der Transportgleichung der kinetischen Energie der gemittelten, makroskopischen Strömung. Dort aber mit anderem Vorzeichen (siehe Gleichung 2.5.40). Er bewerkstelligt also den Austausch kinetischer Energie zwischen der mittleren und der fluktuierenden Strömung. Er wird daher Produktion \mathcal{P} genannt.

$$\mathcal{P} = -\langle u_i u_j \rangle \frac{\partial \langle u_j \rangle}{\partial x_i} \approx 2\nu_T \bar{S}_{ij} \frac{\partial \langle u_j \rangle}{\partial x_i} \quad 2.5.27$$

Es ist evident, dass die Produktion \mathcal{P} nicht in der Transportgleichung für die gesamte kinetische Energie enthalten ist (Gleichung 2.5.39).

Zur Lösung des Schließungs-Problems, muss der Produktionsterm modelliert werden, denn er enthält die Reynoldsschen Spannungen. Die Hypothese der turbulenten Viskosität ist hierfür naheliegend.

Senke - Dissipation

Der zweite Term auf der rechten Seite von 2.5.26 enthält die Dissipationsrate ε , also jene Rate ($\sim s_{ij}$), mit der die turbulente kinetische Energie k aufgrund von Reibung νs_{ij} dissipiert.

$$\varepsilon \equiv 2\nu \langle s_{ij} s_{ij} \rangle \quad 2.5.28$$

Die Reibung νs_{ij} wird dabei von Scherraten s_{ij} verursacht.

$$s_{ij} = 1/2 (\partial u_i / \partial x_j + \partial u_j / \partial x_i) \quad 2.5.29$$

Die Scherraten werden durch die Geschwindigkeitsfluktuationen u_i , insofern der Turbulenz, erzeugt.

Die Dissipationsrate kann daher durch Umformung, in der Transportgleichung von k , explizit ausgedrückt werden:

$$\nu \langle u_j \nabla^2 u_j \rangle = \nu \langle u_j \frac{\partial}{\partial x_i} \left(\frac{\partial u_j}{\partial x_i} \right) \rangle = \nu \langle u_j \frac{\partial}{\partial x_i} \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right) \rangle = 2\nu \langle u_j \frac{\partial s_{ij}}{\partial x_i} \rangle = 2\nu \frac{\partial \langle u_j s_{ij} \rangle}{\partial x_i} - 2\nu \langle s_{ij} \frac{\partial u_j}{\partial x_i} \rangle \quad 2.5.30$$

Der letzte Term aus 2.5.30 entspricht dabei ε :

$$2\nu \langle s_{ij} \frac{\partial u_j}{\partial x_i} \rangle = 2\nu \langle s_{ij} \left[\frac{1}{2} \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right) + \frac{1}{2} \left(\frac{\partial u_j}{\partial x_i} - \frac{\partial u_i}{\partial x_j} \right) \right] \rangle = 2\nu \langle s_{ij} (s_{ij} + r_{ij}) \rangle = 2\nu \langle s_{ij} s_{ij} + s_{ij} r_{ij} \rangle = \varepsilon^4 \quad 2.5.31$$

⁴ Der Tensor s_{ij} ist symmetrisch $s_{ij} = s_{ji}$, r_{ij} ist antisymmetrisch $r_{ij} = -r_{ji}$. Insofern gilt: $s_{ij} r_{ij} = 0$

2.5 k-ε-Modell

Einsetzen der zwei obigen Umformungen in Gleichung 2.5.26 ergibt 2.5.32:

$$\frac{\partial k}{\partial t} + \langle U_i \rangle \frac{\partial k}{\partial x_i} + \frac{\partial}{\partial x_i} \left(\frac{\langle u_i u_j u_j \rangle}{2} + \frac{\langle p' u_i \rangle}{\rho} - 2\nu \langle u_j s_{ij} \rangle \right) = \mathcal{P} - \varepsilon \quad 2.5.32$$

Durch Zusammenfassung vereinfacht sich die Transportgleichung zu:

$$\frac{\bar{D}k}{\bar{D}t} + \nabla \cdot \mathbf{T}' = \mathcal{P} - \varepsilon \quad 2.5.33$$

Die Dissipationsrate stellt dabei die einzige Senke dar, denn sie ist proportional zum Quadrat des Spannungstensors s_{ij} und somit positiv. Im k-ε-Modell wird die Dissipationsrate durch eine zweite Transportgleichung modelliert.

Transportterm

Der Transportterm wiederum ist wie folgt definiert:

$$\nabla \cdot \mathbf{T}' = \frac{\partial}{\partial x_i} \left(\frac{\langle u_i u_j u_j \rangle}{2} + \frac{\langle p' u_i \rangle}{\rho} - 2\nu \langle u_i s_{ij} \rangle \right) \quad 2.5.34$$

Auch \mathbf{T}' muss im Sinne des Schließungs-Problems modelliert werden. Um die von Jones und Launder (1972) getroffene Modellierung besser nachvollziehen zu können, bedarf es kleiner Umformungen des letzten Ausdrucks in 2.5.34:

$$2\nu \frac{\partial \langle u_j s_{ij} \rangle}{\partial x_i} = 2\nu \frac{\partial}{\partial x_i} \langle u_j \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right) \rangle = \nu \frac{\partial}{\partial x_i} \frac{\partial \langle u_j u_j \rangle}{\partial x_i} + 2\nu \frac{\partial^2 \langle u_i u_j \rangle}{\partial x_i \partial x_j} = \nu \frac{\partial^2 k}{\partial x_i^2} + 2\nu \frac{\partial^2 \langle u_i u_j \rangle}{\partial x_i \partial x_j} \quad 2.5.35$$

Der letzte Term beträgt maximal nur wenige Prozent von ε . Zur Erinnerung, der Ausdruck 2.5.35 stammt ebenso wie die Dissipationsrate aus Gleichung 2.5.30. Er kann daher in den meisten Strömungen vernachlässigt werden (S.B. Pope, 2000, p. 132).

Der Transportterm reduziert sich dadurch auf:

$$T'_i = \frac{\langle p' u_i \rangle}{\rho} - \nu \frac{\partial k}{\partial x_i} + \frac{\langle u_i u_j u_j \rangle}{2} \quad 2.5.36$$

Der erste Term von links ist ein turbulenter Transportmechanismus verursacht durch Druck- und Geschwindigkeitsfluktuationen. Diese Korrelation ist messtechnisch schwer zu erfassen und wurde daher von Anfang an nicht modelliert und insofern vernachlässigt (Wilcox, 1993). DNS-Berechnungen zeigten, dass dieser Term in einfachen Strömungen relativ klein ist (Mansour, Kim, & Moin, 1988) und bestätigten somit teilweise Launder und Spaldings Vorgehensweise.

⁵ k ist ein Skalar, durch die Einsteinsche Summenkonvention entspricht $\partial \langle p' u_j \rangle / \partial x_j = \partial \langle p' u_i \rangle / \partial x_i$

2.5 k-ε-Modell

Der zweite Term von links entspricht der molekularen Diffusion der turbulenten kinetischen Energie. Der letzte Term wiederum ist der turbulente Transport der turbulenten kinetischen Energie. Daher jener Transport, welcher durch die turbulenten Fluktuationen der Geschwindigkeit bewerkstelligt wird.

Der turbulente Transport von k wird im k-ε-Modell durch simple Gradienten-Diffusion modelliert. Gleichung 2.5.2 legt eine Diffusivität der Form $\Gamma_T = \nu_T / \sigma_k$ nahe. Mit $\sigma_k = \nu_T / \Gamma_T$ der turbulenten Prandtlzahl, deren Wert Eins beträgt. Der modellierte Transport Term entspricht insofern:

$$T_i' \approx - \left(\nu + \frac{\nu_T}{\sigma_k} \right) \frac{\partial k}{\partial x_i} \approx - \frac{\nu_T}{\sigma_k} \frac{\partial k}{\partial x_i} \quad 2.5.37$$

Wobei die molekulare Diffusion, bedingt durch ihren geringen Einfluss bei hohen Re , vernachlässigt werden kann.

Der Transportterm ist aus numerischen Gründen in zweierlei Hinsicht wichtig. Bei integraler Schreibweise der Transportgleichung kann $\nabla \cdot \mathbf{T}'$, mit dem Integralsatz von Gauß, in ein Oberflächenintegral umgewandelt werden. Der Transportterm ist insofern für die Implementierung von Randbedingungen wesentlich. Des Weiteren schafft er durch Diffusion einen Ausgleich zwischen dem Quell- und dem Senkenterm und ermöglicht somit stabile Lösungen.

Der Transportterm für ε erfüllt aus numerischer Sicht die gleichen Aufgaben. Er wird ebenfalls durch Gradientendiffusion modelliert und unterscheidet sich daher in seiner Form nicht von 2.5.37. Für eine Diskussion des Wertes der turbulenten Prandtlzahl, soll hier daher auf Kapitel 2.5.2.2 verwiesen werden, in dem der Transportterm für ε erläutert wird.

Die durch Modellierung geschlossene Form der k -Transportgleichung kann also zu nachstehendem Ausdruck zusammengefasst werden:

$$\frac{\bar{D}k}{\bar{D}t} = \nabla \cdot \left(\frac{\nu_T}{\sigma_k} \nabla k \right) + \mathcal{P} - \varepsilon \quad 2.5.38$$

Vergleich der Transportgleichungen der kinetischen Energien

Zum besseren Verständnis der auftretenden Terme seien hier noch die Transportgleichung für die gesamte und die mittlere kinetische Energie angeführt:

$$\frac{\bar{D}\langle E \rangle}{\bar{D}t} + \nabla \cdot (\langle \mathbf{u} E \rangle \mathbf{T}) = -\bar{\varepsilon} - \varepsilon \quad 2.5.39$$

$$\frac{\overline{DE}}{\overline{Dt}} + \nabla \cdot \overline{\mathbf{T}} = -\mathcal{P} - \bar{\varepsilon} \quad 2.5.40$$

Wie bereits beschrieben, scheint \mathcal{P} in 2.5.40 mit negativem Vorzeichen (vgl. 2.5.38) auf.

In 2.5.39 ist \mathcal{P} nicht enthalten. Die Transportterme sind dabei ähnlich aufgebaut:

$$T_i = U_i \frac{p}{\rho} - 2\nu U_j S_{ij} \quad 2.5.41$$

$$\bar{T}_i = \langle U_j \rangle \langle u_i u_j \rangle + \langle U_i \rangle \frac{\langle p \rangle}{\rho} - 2\nu \langle U_j \rangle \langle \bar{S}_{ij} \rangle \quad 2.5.42$$

Die Senke der Transportgleichung für die kinetische Energie der mittleren Strömung ist dabei folgende Dissipationsrate:

$$\bar{\varepsilon} \equiv 2\nu \langle S_{ij} S_{ij} \rangle \quad 2.5.43$$

Sie erfasst die Dissipation der kinetischen Energie der mittleren Strömung durch Reibung. Diese wird verursacht von mittleren, makroskopischen Geschwindigkeitsgradienten. Natürlich scheint sowohl $\bar{\varepsilon}$, als auch ε als Senke in der Transportgleichung der gesamten kinetischen Energie auf.

2.5.2.2 Transportgleichung für ε

Auch eine exakte Herleitung der Transportgleichung für die Dissipationsrate ist möglich. Eine physikalisch sinnvolle Berechnung erfordert jedoch die Auflösung der Kolmogorov Skalen. Die Entwicklung der Transportgleichung im k-ε-Modell kann daher als empirisch betrachtet werden.

Ihr schematischer Aufbau gleicht der Transportgleichung für k . Auf der linken Seite befindet sich die substantielle Ableitung. Die rechte Seite setzt sich additiv aus einem Transportterm, einem Quellterm und einer Senke zusammen.

$$\frac{\overline{D\varepsilon}}{\overline{Dt}} = \nabla \cdot \left(\frac{\nu_T}{\sigma_\varepsilon} \nabla \varepsilon \right) + C_{\varepsilon 1} \frac{\mathcal{P}_\varepsilon}{k} - C_{\varepsilon 2} \frac{\varepsilon^2}{k} \quad 2.5.44$$

Der Transportterm gleicht dabei dem Transportterm in der Erhaltungsgleichung für k . Die Modellkonstante σ_ε kann wiederum als eine turbulente Prandtlzahl interpretiert werden. Die Modellierung der einzelnen Terme auf der rechten Seite erfolgte auf Basis von experimentell bestimmten, simplen, turbulenten Benchmark-Strömungen. Die Form der drei Terme wird im Folgenden, anhand ihres Verhaltens in diesen unterschiedlichen Strömungen, erläutert.

Senke - Strömung mit homogener abklingender Turbulenz

Die Senke in der Erhaltungsgleichung für ε kann für eine Strömung mit homogener abklingender Turbulenz, gesondert vom Quell und Transportterm, betrachtet werden.

Bei homogener Turbulenz sind die fluktuierenden Größen $\mathbf{u}(\mathbf{x}, t)$ und $p'(\mathbf{x}, t)$ statistisch gleichmäßig im Raum verteilt. Homogene abklingende Turbulenz kann experimentell, in einem Windkanal, mittels einer einheitlichen Strömung U_0 in x -Richtung, unter Einsatz eines turbulenzerzeugenden Gitters mit der Maschenweite M , generiert werden.

Wie in Abbildung 2.6 ersichtlich, liegt die Turbulenz dabei annähernd isotrop vor. Die turbulente kinetische Energie k und somit der Turbulenzgrad nimmt in x -Richtung ab.

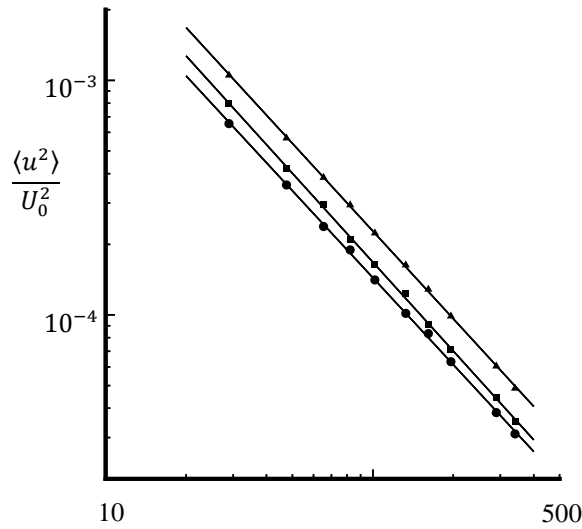


Abbildung 2.6 Abklingende Reynoldssche Spannungen einer turbulenten Gitterströmung: Quadrate $\langle u^2 \rangle / U_0^2$; Kreise $\langle v^2 \rangle / U_0^2$; Dreiecke k / U_0^2 ; Linien proportional zu $(x/M)^{-1.3}$ (Comte-Bellot & Corrsin, 1966).

Die Abnahme von k ist dabei proportional zu einem Potenzgesetz $(x/M)^{-1.3}$. Mit der linearen Beziehung $t = x/U_0$ kann aus dem räumlichen Verlauf des Potenzgesetzes ein zeitlicher Verlauf ermittelt werden.

$$\frac{k}{U_0^2} = A \left(\frac{x}{M} \right)^{-1.3} = A \left(\frac{x}{M} \frac{U_0}{U_0} \right)^{-1.3} = A \left(\frac{t}{B} \right)^{-1.3} \quad 2.5.45$$

Mathematisch betrachtet, verschwinden in einer homogenen, turbulenten Strömung die räumlichen Ableitungen der Turbulenzgrößen k und ε . Liegen zudem keine Gradienten der mittleren Geschwindigkeit vor, fällt auch der Produktionsterm \mathcal{P} weg, die homogene Turbulenz klingt ab. Die Transportgleichung für k reduziert sich auf:

$$\frac{dk}{dt} = -\varepsilon \quad 2.5.46$$

2.5 k-ε-Modell

Insofern entspricht der reale Verlauf von ε ebenfalls einem Potenzgesetz.

$$\frac{dk}{dt} = -\varepsilon = -1.3 \frac{A}{U_0^2} \left(\frac{t}{B}\right)^{-2.3} \quad 2.5.47$$

Das reale Verhalten kann durch die getroffene Modellierung der Transportgleichung für ε gut wiedergegeben werden. Diese reduziert sich aufgrund der homogenen Turbulenz und der nicht vorhandenen Produktion \mathcal{P} zu:

$$\frac{d\varepsilon}{dt} = -C_{\varepsilon 2} \frac{\varepsilon^2}{k} \quad 2.5.48$$

Gültige Lösungen für das vorliegende Differentialgleichungssystem (Gleichung 2.5.46 und 2.5.48) sind insofern Potenzgesetze für k und ε , die den realen Verläufen 2.5.45 und 2.5.47 genügen:

$$k(t) = k_0 \left(\frac{t}{t_0}\right)^{-n}, \quad \varepsilon(t) = \varepsilon_0 \left(\frac{t}{t_0}\right)^{-(n+1)} \quad 2.5.49$$

mit den Konstanten:

$$k_0 = k(t_0), \quad \varepsilon_0 = \varepsilon(t_0), \quad t_0 = n \frac{k_0}{\varepsilon_0}, \quad n = \frac{1}{C_{\varepsilon 2} - 1} \quad 2.5.50$$

Durch Vergleich von 2.5.49 mit 2.5.45 kann der Wert des Exponentenkoeffizient $n = 1.3$ bestimmt werden. Aus 2.5.50 erhalten wir somit für $C_{\varepsilon 2}$:

$$C_{\varepsilon 2} = \frac{n+1}{n} = 1.77 \quad 2.5.51$$

Experimente zeigten, dass n im Bereich von 1.15-1.45 liegt. Die entsprechenden Werte für $C_{\varepsilon 2}$ betragen dabei 1.87-1.69. Nach Mohamed und Larue (1990) stimmen die meisten Ergebnisse mit $n = 1.3$ gut überein. Der Standardwert für $C_{\varepsilon 2} = 1.92$ unterschätzt die Messwerte des Koeffizienten n .

Quelle - Scherströmung mit homogener Turbulenz

Das Verhalten des Quellterms in der Erhaltungsgleichung für ε kann gut anhand einer Scherströmung mit homogener Turbulenz nachvollzogen werden.

Durch Steuerung des Strömungswiderstandes stromaufwärts kann im Windkanal näherungsweise eine Scherströmung mit dem Geschwindigkeitsfeld $\langle \mathbf{U} \rangle = (\langle U \rangle, 0, 0)^T$ und der konstanten Scherrate $\mathcal{S} = \partial \langle U \rangle / \partial x_i = (0, \mathcal{S} = \partial \langle U \rangle / \partial y, 0)^T$ erzeugt werden.

Abbildung 2.7 illustriert die beschriebene Scherströmung. Die homogene Turbulenz wird wiederum durch ein Gitter approximiert.

2.5 k-ε-Modell

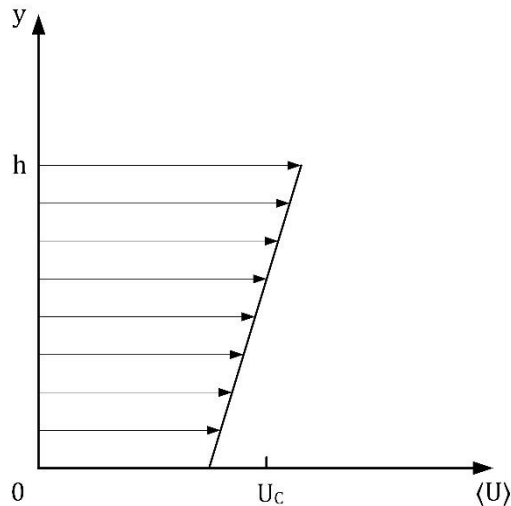


Abbildung 2.7 Schematische Darstellung der Scherströmung

Da in diesen Strömungen die Produktion die Dissipation übersteigt, liegt die homogene Turbulenz nur in y - und z -Richtung vor. Nach Abbildung 2.8 steigen in x -Richtung die Reynoldsschen Spannungen an.

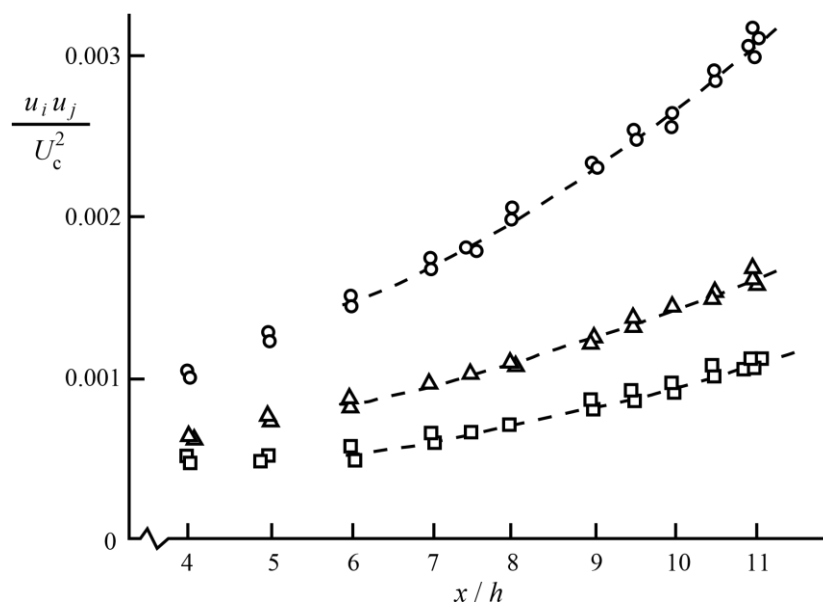


Abbildung 2.8 Reynoldssche Spannungen über dem axialen Abstand zum Gitter, in einer homogenen Scherströmung: Quadrate $\langle v^2 \rangle$; Kreise $\langle u^2 \rangle$; Dreiecke $\langle w^2 \rangle$; (Tavoularis & Corrsin, 1981).

Experimente gaben Aufschluss darüber, dass das Verhältnis zwischen Produktion und Dissipation \mathcal{P}/ε bei Änderung von x konstant bleibt. Dieses Verhalten zeigt auch das Verhältnis zwischen der turbulenten Zeitskala $\tau = k/\varepsilon$ und der Zeitskala der gemittelten Strömung bzw. Scherrate $\mathcal{S}^{-1} = \mathcal{S}k/\varepsilon$. Tabelle 2.1 fasst einen Teil dieser experimentellen Ergebnisse zusammen:

2.5 k-ε-Modell

Tabelle 2.1 Experimentell generierte Ergebnisse für eine turbulente, homogene Scherströmung (Tavoularis & Corrsin, 1981).

	$x/h = 7.5$	$x/h = 11.0$
\mathcal{P}/ε	1.8	1.7
Sk/ε	6.5	6.1

In einem mit der charakteristischen konvektiven Geschwindigkeit U_c bewegten Koordinatensystem, ist die Strömung in allen Richtungen homogen turbulent. Die Erhaltungsgleichung für k reduziert sich auf:

$$\frac{dk}{dt} = \mathcal{P} - \varepsilon \quad 2.5.52$$

Aufgrund der konstanten Scherrate S können wir anhand der Ergebnisse aus Tabelle 2.1 von einer konstanten turbulenten Zeitskala $\tau = k/\varepsilon$ ausgehen. Gleichung 2.5.52 lässt sich daher wie folgt umformen:

$$\frac{\tau}{k} \frac{dk}{dt} = \frac{\mathcal{P}}{\varepsilon} - 1 \quad 2.5.53$$

Mit $\mathcal{P}/\varepsilon = \text{const}$ kann durch Integration k nach Gleichung 2.5.54 angeschrieben werden:

$$k(t) = k(0) \exp \left[\left(\frac{\mathcal{P}}{\varepsilon} - 1 \right) \frac{t}{\tau} \right] \quad 2.5.54$$

Da $\mathcal{P}/\varepsilon > 1$ ist, steigt nach 2.5.54 k exponentiell an. Durch $\varepsilon = k/\tau$ steigt ε ebenfalls exponentiell an.

$$\varepsilon(t) = \frac{k(t)}{\tau} = \frac{k(0)}{\tau} \exp \left[\left(\frac{\mathcal{P}}{\varepsilon} - 1 \right) \frac{t}{\tau} \right] \quad 2.5.55$$

Dieses Verhalten wird auch von der Modellgleichung für ε beschrieben. Diese vereinfacht sich, mit dem beschriebenen Koordinatensystem, für eine homogene, turbulente Scherströmung zu:

$$\frac{d\varepsilon}{dt} = C_{\varepsilon 1} \frac{\mathcal{P}\varepsilon}{k} - C_{\varepsilon 2} \frac{\varepsilon^2}{k} = \varepsilon \left(C_{\varepsilon 1} \frac{\mathcal{P}}{\tau} - C_{\varepsilon 2} \frac{1}{\tau} \right) \quad 2.5.56$$

Hierbei ist der Ausdruck in der Klammer konstant. Durch Trennung der Variablen und Integration kann ε bestimmt werden:

$$\varepsilon(t) = \varepsilon(0) \exp \left[\left(\frac{C_{\varepsilon 1} \mathcal{P}}{\varepsilon} - C_{\varepsilon 2} \right) \frac{t}{\tau} \right] \quad 2.5.57$$

$C_{\varepsilon 1}$ lässt sich durch vergleich der Exponenten aus 2.5.55 und 2.5.57 wie folgt ausdrücken:

$$C_{\varepsilon 1} = \frac{\mathcal{P}/\varepsilon - 1 + C_{\varepsilon 2}}{\mathcal{P}/\varepsilon} \quad 2.5.58$$

Mit dem experimentell gemessenen $\mathcal{P}/\varepsilon = 1.7$ und $C_{\varepsilon 2} = 1.77$ erhalten wir für $C_{\varepsilon 1} = 1.45$. Das Verhältnis \mathcal{P}/ε wird daher vom Standardparametersatz $C_{\varepsilon 1} = 1.44$, $C_{\varepsilon 2} = 1.92$ mit einem Wert von 2.09 stark überschätzt.

Transportterm - vollausgebildete turbulente Scherströmung im Bereich des logarithmischen Wandgesetzes

Zur Beschreibung des Transportterms kann eine vollausgebildete turbulente Scherströmung herangezogen werden. Diese entspricht näherungsweise einer Strömung in einem rechteckigen Kanal (schematisch dargestellt in Abbildung 2.9). Hinreichend weit entfernt vom Einlauf, hat die mittlere Strömung nur eine Geschwindigkeitskomponente in x -Richtung ($\langle U \rangle, 0, 0)^T$ und die Scherrate $\partial \langle U \rangle / \partial x_i = (0, S = \partial \langle U \rangle / \partial y, 0)^T$. Mit $b \gg h = 2d$ liegt für große x und kleine z ein eindimensionales und stationäres Problem in y -Richtung vor.

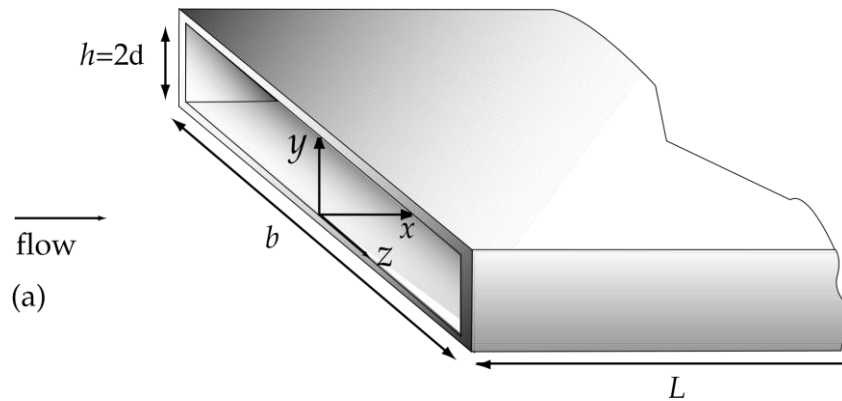


Abbildung 2.9 Skizze der Strömung durch den rechteckigen Kanal (S.B. Pope, 2000).

Damit der Transportterm in einer Strömung auftritt, muss der darin enthaltene Gradienten $\nabla \varepsilon$ ungleich 0 sein. Dies ist durch Wandeinflüsse, die im logarithmischen Wandbereich $y^+ > 30$, $y/d < 0.3$ zu tragen kommen, möglich.

In diesem lässt sich das Geschwindigkeitsprofil wie folgt durch ein logarithmisches Gesetz beschreiben:

$$u^+ = \frac{1}{\kappa} \ln y^+ + B \quad \left(\rightarrow \frac{du^+}{dy^+} = \frac{1}{\kappa y^+} \right) \quad 2.5.59$$

Die dabei auftretenden Variablen sind die dimensionslose Geschwindigkeit $u^+ = \langle U \rangle / u_\tau$, die dimensionslosen Länge $y^+ = y / \delta_v = u_\tau y / \nu$, die Konstante $B = 5.2$, sowie die von Kármán-Konstante $\kappa = 0.41$. Die hierfür herangezogenen konstanten viskosen Skalierungen ($Re = u_\tau \delta_v / \nu = 1$) sind die Schubspannungsgeschwindigkeit $u_\tau = \sqrt{\tau_w / \rho}$ (mit der Wandschubspannung τ_w) und die Rauigkeitslänge δ_v (Kármán, 1930).

2.5 k-ε-Modell

Außerdem kann aus Abbildung 2.10 entnommen werden, dass im logarithmischen Wandbereich Produktion und Dissipationsrate etwa gleich groß sind ($\mathcal{P} = \varepsilon$).

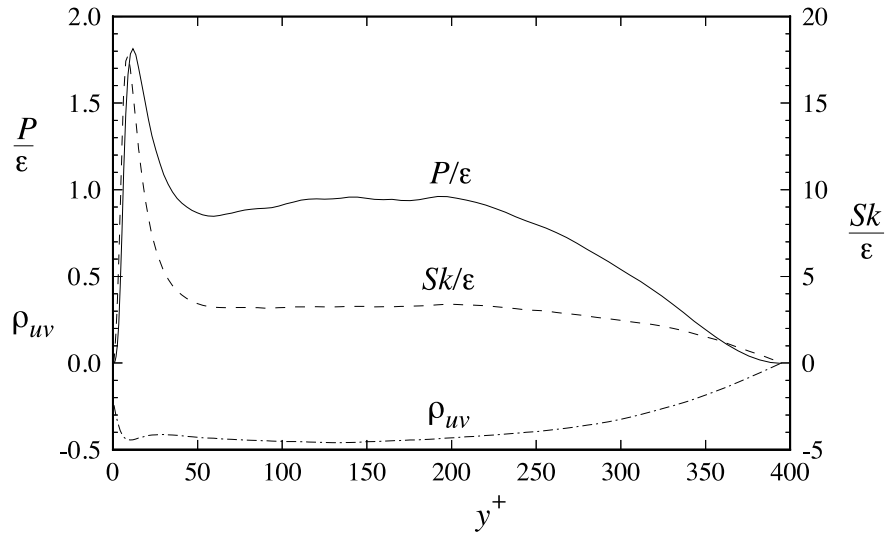


Abbildung 2.10 Verhältnis von Produktion zu Dissipationsrate und anderer Strömungseigenschaften in Abhängigkeit vom dimensionslosen Wandabstand y^+ , Berechnungen einer Kanalströmung mittels DNS (Kim et al., 1987).

Demnach reduziert sich die Transportgleichung von k auf:

$$0 = \frac{d}{dy} \cdot \left(\frac{\nu_T}{\sigma_k} \frac{dk}{dy} \right) + \mathcal{P} - \varepsilon = \frac{d}{dy} \cdot \left(\frac{\nu_T}{\sigma_k} \frac{dk}{dy} \right) \quad 2.5.60$$

Der Gradient von k entspricht im logarithmischen Wandbereich also Null. Dies ist nach Abbildung 2.11 auch näherungsweise der Fall.

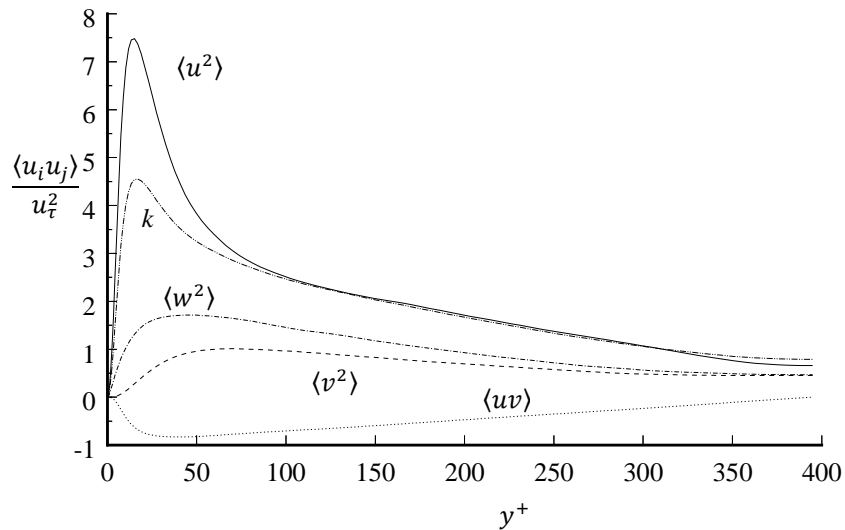


Abbildung 2.11 Turbulente kinetische Energie und Reynoldssche Spannungen in Abhängigkeit vom dimensionslosen Wandabstand y^+ , Berechnungen einer Kanalströmung mittels DNS (Kim et al., 1987).

2.5 k-ε-Modell

Die Erhaltungsgleichung für ε kann wiederum auf folgenden Ausdruck vereinfacht werden:

$$0 = \frac{d}{dy} \cdot \left(\frac{\nu_T}{\sigma_\varepsilon} \frac{d\varepsilon}{dy} \right) + C_{\varepsilon 1} \frac{P_\varepsilon}{k} - C_{\varepsilon 2} \frac{\varepsilon^2}{k} = \frac{d}{dy} \cdot \left(\frac{\nu_T}{\sigma_\varepsilon} \frac{d\varepsilon}{dy} \right) + (C_{\varepsilon 1} - C_{\varepsilon 2}) \frac{\varepsilon^2}{k} \quad 2.5.61$$

Mit $C_{\varepsilon 1} < C_{\varepsilon 2}$ liegt netto eine Senke von ε vor. Wie im Anschluss gezeigt werden soll, variiert diese mit y^{-2} ($k = \text{const}, \varepsilon = f(y^{-1})$). Gegen diesen Gradienten wirkt der Transportterm mittels Diffusion.

Laut der Hypothese der turbulenten Viskosität gilt im logarithmischen Wandbereich unter Verwendung von 2.5.18 ($\nu_T = C_\mu k^2 / \varepsilon$):

$$\langle uv \rangle = -\nu_T \frac{\partial \langle U \rangle}{\partial y} = -\frac{C_\mu k^2}{\varepsilon} \frac{\partial \langle U \rangle}{\partial y} \quad 2.5.62$$

Die Ableitung der mittleren Geschwindigkeit in 2.5.62 kann insofern durch das logarithmische Wandgesetz vereinfacht werden:

$$\frac{\partial \langle U \rangle}{\partial y} = \frac{u_\tau^2}{\nu} \frac{du^+}{dy^+} = \frac{u_\tau^2}{\nu \kappa y^+} = \frac{u_\tau}{\kappa y} \quad 2.5.63$$

Die Reynoldsschen Spannungen können im logarithmischen Wandbereich wie folgt abgeschätzt werden (siehe Abbildung 2.12):

$$\langle uv \rangle \approx \tau_w / \rho \approx -u_\tau^2 \quad 2.5.64$$

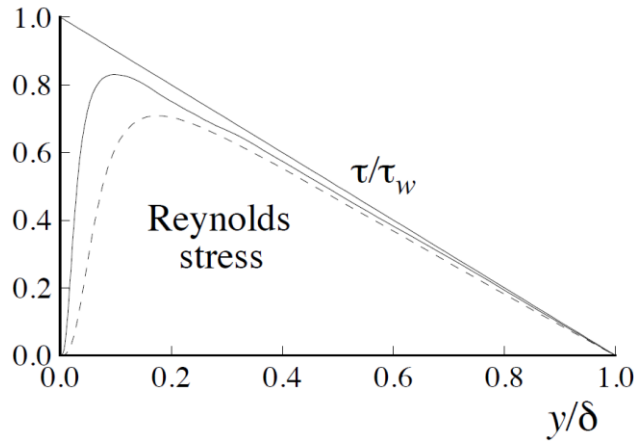


Abbildung 2.12 Profile der Anisotropen Reynoldsschen Spannungen bei unterschiedlichen Reynoldszahlen: strichliert, $Re = 5.600$; durchgezogen, $Re = 13.750$, mit $\delta = d$ der halben Kanalhöhe: DNS-Ergebnisse (Kim et al., 1987)

Die Dissipationsrate lässt sich durch Substitution von 2.5.63 und 2.5.64 in 2.5.62 wie folgt ausdrücken:

$$\varepsilon = \frac{C_\mu k^2}{u_\tau \kappa y} = f(y^{-1}) \quad 2.5.65$$

2.5 k-ε-Modell

Da Produktion und Dissipationsrate etwa gleich groß sind gilt:

$$\varepsilon = \mathcal{P} = -\langle uv \rangle \frac{\partial \langle U \rangle}{\partial y} = \frac{c_\mu k^2}{u_\tau \kappa y} \quad 2.5.66$$

Durch Einsetzen von 2.5.63 und 2.5.64 in 2.5.66 kann Ausdruck 2.5.67 erhalten werden:

$$u_\tau^2 = C_\mu^{1/2} k \quad 2.5.67$$

Substitution von 2.5.18, 2.5.65 und 2.5.67 in die Erhaltungsgleichung für ε ergibt:

$$0 = \frac{d}{dy} \cdot \left[\frac{\left(\frac{c_\mu k^2}{\left(\frac{c_\mu k^2}{u_\tau \kappa y} \right)} \right)}{\sigma_\varepsilon} \frac{d}{dy} \left(\frac{c_\mu k^2}{u_\tau \kappa y} \right) \right] + (C_{\varepsilon 1} - C_{\varepsilon 2}) \frac{1}{k} \left(\frac{c_\mu k^2}{\left(\frac{c_\mu^{1/2} k}{\kappa y} \right)} \right)^2 = -\frac{d}{dy} \cdot \left[\frac{c_\mu k^2}{\sigma_\varepsilon y} \right] + (C_{\varepsilon 1} - C_{\varepsilon 2}) \frac{c_\mu^{3/2} k^2}{\kappa^2 y^2} = \quad 2.5.68$$

$$\frac{c_\mu k^2}{\sigma_\varepsilon y^2} + (C_{\varepsilon 1} - C_{\varepsilon 2}) \frac{c_\mu^{3/2} k^2}{\kappa^2 y^2}$$

Durch Umformung erhalten wir schlussendlich:

$$\kappa^2 = \sigma_\varepsilon C_\mu^{1/2} (C_{\varepsilon 2} - C_{\varepsilon 1}) \leftrightarrow 0.43^2 \approx 1.3 \cdot 0.09^{1/2} (1.92 - 1.44) \quad 2.5.69$$

$$0.41^2 \approx 1.75 \cdot 0.09^{1/2} (1.77 - 1.45)$$

Wird in Gleichung 2.5.69 der Standardparametersatz eingesetzt nimmt die Kármán-Konstante einen Wert von 0.43 an. Dieser liegt leicht über dem oftmals üblichen Wert von 0.41. Mit $\kappa = 0.41$ und den aus Gleichung 2.5.51 und 2.5.58 gewonnenen Werten für $C_{\varepsilon 2} = 1.77$ bzw. $C_{\varepsilon 1} = 1.45$ ergibt sich σ_ε zu 1.75. Eine konsistente Beschreibung der Standardparameter ist insofern durch die vorgestellten Benchmark-Strömungen in Kapitel 2.5.2.2 nicht möglich.

Die Standardwerte für σ_ε und σ_k , die mit 1.30 bzw. 1.00 nahe an der Eins liegen, lassen sich durch die folgenden Überlegungen erklären:

Die Prandtlzahl $Pr = \nu/a$ setzt den Impulstransport in Folge von viskosen Spannungen ν , mit dem Wärmetransport (Temperaturleitfähigkeit a) ins Verhältnis. Sie vergleicht in gewisser Weise die Geschwindigkeitsverteilung mit der Verteilung eines Skalars (Temperatur). Die turbulente Prandtlzahl vergleicht wiederum den turbulenten Impulstransport in Folge von Scheinspannungen ν_T , mit dem turbulenten Transport eines Skalars. Der turbulente Transport von k und ε ist in den Erhaltungsgleichungen durch Diffusion modelliert. Die turbulente Prandtlzahl für k und ε setzt insofern die turbulente Viskosität mit den Diffusivitäten ins Verhältnis ($Pr_T = \nu_T/\Gamma_T$). Bei einer turbulenten Prandtlzahl von 1, erfolgt der turbulente Skalartransport analog zum turbulenten Impulstransport. Auf diesen Sachverhalt wird im Weiteren genauer eingegangen:

2.5 k-ε-Modell

Zur Bestimmung der turbulenten Prandtlzahl eignet sich die Temperatur als transportiertes Skalar. Die Diffusivität Γ_T entspricht dann der turbulenten Temperaturleitfähigkeit $a_T \rightarrow Pr_T = \sigma_T = \nu_T/a_T$.

Im logarithmischen Wandbereich, gilt für die vorliegenden Spannungen $\tilde{\tau}$ bzw. für die vorhandene Wärmestromdichte q (Blom & de Vries, 1970):

$$\frac{\tilde{\tau}}{\rho} = (\nu + \nu_T) \frac{\partial \langle U \rangle}{\partial y} = \nu_{eff} \frac{\partial \langle U \rangle}{\partial y} \quad 2.5.70$$

$$\frac{q}{\rho c_p} = -(a + a_T) \frac{\partial \langle T \rangle}{\partial y} = -a_{eff} \frac{\partial \langle T \rangle}{\partial y} \quad 2.5.71$$

Mit der viskosen Temperatur $T_\tau = q_w/(\rho c_p u_\tau)$, die auf die Wandtemperatur T_w als Referenztemperatur bezogen ist, können die obigen Gleichungen, analog zu 2.5.63, in ihre dimensionslose Form überführt werden:

$$\frac{\tilde{\tau}}{\tau_w} = \frac{\nu_{eff}}{\nu} \frac{du^+}{dy^+} \approx \frac{\nu_T}{\nu} \frac{du^+}{dy^+} \quad 2.5.72$$

$$\frac{q}{q_w} = \frac{a_{eff}}{\nu} \frac{dT^+}{dy^+} \approx \frac{a_T}{\nu} \frac{dT^+}{dy^+} = \frac{\nu_T}{\nu} \frac{1}{\sigma_T} \frac{dT^+}{dy^+} \quad 2.5.73$$

Dabei ist die dimensionslose Temperatur $T^+ = (T_w - T)/T_\tau$ ebenfalls auf T_w bezogen. Im vollturbulenten Bereich kann außerdem der molekulare Impuls- und Wärmetransport gegenüber dem turbulenten Transport vernachlässigt werden. Mit 2.5.72 und 2.5.73 kann ν_T/ν eliminiert werden:

$$\frac{dT^+}{dy^+} = \frac{q}{q_w} \frac{\tau_w}{\tilde{\tau}} \sigma_T \frac{du^+}{dy^+} \quad 2.5.74$$

Unter der in Wandnähe näherungsweise gültigen Vereinfachungen $\tau_w/\tilde{\tau} \approx 1$ und $q/q_w \approx 1$, kann unter der Annahme $Pr_T = const$, das Temperaturprofil durch Integration erhalten werden.

$$T^+ = \sigma_T(u^+ + C) = \sigma_T \left(\frac{1}{\kappa} \ln y^+ + B + C \right) = A_T \ln y^+ + B_T \quad 2.5.75$$

Für $\sigma_T = 1$ unterscheidet sich das dimensionslose Temperaturprofil nur um einen konstanten Offset C von dem dimensionslosen Geschwindigkeitsprofil. Der turbulente Temperatur-/Skalartransport erfolgt daher analog zum turbulenten Impulstransport. Aus Tabelle 2.2 ist zu entnehmen, dass für den logarithmischen Wandbereich die Werte für die turbulente Prandtlzahl nahe bei 1 liegen.

2.5 k-ε-Modell

Tabelle 2.2 Auszug experimentell bestimmter Daten für A_T und B_T , sowie berechnete Werte für Pr_T (Blom & de Vries, 1970).

Strömung	Rohr	Rohr	Rohr	Rohr	Rohr	Rohr	Rohr	Rohr	Ebene Platte	Kanalströmung
A_T	2.2	2.04	2.55	2.18	2.58	2.52	2.46	2.11	2.07	1.96
B_T	3.8	-7.25	28	3	34.5	76.3	-13.6	4.7	3.85	3.8
Pr_T	0.88	0.82	1.02	0.87	1.03	1.01	0.98	0.84	0.83	0.78

2.5.3 Diskussion des k-ε-Modells

Das k-ε-Modell ist eines der simpelsten, kompletten Turbulenzmodelle. Denn durch die Transportgleichungen für k und ε , bedarf es keiner algebraischen Beschreibung der Geschwindigkeits- $u^* \approx k^{1/2}$ und Längenskalierung $l^* \approx k^{3/2}/\varepsilon$, wobei auch eine Zeitskalierung $\tau \approx k/\varepsilon$ vorliegt. Demnach ist es eines der am häufigsten angewendeten Turbulenzmodelle.

Die dem k-ε-Modell zugrundeliegenden Annahmen sind jedoch schon bei einfachen Strömungen, wie etwa den Benchmark-Strömungen in Kapitel 2.5.2.2, nicht zulässig.

2.5.3.1 Diskussion - Hypothese der Gradienten Diffusion

Nach der Hypothese der Gradienten Diffusion, erfolgt der turbulente Transport eines Skalars entgegen dessen Gradienten. In homogenen, turbulenten Scherströmungen konnte jedoch ein Winkel von 65° zwischen $\nabla\phi$ und $-\langle u\phi' \rangle$ gemessen werden (Tavoularis & Corrsin, 1981).

2.5.3.2 Diskussion - Hypothese der turbulenten Viskosität

Analog gilt für die Hypothese der turbulenten Viskosität, dass der anisotrope, turbulente Scheinspannungstensor die gleiche Ausrichtung wie der Scherratenstensor aufweisen ($-a_{ij} = 2\nu_T \bar{S}_{ij}$). Dies trifft ebenfalls nicht bei einfachen, homogen turbulenten Scherströmungen ($\partial\langle U \rangle/\partial x_i = (0, \partial\langle U \rangle/\partial y, 0)^T$) zu (S.B. Pope, 2000).

$$-a_{ij} = -k \begin{bmatrix} 0.4133 & -0.32 & 0 \\ -0.32 & -0.2667 & 0 \\ 0 & 0 & -0.1467 \end{bmatrix} \neq 2\nu_T \begin{bmatrix} 0 & 1/2 \cdot \partial\langle U \rangle/\partial y & 0 \\ 1/2 \cdot \partial\langle U \rangle/\partial y & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = 2\nu_T \bar{S}_{ij} \quad 2.5.76$$

2.5.3.3 Diskussion - Transportgleichung für ε

Die modellierte Transportgleichung für ε wird oftmals bei schlechter Performance des k-ε-Modells verantwortlich gemacht. Für viele Strömungen (insbesondere der Benchmark-Strömungen) kann durch Parameteranpassung ($C_{\varepsilon 1}, C_{\varepsilon 2}$), sowie Korrekturterme, eine

2.5 k- ϵ -Modell

Verbesserung der Ergebnisse bewerkstelligt werden. Es konnte noch keine Anpassung des k- ϵ -Modells gefunden werden, die in einer weiten Bandbreite an Strömungen vorteilhaft wäre.

2.5.3.4 Regionen in Wandnähe

Regionen in Wandnähe zeichnen sich durch folgende Eigenschaften aus (Ferziger & Peric, 2008):

- i. Die Reynoldszahl sinkt in Wandnähe, eine Turbulenzmodellierung ist für niedrige Reynoldszahlen jedoch nur bedingt sinnvoll.
- ii. isotrope Turbulenz (eine Voraussetzung für Abschätzungen auf Basis des Prandtlschen Mischungsweges und der Energiekaskade) ist in Wandnähe nicht gewährleistet ($\langle v^2 \rangle = f(y^4)$, $\langle u^2 \rangle$ & $\langle w^2 \rangle = f(y^2)$). Auch der Wert des Parameters $C_{\epsilon 2}$ wurde durch Untersuchungen annähernd isotroper, homogen turbulenter Strömungen gewonnen.
- iii. Die höchsten Scherraten treten in der Nähe der Wand auf. Es kommt daher zu Spitzen der Verläufe von k und ϵ .

Aus oben genannten Gründen, ist die Anwendung des k- ϵ -Modells in Wandnähe problematisch. Um den Widrigkeiten entgegen zu wirken, bedarf es in diesem Bereich einer hohen Gitterauflösung, sowie einer Anpassung des k- ϵ -Modells.

Ein Vergleich des k- ϵ -Modells mit DNS-Berechnungen zeigt (siehe Abbildung 2.13), dass die turbulente Viskosität durch das k- ϵ -Modell stark überschätzt wird. Durch Substitution von einem k mit $\langle v^2 \rangle$ und Anpassung der Konstante C_μ , kann für $\nu_T = C_\mu k^2 / \epsilon \leftrightarrow C'_\mu (k \langle v^2 \rangle) / \epsilon$ der starken Variation von $\langle v^2 \rangle = f(y^4)$ Rechnung getragen werden.

2.5 k-ε-Modell

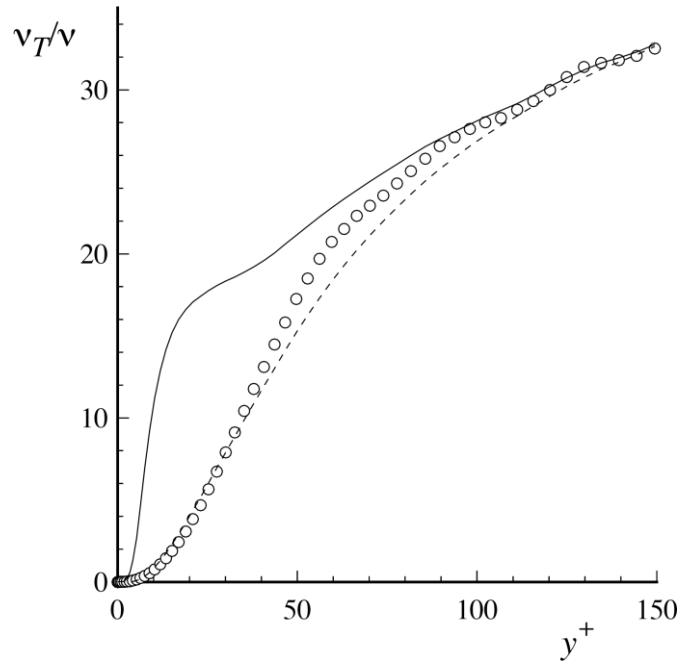


Abbildung 2.13 Turbulente Viskosität in Abhängigkeit von y^+ in einer Kanalströmung mit $Re = 13750$: Kreise DNS-Ergebnisse (Kim et al., 1987); Linie $0.09k^2/\varepsilon$; strichlierte Linie $0.22\langle v^2 \rangle k/\varepsilon$.

Weitere Ansätze sind der Einsatz von Dämpfungsfunktionen $f_\mu = f(Re)$, welche bereits in der ursprünglichen Veröffentlichung von Jones und Launder (1972) vorgestellt wurden. Nachfolgend ein Beispiel:

$$\nu_T = f_\mu C_\mu \frac{k^2}{\varepsilon} \quad 2.5.77$$

Diese entbehren jedoch jeglicher physikalischen Grundlage und sind nicht universell einsetzbar.

Der Variation von C_μ in Wandnähe (siehe Abbildung 2.5) wird im realizable k-ε-Turbulenzmodell Rechnung getragen. In diesem Modell wird, neben einer kleinen Anpassung des Senkenterms in der ε -Transportgleichung, C_μ durch eine Funktion beschrieben (Souza et al., 2017).

Außerdem wurden angepasste Transportgleichungen für den wandnahen Bereich mit niedrigen Reynoldszahlen entwickelt.⁶ Für diese kann die Randbedingung für ε an der Wand W aus der Transportgleichung von k gewonnen werden. Die dafür notwendigen

⁶ Zum Beispiel wurde in den Transportgleichungen die Zeitskala $\tau = k/\varepsilon$, welche in Richtung zur Wand gegen Null konvergiert ($k_W = 0$), mit $T = \max(\tau, 6\tau_\eta)$ substituiert. T liefert im Wandbereich mindestens das sechsfache der Kolmogorov-Zeitskalen, als kleinste relevante Zeitskalierung.

2.5 k-ε-Modell

Annahmen sind $\mathcal{P}_W = 0$, $(\mathbf{u}_W = \mathbf{0})$ und $(\bar{D}k/\bar{D}t)_W = 0$ (gültig für stationäre Strömungen hinreichend weit entfernt vom Einlauf).

$$\varepsilon \approx \nu \left(\frac{\partial^2 k}{\partial n^2} \right)_W \quad 2.5.78$$

Die turbulente kinetische Energie an der Wand k_W ist dabei Null.

$$k_W = 0 \text{ mit } (\mathbf{0} = \mathbf{U}_W = \langle \mathbf{U}_W \rangle + \mathbf{u}_W = \mathbf{0} + \mathbf{0}) \quad 2.5.79$$

Ansätze dieser Art werden *low-Re turbulence models* oder *low-Re wall functions* genannt. Sie bedürfen einer hohen Gitterauflösung in Wandnähe. Ein erheblicher Anteil des Rechenaufwandes fällt durch diese Modelle für die wandnahen Regionen an.

Neuere Ansätze bedürfen einer niedrigeren Gitterauflösung in Wandnähe. Sie legen in die wandnächste Zellschicht logarithmische Geschwindigkeitsverläufe und definieren damit nicht nur Werte an der Wand, sondern für die ganze Zelle. Die Transportgleichungen von k und ε werden dabei ebenfalls an das logarithmische Wandgesetz angepasst. Solche *high-Re Turbulence Models* oder *high-Re wall functions* genannten *Boundarylayers* basieren daher auf Überlegungen, welche bereits in Kapitel 2.5.2.2 (- Transportterm) erläutert wurden.

Für ε lässt sich durch Einsetzen von 2.5.67 in 2.5.65 folgender Wert für den Zellmittelpunkt P und somit für eine Zelle berechnen:

$$\varepsilon_P = \frac{C_\mu^{3/4} k_P^{3/2}}{\kappa y_P^+} \quad 7 \quad 2.5.80$$

Zur Bestimmung von ε_P bedarf es k_P . Der Wert kann durch die Transportgleichung für k berechnet werden. Wie in Kapitel 2.5.3.4 beschrieben, kann im logarithmischen Wandbereich in der Gleichung für k der diffusive Transport Null gesetzt werden. Die Produktion lässt sich durch Kombination von 2.5.66 und 2.5.67 abschätzen. Für die Senke wiederum wird Gleichung 2.5.80 eingesetzt.

Durch high-Re Turbulence Models kann der Rechenaufwand in Wandnähe stark reduziert werden. Unter vielen Strömungsbedingungen, wie zum Beispiel bei starken

⁷ Dies ähnelt bei Dimensionsvergleich $\kappa y_P^+ \approx l_0$ der geläufigen Eintrittsrandbedingung $\varepsilon_{in} = C_\mu^{0.75} k_{in}^{1.5} / l_0$, mit $k_{in} = 3/2 (\mathcal{I} U_{in})$. Der Turbulenzgrad für eine isotrop turbulente Strömung $\mathcal{I} = \sqrt{\langle uu \rangle} / U$ wird oft mit 5-10% abgeschätzt.

Druckgradienten, ist die physikalische Grundlage jedoch unsicher und die Genauigkeit unzureichend (Wilcox, 1993).

2.5.3.5 Genauigkeit und Eignung

Das k- ϵ -Modell kann auf alle turbulenten Strömungen angewandt werden. Da das Modell die Längen- und Zeitskalierung turbulenter Strömungen beschreibt, kann es außerdem zur Modellierung anderer Prozesse (Reaktionen, Mehrphasenströmungen, usw.) herangezogen werden.

Im Folgenden sollen für unterschiedliche Strömungen die Eignung des k- ϵ -Modell beschrieben werden (Menter, 1994):

- i. Das k- ϵ -Modell ist für dünne, zweidimensionale Scherströmungen mit geringer Stromlinienkrümmung und kleinen Druckgradienten gut geeignet.
- ii. In Grenzschichten mit hohen Druckgradienten ist das k- ϵ -Modell ungeeignet.
- iii. Bei Strömungen, die weit von Scherströmungen abweichen,, kann das k- ϵ -Modell versagen.

Erwähnenswert ist an dieser Stelle das k- ω -Modell. Dieses ist ebenfalls ein Zweigleichungsmodell, auf Basis der Hypothese der turbulenten Viskosität. In den meisten Strömungen ist das k- ω -Modell dem k- ϵ -Modell überlegen (Wilcox, 1993). In Strömungen mit ausgeprägter Stromlinienkrümmung, starken Wirbeln und sprunghaften Variationen des mittleren Geschwindigkeitsfeldes, sind Reynoldsspannungsmodelle diesen Zweigleichungsmodellen vorzuziehen (Bradshaw, Launder, & Lumley, 1996).

2.5.3.6 Parameterwahl

Das k- ϵ -Modell wird Jones und Launder (1972) zugeschrieben. Sie entwickelten das Modell zur Simulation von Grenzschichtströmungen. In ihrer Veröffentlichung gingen sie nicht auf die Herkunft der Parameter ein. Erst in einem Artikel von Jones und Launder (1973) wird die Parameterbestimmung kurz erklärt und für weitere Details auf Hanjalic und Launder (1972) verwiesen. Die Parameterbestimmung erfolgte für C_μ analog zum Vorgehen in Kapitel 2.5.2. Der Parameter $C_{\epsilon 2}$ wurde an Strömungen mit abklingender Gitterturbulenz angepasst. $C_{\epsilon 1}$ wiederum an eine Kármán-Konstante von 0.42. Für die Bestimmung der turbulenten Prandtlzahlen σ_k und σ_ϵ durch numerische Optimierung, wird wiederum auf den Artikel von Jones und Launder aus dem Jahre 1972 verwiesen. In Letzterem ist zu lesen, dass für die turbulenten Prandtlzahlen Werte nahe Eins sinnvoll

2.5 k- ϵ -Modell

sind. Auf die exakte Bestimmung wird nicht eingegangen. Der geläufige Standard-Parametersatz wurde von Launder und Sharma (1974) entwickelt. Diese nahmen eine Parameteranpassung ($C_{\epsilon 1}, C_{\epsilon 2}$) für eine wirbelnde Strömung, ausgelöst durch eine rotierende Platte, vor. Der dabei erhaltene Parametersatz eignet sich ebenfalls für die ursprünglich zur Entwicklung herangezogenen Grenzschichtströmungen und kann somit universeller eingesetzt werden.

Tabelle 2.3 Entwicklung der k- ϵ -Modellparameter.

C_μ	$C_{\varepsilon 1}$	$C_{\varepsilon 2}$	σ_k	σ_ε	Quelle:	
	0.09	2.00	1.55	1.00	1.30	(Jones & Launder, 1972)
	0.09	2.00	1.45	1.00	1.30	(Jones & Launder, 1973)
	0.09	1.92	1.44	1.00	1.30	(Launder & Sharma, 1974)
(realizable)	1.37	1.89	1.02	1.14		(Souza et al., 2017)

Derzeit ist kein anderer Parametersatz ähnlich geläufig wie der von Launder und Sharma (1974) empfohlene. Dies lässt vermuten, dass kein signifikant besserer, universaler Parametersatz existiert. Von Souza et al. (2017) wurde eine Parameteroptimierung für das realizable k- ϵ -Turbulenzmodell (siehe Kapitel 2.5.3.4), anhand mehrerer unterschiedlicher Benchmark-Strömungen, durchgeführt. Der von ihnen ermittelte, universelle Parametersatz liegt, wie in Tabelle 2.3 ersichtlich, nahe an dem Standardparametersatz von Launder und Sharma (1974).

In Hinblick auf Kapitel 2.5.2 und insbesondere 2.5.2.2 erscheint eine Anpassung an konkrete Strömungsprobleme jedoch sinnvoll. So ist eine Reduktion von $C_{\epsilon 2}$ auf 1.77 bei homogen abklingender Gitterturbulenz zu empfehlen. Bei Scherströmungen mit homogener Turbulenz, wird durch die Standardparameter das konstante Verhältnis \mathcal{P}/ϵ mit 2.1 abgeschätzt. Dieser Wert liegt signifikant über den gemessenen Werten 1.7-1.8 für \mathcal{P}/ϵ . Zudem wurden die Standardwerte der Parameter σ_ϵ , $C_{\epsilon 1}$ und $C_{\epsilon 2}$ auf eine Kármán-Konstante von 0.43 angepasst. Die Grenzen der Wirksamkeit einer Optimierung durch Parameteranpassung sollten jedoch unbedingt erhoben werden.

3 Theorie - Optimierung

Ziel einer Optimierung in mathematischem Sinne ist die Minimierung der Zielfunktion $f(\mathbf{x})$ mit den Optimierungsvariablen x_1, \dots, x_n .

$$\min_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}) \quad 2.5.1$$

Um das obige Ziel zu erreichen, wurden im Rahmen dieser Masterarbeit die in den folgenden zwei Kapiteln (3.1 und 3.2) beschriebenen Konzepte angewandt.

3.1 Lokale Optimierung

Notwendige und hinreichende Bedingungen für ein striktes lokales Minimum \mathbf{x}^* sind (Kelley, 1999):

- i. $f: U \subset \mathbb{R}^n \rightarrow \mathbb{R}$ ist zweimal stetig differenzierbar in $U \subset \mathbb{R}^n$ und $\mathbf{x}^* \in U$
- ii. \mathbf{x}^* ist ein stationärer Punkt: $\nabla f(\mathbf{x}^*) = 0$ (Extremum) 3.1.1
- iii. die Hesse-Matrix ist positiv definit: $\nabla^2 f(\mathbf{x}^*) > 0$ (Minimum) 3.1.2

3.1.1 Newtonverfahren

Das Newtonverfahren ist ein iteratives Verfahren zur Lösung nichtlinearer Gleichungssysteme $\mathbf{g}(\mathbf{x}) = \mathbf{0}$. Es approximiert $\mathbf{g}(\mathbf{x})$ linear durch eine Tangente am Iterationspunkt \mathbf{x}_k und verwendet die Nullstelle der Tangente $\mathbf{x}_k + \mathbf{s}_k$ als verbesserte Schätzung der Nullstelle von $\mathbf{g}(\mathbf{x})$. Der nächste Iterationspunkt wird demnach wie folgt bestimmt:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k = \mathbf{x}_k - \nabla \mathbf{g}(\mathbf{x}_k)^{-1} \mathbf{g}(\mathbf{x}_k) \quad 3.1.3$$

3.1.2 Gedämpftes Newtonverfahren

Das Newtonverfahren kann auf ein Optimierungsproblem übertragen werden. Hierfür wird die Stationaritätsbedingung (3.1.1) als zu lösendes Gleichungssystem $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ angesehen. Beim gedämpften Newtonverfahren handelt es sich daher um eine quadratische Approximation.

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k = \mathbf{x}_k - \alpha_k \nabla^2 f(\mathbf{x}_k)^{-1} \mathbf{g}(\mathbf{x}_k) = \mathbf{x}_k - \alpha_k \mathbf{d}_k \quad 3.1.4$$

Der Dämpfungsparameter α_k gibt die Schrittweite in Suchrichtung \mathbf{d}_k vor. Er wird in jedem Iterationsschritt durch ein Liniensuchverfahren bestimmt und soll Divergenz $f(\mathbf{x}_{k+1}) > f(\mathbf{x}_k)$ unterbinden.

3.1 Lokale Optimierung

Die Abstiegsbedingung $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$ fordert, dass Suchrichtung \mathbf{d}_k und Gradient \mathbf{g}_k näherungsweise in entgegengesetzte Richtung zeigen. Das Skalarprodukt zwischen den beiden Vektoren muss insofern negativ sein.

$$\mathbf{d}_k^T \mathbf{g}_k = -\mathbf{g}_k^T \nabla^2 f(\mathbf{x}_k)^{-1} \mathbf{g}_k < 0 \quad 3.1.5$$

Die Abstiegsbedingung ist daher nur dann erfüllt, wenn die Hesse-Matrix positiv definit ist. Die positive Definitheit ist in den meisten Anwendungen nahe des Minimums gegeben und gewährleistet die zur Lösung von Gleichung 3.1.4 notwendige Invertierbarkeit.

Das Newtonverfahren weist hohe Konvergenzgeschwindigkeiten bei Startwerten nahe des Minimums auf. Die Berechnung der per Definition symmetrischen Hesse-Matrix ist jedoch bei einer großen Anzahl an Optimierungsvariablen aufwendig (Papageorgiu, 2015).

3.1.3 Quasi-Newtonverfahren

Um den Rechenaufwand von Newtonverfahren zu reduzieren und gleichzeitig die hohe Konvergenzgeschwindigkeit beizubehalten, wird bei Quasi-Newtonverfahren die invertierte Hesse-Matrix in 3.1.4 mit einer Matrix \mathbf{H}_k iterativ approximiert. Die Suchrichtung wird dann nach folgendem Schema bestimmt:

$$\mathbf{d}_k = -\mathbf{H}_k \mathbf{g}_k \text{ mit } \mathbf{H}_0 = \mathbf{I} \leftrightarrow \mathbf{s}_{k,Newton} = -\nabla^2 f(\mathbf{x}_k)^{-1} \mathbf{g}_k \quad 3.1.6$$

Die Approximation der Hessematrix für den nächsten Iterationsschritt beruht auf der Quasi-Newton-Gleichung, die einer Sekanten Gleichung entspricht:

$$\mathbf{H}_{k+1} = \frac{\nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)}{\mathbf{x}_{k+1} - \mathbf{x}_k} = \frac{\mathbf{y}_k}{\mathbf{s}_k} \quad 3.1.7$$

3.1.3.1 BFGS

Um den oben diskutierten Forderungen nach positiver Definitheit und Symmetrie zu genügen, erfolgt die iterative Bestimmung durch symmetrische Rang-2-Aufdatierung, die zudem Gleichung 3.1.7 genügt (Biegler, 2010):

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{s}_k^T \mathbf{y}_k} - \frac{\mathbf{H}_k \mathbf{s}_k \mathbf{s}_k^T \mathbf{H}_k}{\mathbf{s}_k^T \mathbf{H}_k \mathbf{s}_k} \quad 3.1.8$$

Diese Aufdatierung wurde von Broyden (1970), Fletcher (1970), Goldfarb (1970) und Shanno (1970) entwickelt und wird *BFGS-Formel* genannt. Darauf aufbauende *BFGS-Verfahren* haben sich in der Praxis als effiziente, schnell konvergierende und stabile Quasi-Newton-Verfahren bewährt (Ulbrich, 2012).

3.1 Lokale Optimierung

3.1.3.2 L-BFGS

Die BFGS-Aufdatierung kann auch wie folgt angeschrieben werden:

$$\mathbf{H}_{k+1} = \mathbf{H}_k + [\mathbf{v}_k \mathbf{v}_k^T - \mathbf{w}_k \mathbf{w}_k^T] \quad 3.1.9$$

Um den benötigten Speicherplatz zu reduzieren, kann für eine große Anzahl an Optimierungsvariablen $n > 1000$ die Anzahl der BFGS-Aufdatierungen zur Approximation der Hessematrix limitiert werden *L-BFGS*:

$$\mathbf{H}_{k+1} = \mathbf{H}_0 + \sum_{i=\max(0, k-m+1)}^k [\mathbf{v}_i \mathbf{v}_i^T - \mathbf{w}_i \mathbf{w}_i^T] \quad 3.1.10$$

Diese Limitierung reduziert den Rechenaufwand für große n erheblich, geht jedoch mit einer geringeren Konvergenzrate einher (Biegler, 2010).

3.1.3.3 L-BFGS-B

Die Möglichkeit zur Vorgabe von Boundaries wurde für den L-BFGS-Algorithmus im *L-BFGS-B* Verfahren implementiert. Die Vorgabe der Grenzen erfolgt dann in nachstehender Form:

$$\mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \quad 3.1.11$$

Mit den Vektoren für die Unter- und Obergrenze (engl. lower bound \mathbf{l} , upper bound \mathbf{u}) der Optimierungsvariablen \mathbf{x} .

Der L-BFGS-B Algorithmus läuft hierfür auf folgende Weise ab (Zhu, Byrd, Lu, & Nocedal, 1997):

- i. Aufdatierung der Hesse-Matrix durch L-BFGS
- ii. Erstellung eines quadratischen Modells für die Zielfunktion f
- iii. Bestimmung der Suchrichtung:
 - a. Ermittlung der freien Parameter ($l_i \leq x_i \leq u_i$) durch eine Gradientenmethode
 - b. Festhalten der unfreien Parameter ($l_i > x_i \vee u_i < x_i$) auf ihren Grenzwerten (active bounds)
 - c. Bestimmung der Suchrichtung mit den freien Parametern
- iv. Liniensuche

3.2 Globalisierungs Strategien

Lokale Optimierer neigen stark dazu aufgrund ihrer Konzeption ihre Suchrichtung hin zum nächsten lokalen Minimum auszurichten. Liegen mehrere unterschiedlich stark ausgeprägte Minima vor, ist das Ergebnis bzw. der Erfolg der Optimierung vom Startpunkt abhängig. Lokale Optimierer sind daher nicht für die Findung globaler Minima geeignet. Durch Globalisierungsstrategien können geeigneten Startpunkte in Nähe des globalen Minimums eruiert werden. Dabei wird die Zielfunktion an gleichmäßig verteilten Punkten, in dem durch die Optimierungsvariablen aufgespannten Raum, ermittelt.

3.2.1 Hammersley-Verteilung

Koordinaten für gleichmäßig verteilte Punkte können durch Folgen mit niedriger Diskrepanz D_N generiert werden.

$$D_N = \sup_{a \leq c \leq d \leq b} \left| \frac{|\{s_1, \dots, s_N\} \cap [c, d]|}{N} - \frac{d-c}{b-a} \right| \quad 3.2.1$$

Diese ist Null, wenn der Anteil an Folgegliedern in einem Bereich $[c, d]$ an der Gesamtanzahl der Folgeglieder N , den Anteil des Bereichs $[c, d]$ am Gesamtbereich entspricht $[a, b]$.

Eine niedrig Diskrepanz Folge ist hierbei die Hammersley-Folge (Diwekar & Kalagnanam, 1997). Sie basiert auf der Verwendung von Primzahlen. Die Punkte werden dabei wie folgt berechnet (Wong, Luk, & Heng, 1997):

Jede natürliche Zahl k lässt sich durch die untenstehende Reihe unter Verwendung einer Primzahl p darstellen.

$$k = a_0 + a_1 p + a_2 p^2 + \dots + a_r p^r \quad 3.2.2$$

Die Koeffizienten a_i sind dabei natürliche Zahlen mit Werten zwischen $[0, p - 1]$. Mit den für k gewonnenen Koeffizienten kann dann die untenstehende Funktion berechnet werden:

$$\phi_p(k) = \frac{a_0}{p} + \frac{a_1}{p^2} + \frac{a_2}{p^3} + \dots + \frac{a_r}{p^{r+1}} \quad 3.2.3$$

Für eine feste Primzahl p und fortlaufende k , liefert sie die einzelnen Glieder einer Hammersley-Folge. In einem von d Optimierungsvariablen aufgespannten d -dimensionalen Raum bedarf es daher für die einzelnen Koordinaten $d - 1$ Folgen, gleichbedeutend mit $d - 1$ Primzahlen. Die Koordinaten des k ten Punkts lauten dann:

3.2 Globalisierungs Strategien

$$\left(\frac{k}{n}, \phi_{p_1}(k), \phi_{p_2}(k), \dots, \phi_{p_{d-1}}(k)\right) \quad 3.2.4$$

Hierbei ist n die Gesamtanzahl an Hammersley-Punkten, $p_1 < p_2 < \dots < p_{d-1}$ und $k = 0, 1, 2, \dots, n-1$.

Abbildung 3.1 zeigt die Verteilung von 100 Hammersley-Punkten in einem zweidimensionalen Raum.

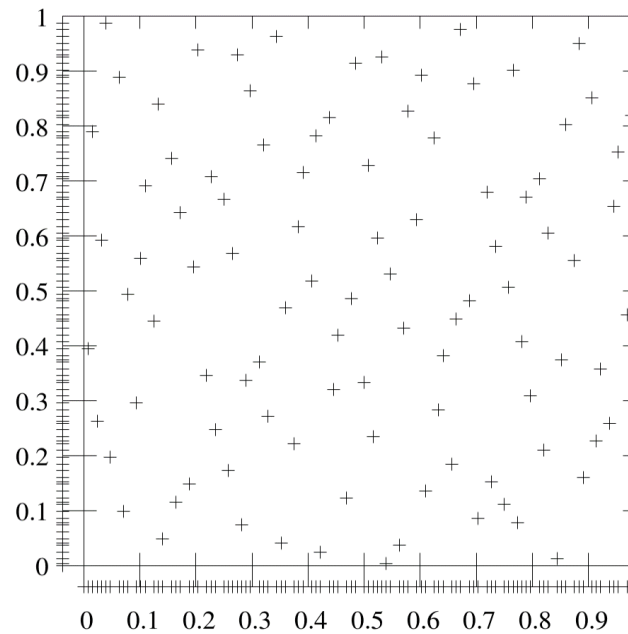


Abbildung 3.1 Hammersley-Verteilung im zweidimensionalen Raum und Projektion der Hammersley-Verteilung auf die Koordinatenachsen

Im Gegensatz zu einem orthogonalen Gitter sind die Hammersley-Punkte nicht nur auf der Fläche, sondern auch deren Projektionen auf die Koordinatenachsen (linker und unterer Seitenrand von Abbildung 3.1) gleichmäßig verteilt. Des Weiteren ist ersichtlich, dass die Hammersley-Verteilung für 100 Punkte 100 verschiedene Werte für x und y heranzieht.

4 Umsetzung

Diese Arbeit setzt sich aus drei großen Bausteinen zusammen – die Strömungssimulation, die Optimierung und als Bindeglied das Framework. Sie werden in diesem Kapitel beschrieben.

4.1 Strömungssimulation

4.1.1 Strömungssimulator - OpenFOAM

Voraussetzung für die Funktion des Frameworks ist die Bereitstellung der Simulationsergebnisse in einer für den Optimierer verwertbaren Form. Daher wurde für die Strömungssimulationen ein vom Open-Source Projekt OpenFOAM® zur Verfügung gestellter CFD-Code herangezogen. Dieser umfasst eine frei zugängliche Bibliothek an Postprocessingtools, die genutzt werden können, um die umfangreichen Ergebnisse der Strömungssimulation für den Optimierer auszuwerten.

4.1.1.1 Aufbau OpenFOAM

OpenFOAM Programme werden über die Kommandozeile aufgerufen. Die meisten Einstellungen werden dabei in Textfiles vorgenommen. Auch die Ergebnisse werden in Textfiles ausgegeben. Alle diese Files liegen in einem Ordner, dem OpenFOAM-Case-File vor und sind durch Unterordner strukturiert. Die grundlegenden Unterordner eines OpenFOAM-Case-Files, insofern einer Simulation, sind die folgenden:

system

Der Ordner system beinhaltet Berechnungseinstellungen, wie etwa die Wahl des Solvers, der Diskretisierungen sowie die Höhe der Residuen. Auch die Anzahl der maximalen Iterationsschritte wird hier festgelegt.

constant

Im Ordner constant werden Systeminformationen, wie etwa das Rechengitter, Strömungseigenschaften und Stoffeigenschaften hinterlegt.

Timedirectories

In den Timedirectories (dt. Zeitordnern) liegen die Lösungen (zum Beispiel das Feld $U(t)$) der einzelnen Zeitschritte bzw. im stationären Fall, der einzelnen Iterationsschritte vor.

4.1 Strömungssimulation

Das Timedirectory wird nach der Rechenzeit t benannt. Auch die Anfangswerte, Startwerte und Randbedingungen werden in ein Timedirectory „0“ eingetragen.

4.1.2 Strömung

Die zur Entwicklung des Frameworks herangezogene Strömung wird durch einen sphärischen Partikel ausgelöst. Dieser rollt über eine ebene Platte und verwirbelt die zuvor ruhende Atmosphäre. Außerdem weist der Partikel eine höhere Temperatur als die Platte und das Gas auf. Es kommt zu Wärmetransport durch unterschiedliche Mechanismen. Diese wurden von der DNS berücksichtigt, aber von den k - ϵ -Simulationen in dieser Arbeit vernachlässigt.

4.1.3 Rechengitter

Das Framework gleicht zur Anpassung der k - ϵ -Modellparameter die Strömungsfelder der k - ϵ -Simulation mit jenen der DNS ab. Um numerischen Fehlern beim Abgleich entgegenzuwirken, sollten die beiden Rechengitter insofern ähnlich sein. Wie in Kapitel 2.3 und 2.4 beschrieben, benötigen die beiden Berechnungsmethoden jedoch unterschiedlich hoch aufgelöste Rechengitter. Um die Reduktion des Berechnungsaufwands durch die k - ϵ -Simulation gut nutzen zu können, handelt es sich beim k - ϵ -Gitter daher um eine Vergröberung des DNS-Gitters.

Letzteres weist an der Kugel- und Plattenwand, mit einer Zellgröße von maximal $25\mu\text{m}$, eine hohe Auflösung auf. Diese ist notwendig um exakte Vorhersagen über den Wärmeübergang treffen zu können und die Kolmogorov-Skalen aufzulösen. Aus dem zweit genannten Grund, wurde außerdem ein Zellrefinement des DNS-Gitters, kurz vor der Kugel und in der ersten Hälfte des Nachlaufs, vorgenommen.

Für das Rechengitter der k - ϵ -Simulation (dargestellt in Abbildung 4.1) wurde, im Gegensatz zur hohen Auflösung an der Kugel- und Plattenwand, das Zellrefinement nicht beibehalten. Durch die Vergröberung des Rechengitters konnte die Anzahl der Zellen von ca. 6 auf 1.6-Million reduziert werden. Beide Gitter wurden mit dem OpenFoam-Tool SnappyHexMesh generiert.

4.1 Strömungssimulation

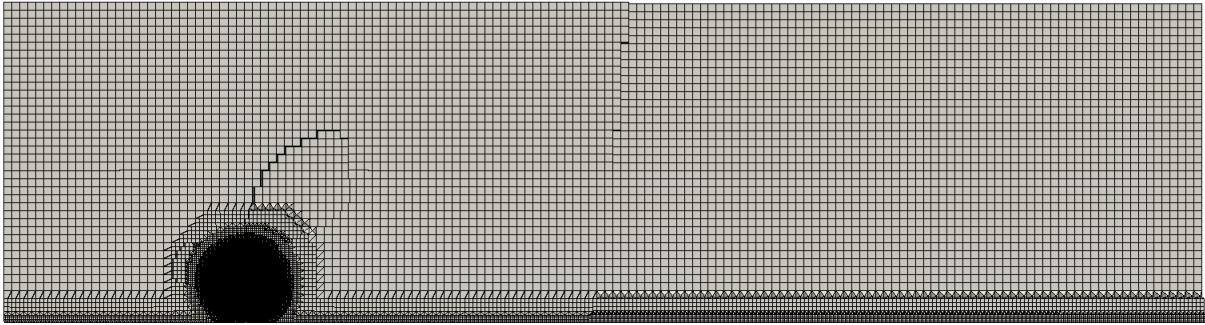


Abbildung 4.1 Rechengitter der k - ϵ -Simulation

Das dreidimensionale Gitter ist 15mm lang und 4mm hoch sowie tief. Der Mittelpunkt des sphärischen Partikels liegt 3mm hinter dem Einlass und ist in der Tiefe zentral. Der Abstand des Partikels zur Platte beträgt $1\mu\text{m}$. Da die Optimierung eine Verbesserung des Strömungsfeldes der Gasphase als Ziel hat, wird weder die Platte, noch der Partikel mit einem Gitter aufgelöst.

4.1.4 k - ϵ -Simulation

Für die k - ϵ -Simulation wurde der SIMPLE-Solver (simpleFoam) herangezogen. Die vom Solver zu erreichenden Residuen betragen $1e-5$. Die Diskretisierung der Divergenz erfolgte mit Upwind-Interpolation und die des Gradienten durch lineare Interpolation.

4.1.4.1 Startbedingungen

Das gewählte Koordinatensystem befindet sich im Mittelpunkt der Kugel und bewegt sich mit diesem mit. Die Koordinate z weist nach rechts, y nach oben und x in die Zeichenebene hinein. Die für dieses Koordinatensystem verwendeten Startbedingungen werden in nachstehender Tabelle 4.1 zusammengefasst:

4.1 Strömungssimulation

Tabelle 4.1 Startbedingungen der k-ε-Simulation

Randregion	$\langle U \rangle$	p	k	ε	ν_T
Einlass	(0, 0, 5)	grad=0	grad=0	grad=0	calculated**
Auslass	grad=0*	0	grad=0*	grad=0*	calculated**
Kugelwand	rotierend: - Ursprung: (0, 0, 0) - Achse: (1, 0, 0) - ω : - 10000	grad=0	Low-Re Wall Function	Low-Re Wall Function	Low-Re Wall Function
Plattenwand	(0, 0, 5)	grad=0	Low-Re Wall Function	Low-Re Wall Function	Low-Re Wall Function
Vorder/Hinterseite	grad=0	grad=0	grad=0	grad=0	calculated**
Oberseite	grad=0	grad=0	grad=0	grad=0	calculated**
Inneres Feld	(0, 0, 5)	1	0.5	14.85	calculated**
Anmerkung	*bei Rückfluss wird ein fixer Wert vorgegeben **kann aus den restlichen Startbedingungen berechnet werden				

4.1.4.2 Ergebnisse der k-ε-Simulation - Vergleich DNS

Die Parameteroptimierung wird durchgeführt, um verbesserte und leicht verfügbare Strömungsfelder für die Untersuchung des Wärmeübergangs zu erhalten. Die Darstellung und Diskussion der Ergebnisse beschränkt sich daher auf Felder, welche für den Wärmeübergang wesentlich sind. Das ist zum einen das Geschwindigkeitsfeld, zur Untersuchung des konvektiven Wärmetransportes und zum anderen die Felder der Turbulenzgrößen k und ε , zur Skalierung des turbulenten Wärmetransportes. Der Einfluss des Druckfeldes auf den Wärmeübergang wurde in dieser näherungsweise inkompressiblen Strömung als vernachlässigbar angesehen. Da die Energieerhaltung in der k-ε-Simulation nicht berücksichtigt wurde, liegen keine Temperaturfelder zur Anpassung vor. Die folgenden Plots zeigen die dreidimensionalen Felder von $\langle U \rangle$, k und ε auf der Symmetrieebene der Geometrie.

Abbildung 4.2 zeigt den Betrag des mittleren Geschwindigkeitsfeldes berechnet durch die k-ε-Simulation und vergleicht dieses mit dem Ergebnis der DNS.

4.1 Strömungssimulation

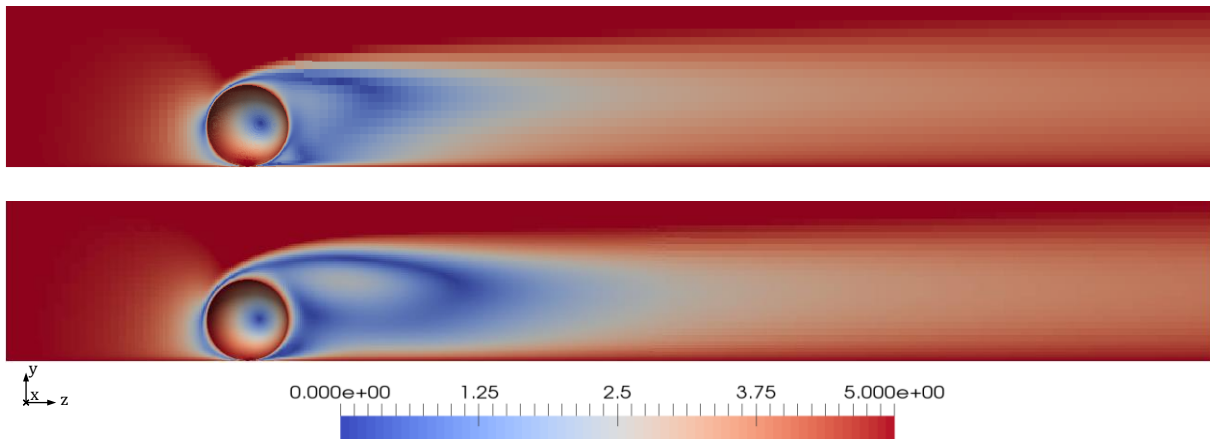


Abbildung 4.2 Betrag des gemittelten Geschwindigkeitsfeldes $\langle \mathbf{U} \rangle$ in m/s , berechnet mit dem $k-\varepsilon$ -Turbulenzmodell (oben) und DNS (unten)

An der Innenseite der Kugel ist die Rotationsbewegung des Partikels gut erkennbar. Dieser schiebt eine kleine Menge Luft vor sich her und erzeugt im Nachlauf eine Wirbelschlepe in der Größe des Partikels selbst. Die definierte Wirbelstruktur zerfällt sukzessive in z -Richtung und weicht, nach dem ersten Viertel des Nachlaufes, einer homogenen Geschwindigkeitsverteilung. Die DNS-Ergebnisse prognostizieren einen ähnlichen Verlauf. Die größten Diskrepanzen zwischen DNS und $k-\varepsilon$ -Modell treten bei der Wirbelschlepe im Nachlauf auf. Diese ist in der DNS-Berechnung etwa doppelt so lang und tendenziell breiter. Dadurch ist das homogene Geschwindigkeitsfeld erst im zweiten Drittel des Nachlaufs zu finden. Identifikation der einzelnen Strömungsgebiete in der $k-\varepsilon$ -Simulation (Staugebiet, Wirbelschlepe, homogener Nachlauf) und Vergleich der entsprechenden Regionen in der DNS-Lösung, zeigt quantitativ eine gute Übereinstimmung.

Die höchste turbulente kinetische Energie ist im oberen Bereich des Nachlaufs bzw. der Wirbelschlepe zu finden (siehe Abbildung 4.3). Dort können die hohen, über einen längeren Bereich vorliegenden Geschwindigkeitsgradienten Turbulenz erzeugen (Gleichung 2.5.27). Im Bereich des homogenen Nachlaufs nimmt diese dann stromabwärts ab. Es entsteht ein Verlauf ähnlich einer Rauchfahne.

4.1 Strömungssimulation

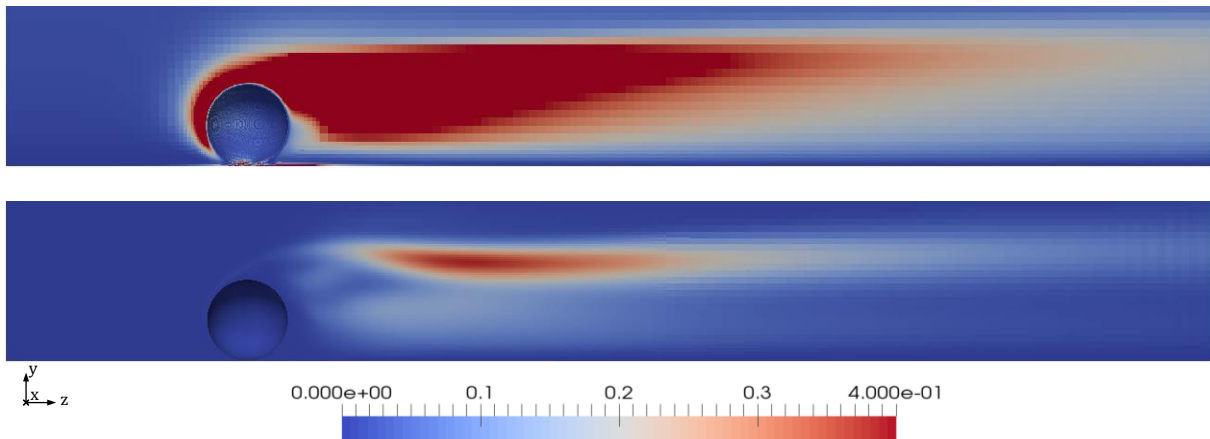


Abbildung 4.3 turbulente kinetische Energie k in m^2/s^2 , berechnet mit dem k - ε -Turbulenzmodell (oben) und DNS (unten)

Die quantitative starke Abweichung zwischen DNS und k - ε -Modell von etwa einer Größenordnung, täuscht über den qualitativ ähnlichen Verlauf hinweg (siehe Abbildung 4.4).

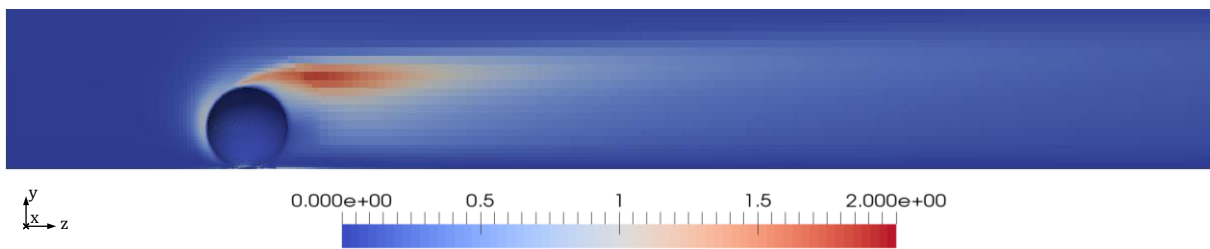


Abbildung 4.4 erhöhte Skalierung der turbulenten kinetischen Energie k in m^2/s^2 , berechnet mit dem k - ε -Turbulenzmodell

Die höchste turbulente kinetische Energie liegt beim k - ε -Modell, ebenfalls im oberen Bereich des Nachlaufes, bei den höchsten Geschwindigkeitsgradienten an der Wirbelschleppe. Diese wird vom k - ε -Modell in ihrer Länge unterschätzt. Die Fahne fällt daher kürzer aus als in der DNS-Berechnung aus. Außerdem setzt sie früher, am oberen Pol des Partikels an.

Auch die Verläufe der Dissipationsrate (Abbildung 4.5) sind qualitativ relativ ähnlich. Die höchsten Werte treten in Bereichen mit hohen Geschwindigkeitsgradienten auf, sind daher in der Nähe der Wände und an der Oberseite der Wirbelschleppe zu finden. Es bildet sich wieder eine Fahnenstruktur aus. Der Einfluss der Gradienten auf ε wird von der DNS wesentlich stärker prognostiziert. Der Wandeinfluss auf die Strömung ist daher stärker und die Fahne größer und deutlicher ausgeprägt.

4.1 Strömungssimulation

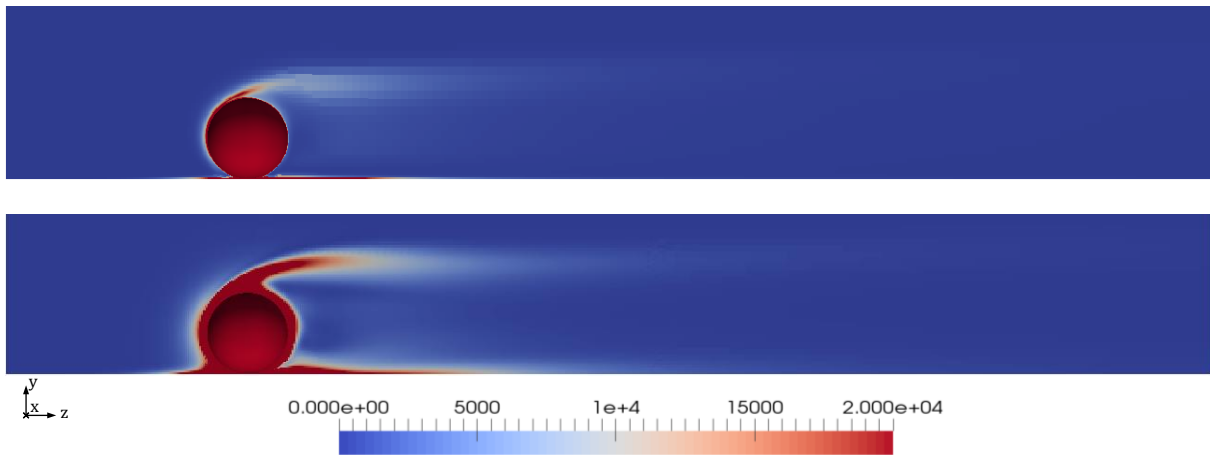


Abbildung 4.5 Dissipationsrate ε in m^2/s^3 , berechnet mit dem k- ε -Turbulenzmodell (oben) und DNS (unten)

Durch die gewählte Skalierung werden die Spitzenwerte der DNS abgeschnitten. Eine quantitative Übereinstimmung ist nicht gewährleistet.

4.1.5 Sensitivitätsanalyse

Für eine erste Abschätzung des vorliegenden Optimierungsproblems wurde zunächst der Einfluss der k- ε -Modellparameter auf die Feldgrößen $\langle \mathbf{U} \rangle$, k und ε untersucht. Hierbei erfolgte eine Variation von einem Parameter, unter Beibehaltung der Standardwerte für die restlichen Parameter.

Im Zuge dieser Arbeiten wurden daher drei Felder auf Änderung von fünf Parametern untersucht. Infolgedessen sind die erhaltenen Ergebnisse sehr umfangreich. Die im Zuge der Sensitivitätsanalyse angefertigten Plots liegen deshalb, in ihrer Gesamtheit, nur im digitalen Anhang bereit.

4.1.5.1 Parametervariation

Im Folgenden soll der Einfluss der einzelnen Modellparameter anhand der durch die Parametervariation gewonnenen Strömungsfelder erörtert werden. Viele der berechneten Felder ähneln sich stark. In den meisten dieser Fälle wird deshalb in den nachstehenden Kapiteln nur ein repräsentativer Plot dargestellt.

Parametervariation - $C_{\varepsilon 1}$

Durch Steigerung des Parameters $C_{\varepsilon 1}$ kommt es zu einem Abfall der turbulenten kinetischen Energie im gesamten Feld (siehe Abbildung 4.6). Diese ist für unterschiedliche $C_{\varepsilon 1}$ am höchsten in der Nähe des Staupunkts und am Ablösepunkt

4.1 Strömungssimulation

oberhalb der Kugel. Eine qualitative Änderung des Strömungsfeldes ist nicht zu verzeichnen. $C_{\varepsilon 1}$ scheint daher zur quantitativen Anpassung des k -Feldes geeignet zu sein.

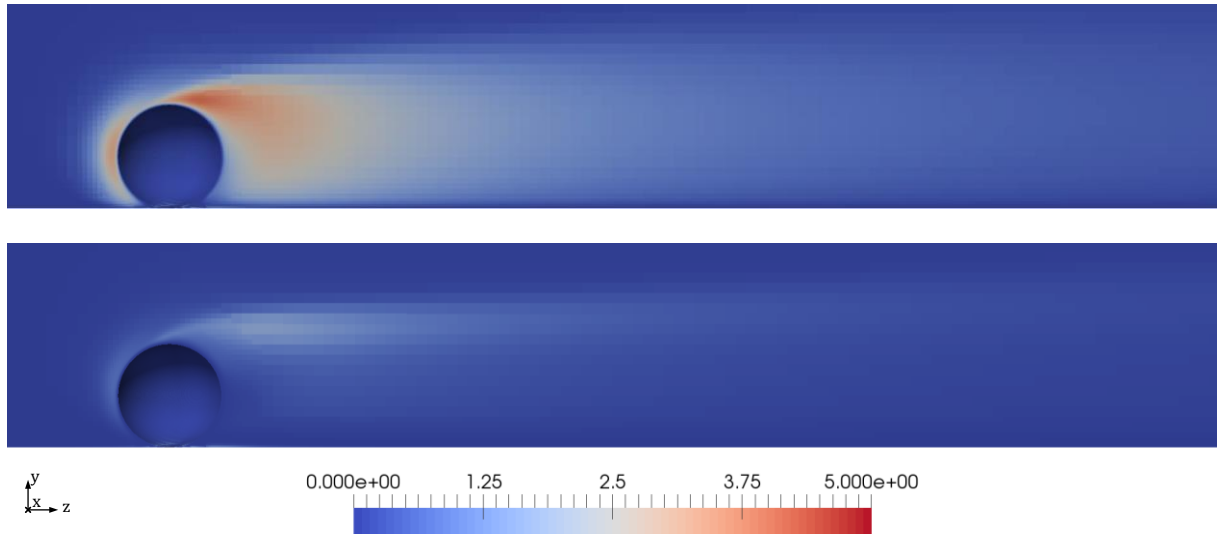


Abbildung 4.6 Einfluss von $C_{\varepsilon 1}$ auf das Feld von k in m^2/s^2 : oben $C_{\varepsilon 1} = 0.72$, unten $C_{\varepsilon 1} = 1.5$

Mit einer Steigerung des Parameters $C_{\varepsilon 1}$ geht ebenfalls ein Abfall der Dissipationsrate einher (Abbildung 4.7). Der an der Kugel angrenzende Bereich mit hohen Dissipationsraten verkleinert sich für höhere $C_{\varepsilon 1}$. Des Weiteren verschiebt sich der relativ breite Bereich mit hohen Dissipationsraten am Ablösepunkt mit steigendem $C_{\varepsilon 1}$ nach vorne und geht schlussendlich in eine dünne Fahne im Nachlauf über.

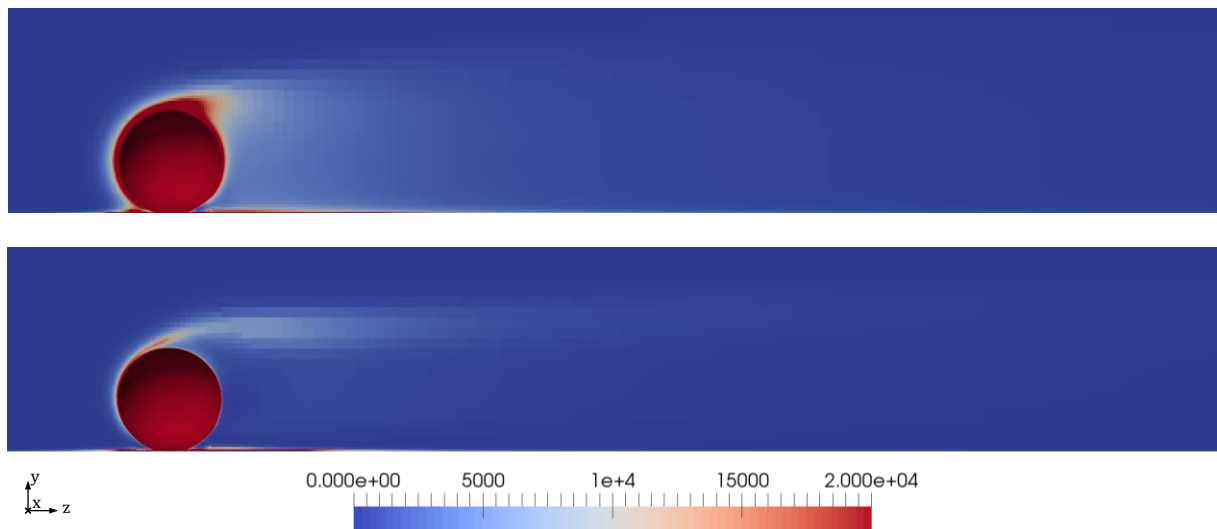


Abbildung 4.7 Einfluss von $C_{\varepsilon 1}$ auf das Feld von ε in m^2/s^3 : oben $C_{\varepsilon 1} = 0.72$, unten $C_{\varepsilon 1} = 1.5$

Es ist eine kleine qualitative Änderung des Feldes zu verzeichnen. Quantitative Änderungen begrenzen sich auf relativ wenige Zellen in Kugelnähe.

Im Hinblick auf die obigen zwei Abbildungen kann zusammengefasst werden, dass sowohl k als auch ε mit steigendem $C_{\varepsilon 1}$ fallen. Ähnliches Verhalten weist auch die Benchmark-

4.1 Strömungssimulation

Strömung in Kapitel 2.5.2.2 auf (Scherströmung mit homogener Turbulenz, siehe Gleichungen 2.5.54 und 2.5.55), die zur Abschätzung bzw. Bestimmung von $C_{\varepsilon 1}$ herangezogen worden ist. Eine Voraussetzung hierfür ist das in der Benchmark-Strömung gegebene Verhältnis $\mathcal{P}/\varepsilon > 1$.

Nach Abbildung 4.8 ändert sich das Geschwindigkeitsfeld qualitativ durch die Änderung von $C_{\varepsilon 1}$. Grundsätzlich kann festgestellt werden, dass an der Schattenseite der Kugel eine Wirbelschleppe anliegt, deren Größe mit steigendem $C_{\varepsilon 1}$ zunimmt. Außerdem sind für höhere $C_{\varepsilon 1}$ größere Geschwindigkeitsgradienten zu verzeichnen.

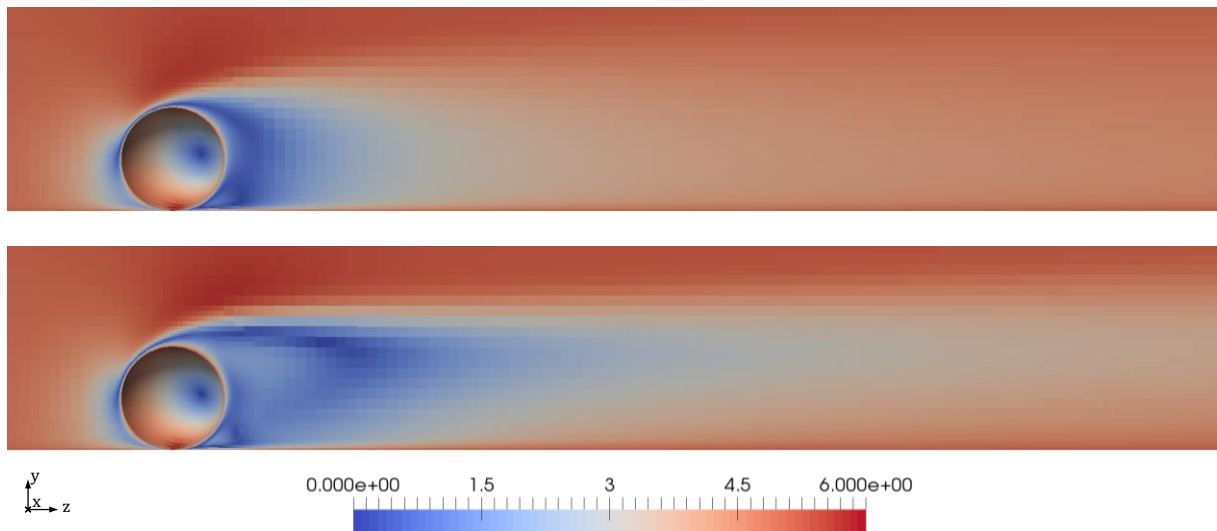


Abbildung 4.8 Einfluss von $C_{\varepsilon 1}$ auf $\langle \mathbf{U} \rangle$ in m/s : oben $C_{\varepsilon 1} = 0.72$, unten $C_{\varepsilon 1} = 1.5$

Zusammenfassend ist zu sagen, dass die qualitativen Verläufe aller durch die DNS berechneten Felder, durch höhere $C_{\varepsilon 1}$ besser wiedergegeben werden.

4.1 Strömungssimulation

Parametervariation - $C_{\varepsilon 2}$

Mit einer Variation von $C_{\varepsilon 2}$ können sowohl qualitativ als auch quantitativ ähnliche Ergebnisse wie mit einer Variation von $C_{\varepsilon 1}$ erzielt werden. Im Unterschied zur $C_{\varepsilon 1}$ -Variation steigt k jedoch im Allgemeinen bei Erhöhung von $C_{\varepsilon 2}$ (Abbildung 4.9).

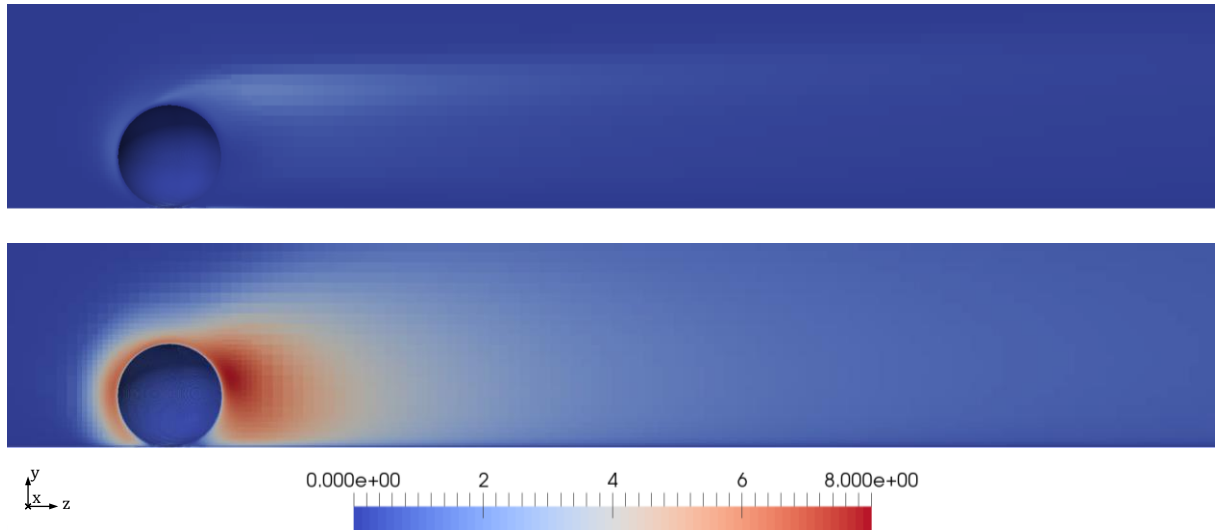


Abbildung 4.9 Einfluss von $C_{\varepsilon 2}$ auf das Feld von k in m^2/s^2 : oben $C_{\varepsilon 2} = 1.87$, unten $C_{\varepsilon 2} = 19.2$

Auch das ε -Feld nimmt mit Zunahme von $C_{\varepsilon 2}$ höhere Werte an (Abbildung 4.10). Dieses grundsätzliche Verhalten der k - und ε -Felder weisen auch die Benchmark-Strömungen in Kapitel 2.5.2.2 – homogen abklingende Turbulenz (Gleichung 2.5.48 mit 2.5.49) bzw. Scherströmung mit homogener Turbulenz (Gleichungen 2.5.54 und 2.5.55) – auf, die zur Ermittlung von $C_{\varepsilon 1}$ und $C_{\varepsilon 2}$ herangezogen wurden.

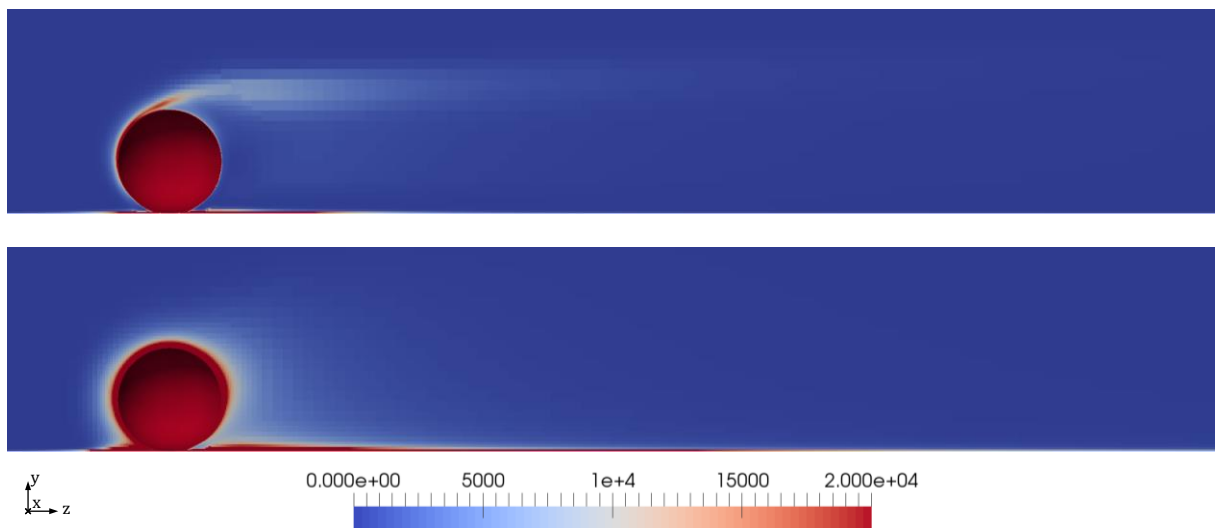


Abbildung 4.10 Einfluss von $C_{\varepsilon 1}$ auf das Feld von ε in m^2/s^3 : oben $C_{\varepsilon 1} = 1.87$, unten $C_{\varepsilon 1} = 19.2$

4.1 Strömungssimulation

Im Gegensatz zur Variation von $C_{\varepsilon 1}$ geht die Fahne nicht in den relativ breiten Bereich mit hohen Dissipationsraten am Ablösepunkt über. Stattdessen wird der Bereich mit hohen Dissipationsraten gleichmäßig über die (Kugel-)Wand breiter.

Auch die durch $C_{\varepsilon 2}$ -Variation erhaltenen Geschwindigkeitsfelder (Abbildung 4.11) ähneln den Ergebnissen der $C_{\varepsilon 1}$ -Variation. Die Größe der Wirbelschleppe nimmt jedoch mit sinkendem $C_{\varepsilon 2}$ zu.

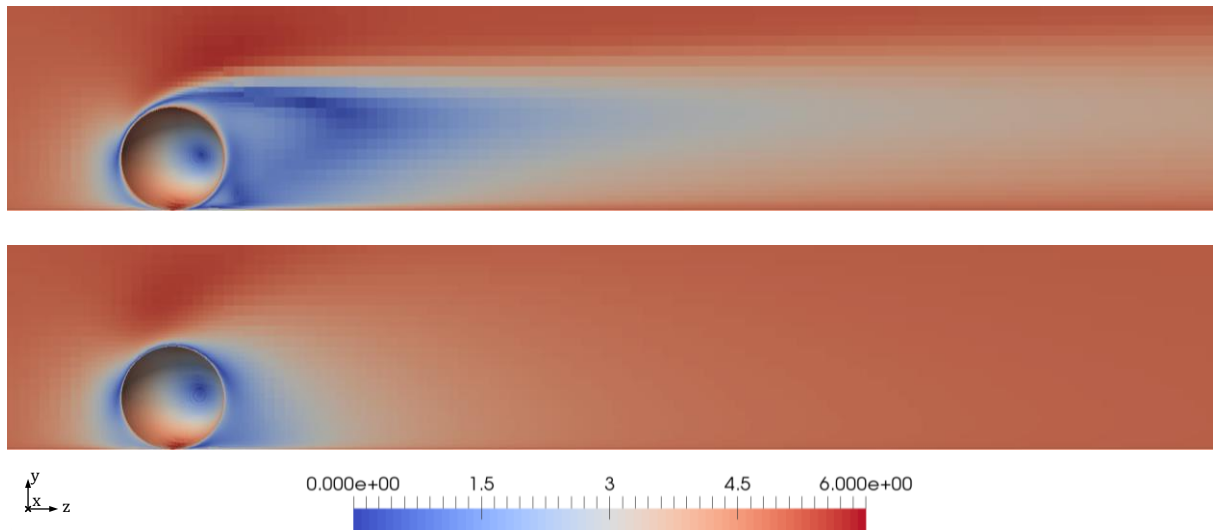


Abbildung 4.11 Einfluss von $C_{\varepsilon 1}$ auf $\langle U \rangle$ in m/s: oben $C_{\varepsilon 2} = 1.87$, unten $C_{\varepsilon 2} = 19.2$

Im Allgemeinen kann zusammengefasst werden, dass eine Senkung vom Parameter $C_{\varepsilon 2}$ zu einer Angleichung aller untersuchten Felder an die DNS-Ergebnisse führt.

Parametervariation - C_{μ}

Die durch Variation des Parameters C_{μ} erhaltenen Ergebnisse ähneln denen der $C_{\varepsilon 1}$ - und $C_{\varepsilon 2}$ -Variation. Die Wirkung einer Erhöhung von C_{μ} auf das Geschwindigkeitsfeld und das Dissipationsratenfeld kann am besten mit dem Einfluss bei Steigerung des Parameters $C_{\varepsilon 2}$ verglichen werden. Das Verhalten des Feldes k bei Senkung von C_{μ} entspricht näherungsweise dem Verhalten von k bei Erhöhung von $C_{\varepsilon 1}$. Die Felder der DNS werden daher mit niedrigeren C_{μ} besser wiedergegeben.

4.1 Strömungssimulation

Parametervariation - σ_ε

Durch eine Erhöhung von σ_ε lässt sich der Bereich mit der höchsten turbulenten kinetischen Energie vor die Kugel verlagern (Abbildung 4.12). Im Hinblick auf die DNS Ergebnisse ist dieses Verhalten jedoch nicht wünschenswert.

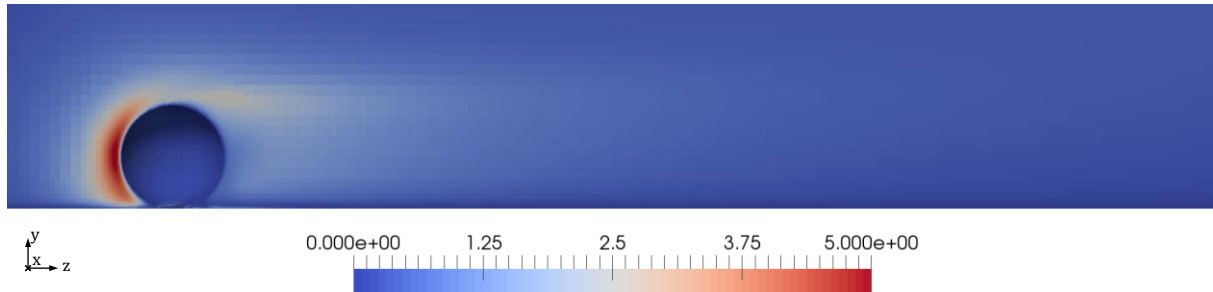


Abbildung 4.12 Einfluss von σ_ε auf das Feld von k in m^2/s^2 mit $\sigma_\varepsilon = 13$

Der Parameter σ_ε ist im Transportterm der Erhaltungsgleichung für ε enthalten. Der Transport wird durch Diffusion modelliert, σ_ε ist dabei indirekt proportional zur Diffusionskonstante. Die in Abbildung 4.13 ersichtliche Verbreiterung des Bereiches mit hohen ε kann jedoch nicht durch erhöhte Diffusion erklärt werden (die Diffusionskonstante sinkt mit steigendem σ_ε). Sie ist vielmehr durch die Änderung des Geschwindigkeitsfeldes verursacht, denn Geschwindigkeitsgradienten sind die Quellen der Dissipationsrate (siehe Gleichung 2.5.44 und 2.5.27).

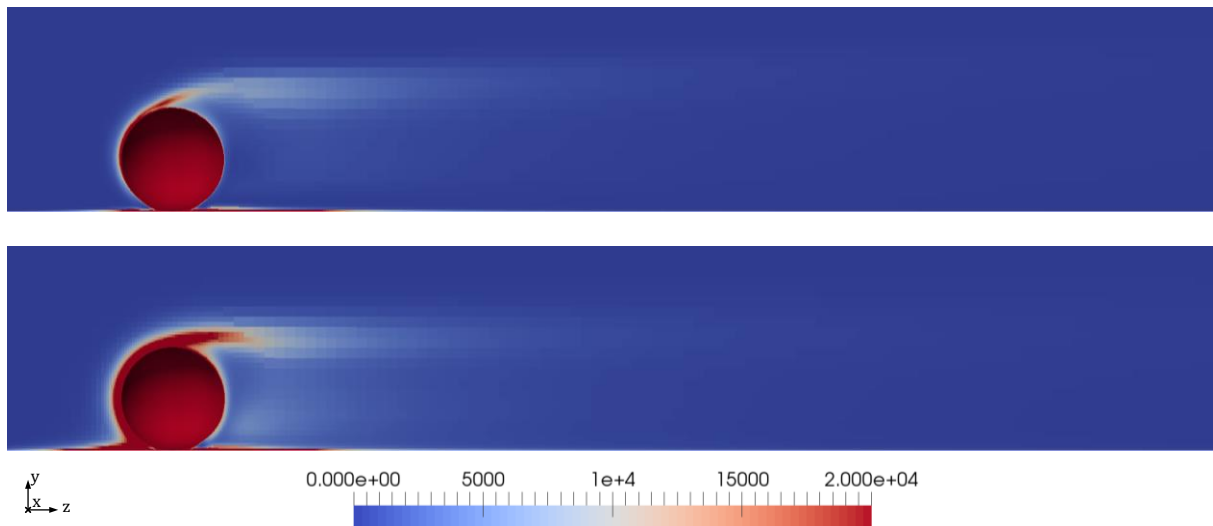


Abbildung 4.13 Einfluss von σ_ε auf das Feld von ε in m^2/s^3 : oben $\sigma_\varepsilon = 1.30$ (Standard), unten $\sigma_\varepsilon = 13.00$

Der Einfluss von σ_ε auf die Form des Nachlaufes und auf das Geschwindigkeitsfeld im Allgemeinen, ist geringer als bei den Parametern $C_{\varepsilon 1}$, $C_{\varepsilon 2}$ und C_μ . Die Geschwindigkeitsgradienten nehmen mit Steigerung von σ_ε zu, das Strömungsprofil wird insgesamt schärfer (Abbildung 4.14). Die leichte Umformung des Nachlaufes und die Steigerung der Gradienten erklärt die Änderung des Dissipationsratenfeldes besser.

4.1 Strömungssimulation

Die theoretisch zu erwartende Änderung des diffusiven Transportes konnte anhand der Ergebnisse nicht nachvollzogen werden.

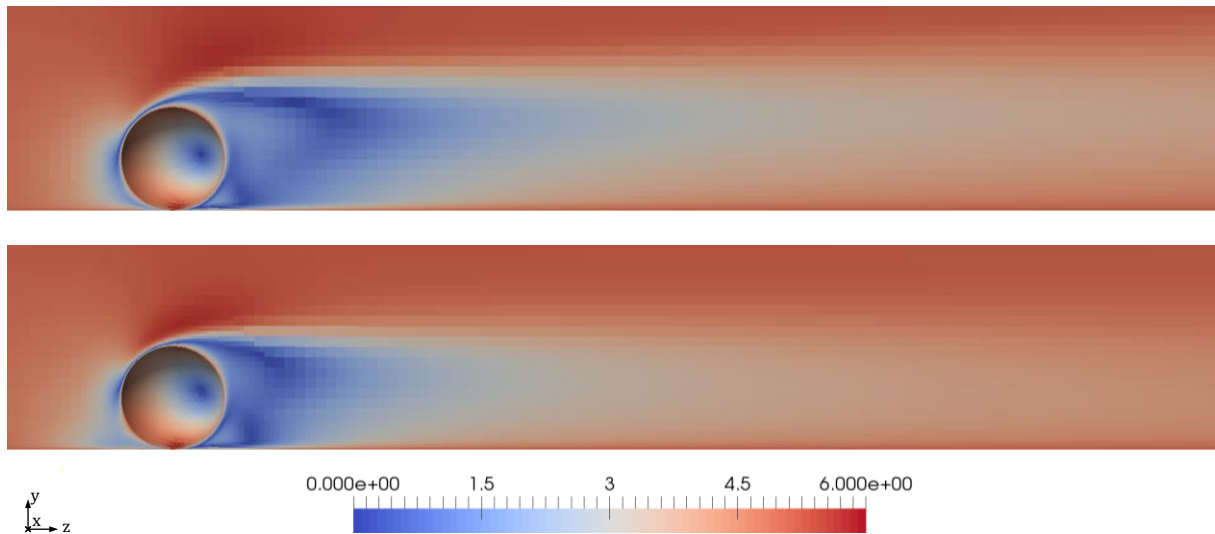


Abbildung 4.14 Einfluss von σ_ε auf $\langle \mathbf{U} \rangle$ in m/s: oben $\sigma_\varepsilon = 1.30$ (Standard), unten $\sigma_\varepsilon = 13.00$

Parametervariation - σ_k

Auch ein Einfluss von σ_k auf den diffusiven Transport von k konnte in den vorliegenden Ergebnissen nicht abgeleitet werden. Im Allgemeinen ist der Einfluss von σ_k auf die untersuchten Felder sehr gering. Bei Erhöhung von σ_k um das Zehnfache sind die gewonnenen Ergebnisse auf den ersten Blick nicht von den Ergebnissen der Standardparameter zu unterscheiden. Von einer Darstellung soll daher abgesehen werden.

Parametervariation - Instabilität der Lösung

Eine Konvergenz der Lösung ist nicht für alle beliebigen Parametersätze gewährleistet. Es konnten drei mögliche Berechnungsausgänge beobachtet werden. Abbildung 4.15 zeigt diese exemplarisch anhand der Residuenverläufe von drei unterschiedlichen Parametersätzen.

4.1 Strömungssimulation

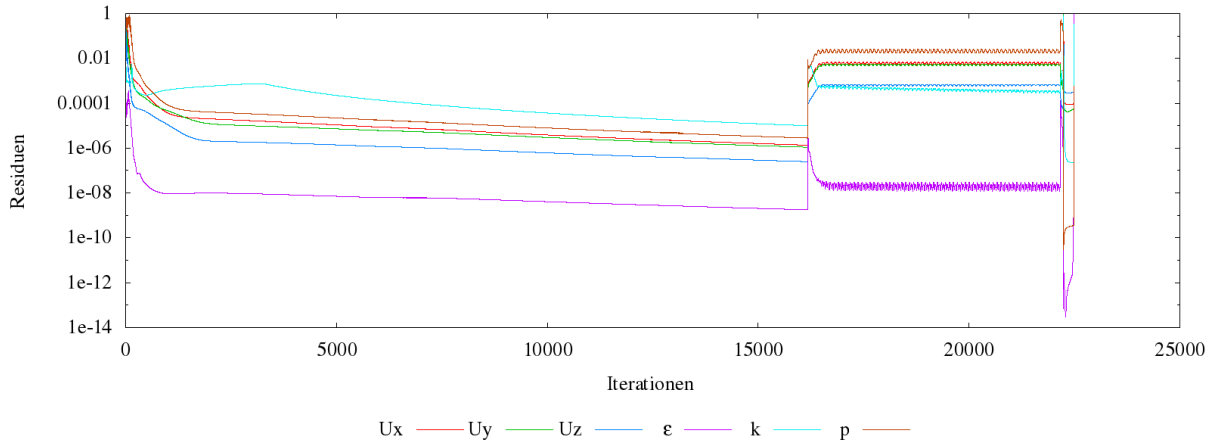


Abbildung 4.15 Die drei möglichen Berechnungsausgänge, von links nach rechts: konvergiert, nicht konvergiert und abgestürzt

Der erste Parametersatz konvergiert und erreicht nach etwa 16000 Iterationen mit allen Residuen die Toleranz von $1E-5$. Der zweite Parametersatz konvergiert nicht. Die Residuen der meisten Felder beginnen um einen konstanten Wert zu oszillieren. Nach 6000 Iterationen wird die Berechnung von OpenFOAM beendet. Beim dritten Parametersatz divergieren die Residuen, der SIMPLE-Solver von OpenFOAM stürzt ab.

Die Stabilitätsgrenzen einzelner Parameter hängen von den gewählten Werten der anderen Parameter ab. Sie sollen in Kapitel 5.1 anwendungsorientiert diskutiert werden. Werte kleiner oder gleich Null sind in jedem Fall instabil.

4.1.5.2 Sensitivitäten

Die durch die Einzelparametervariation gewonnenen Felder wurden zur Berechnung von Sensitivitäten herangezogen. Die Sensitivitäten bezeichnen in diesem konkreten Fall die partiellen Ableitungen der Feldgrößen nach den Modellparametern. Die Sensitivität eines beliebigen Feldes ϕ bezüglich eines Parameters P_i lautet:

$$S_i(\mathbf{x}, P_i, \Delta P_i) = \frac{\phi(\mathbf{x}, P_i + \Delta P_i) - \phi(\mathbf{x}, P_i)}{\Delta P_i} \quad 4.1.1$$

Sie ist für die stationären Lösungen des k- ϵ -Modells eine Funktion des Ortes. Ihre Berechnung wurde durch das OpenFOAM-Post-Processing-Tool *Sensitivity* bewerkstelligt. Dieses wurde im Zuge dieser Arbeit auf Basis des in OpenFOAM 2.4.0 mitgelieferten Post-Processing-Tool *Mach* entwickelt. Die hier dargestellten Sensitivitäten wurden an den Standardwerten $P_i = P_{i,standard}$ mit unterschiedlichen Schrittweiten ΔP_i und Schrittrichtungen $\pm \Delta P_i$ gebildet. Auch die Sensitivitätenfelder ähneln sich in vielen Fällen. Hier daher nur ein Auszug der repräsentativsten und aufschlussreichsten Plots.

4.1 Strömungssimulation

Sensitivitäten - k

Wie in Kapitel 4.1.5.1 beschrieben, steigt k , im gesamten vom Partikel beeinflussten Strömungsfeld, mit fallendem $C_{\varepsilon 1}$ bzw. mit höher werdendem $C_{\varepsilon 2}$. Die Sensitivitätenfelder dieser zwei Parameter sind dementsprechend einfach aufgebaut (siehe Abbildung 4.16) und weisen nur ein Vorzeichen, unabhängig von der Schrittweite und Schrittrichtung auf. $C_{\varepsilon 1}$ und $C_{\varepsilon 2}$ sind dementsprechend für eine quantitative Anpassung von k geeignet. Das einfache Verhalten ist auch für einen Optimierer gut nutzbar.

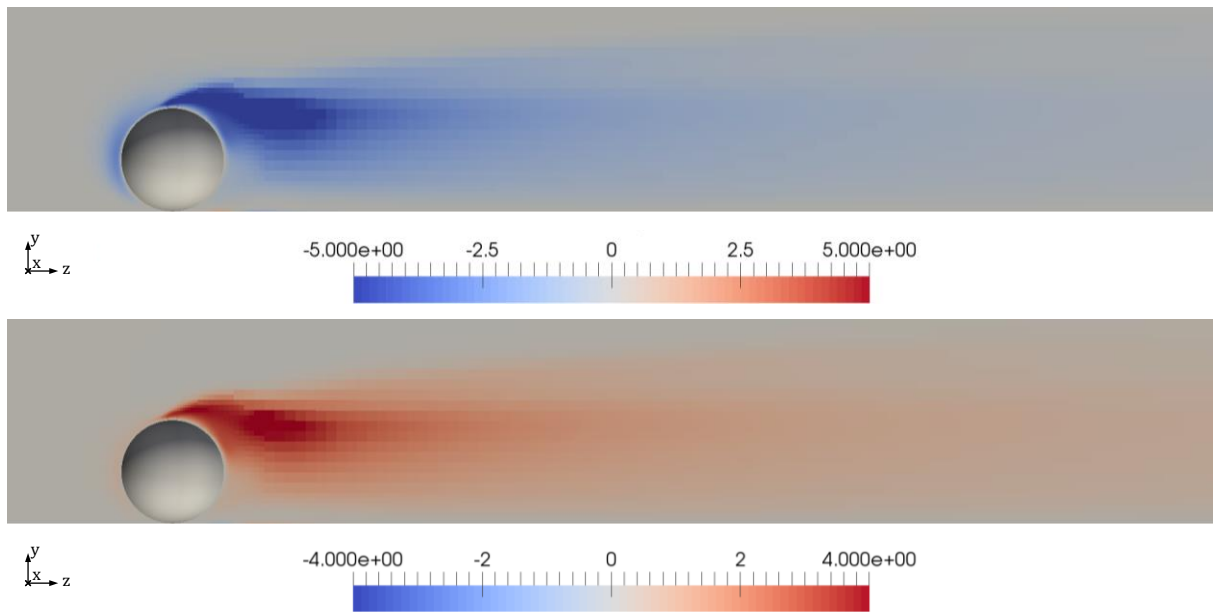


Abbildung 4.16 k - $C_{\varepsilon i}$ -Sensitivitätenfelder in m^2/s^2 : oben bzgl. $C_{\varepsilon 1} = 1.44$ mit $\Delta C_{\varepsilon 1} = 0.03$, unten bzgl. $C_{\varepsilon 2} = 1.92$ mit $\Delta C_{\varepsilon 2} = -0.05$

Auch die k -Sensitivitätenfelder des Parameters σ_k (Abbildung 4.17) sind ähnlich einfach aufgebaut. Ihr Vorzeichen wechselt aber mit der Schrittrichtung. Werden die Sensitivitäten mit Werten unterhalb des Standardwertes gebildet, liegen diese negativ vor. Bei Bildung der Sensitivitäten mit Werten oberhalb des Standardwerts sind diese positiv. In der Nähe des Standardwertes für σ_k liegt also ein Minimum für k vor. Abschließend ist zu sagen, dass die k -Sensitivitäten von σ_k , mit einer Größenordnung von $E-3$, die kleinsten aller Parameter sind.

4.1 Strömungssimulation

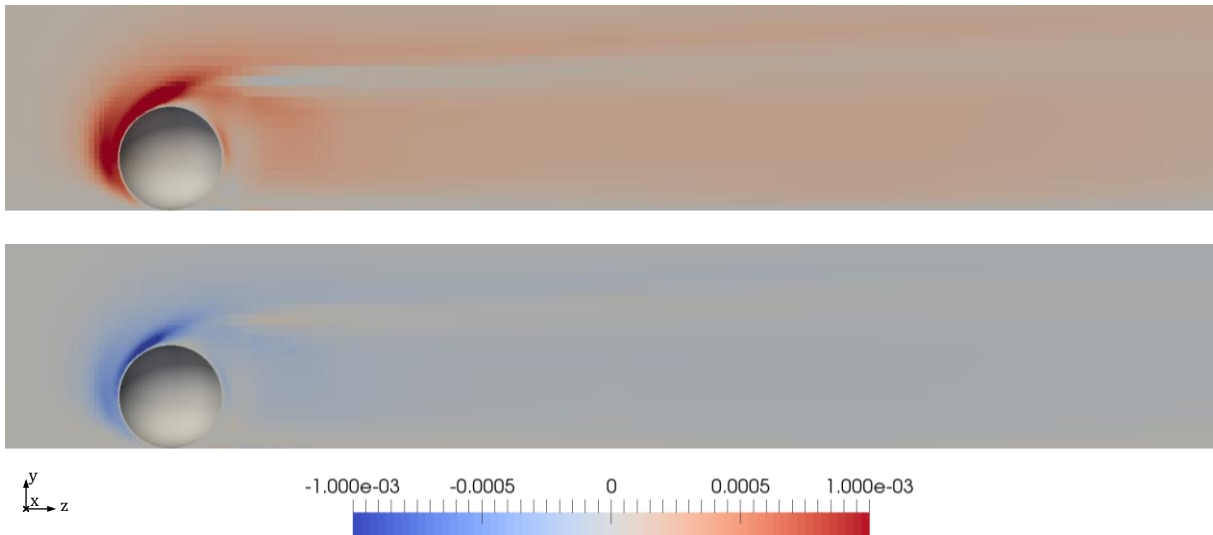


Abbildung 4.17 k - σ_k -Sensitivitätenfelder in m^2/s^2 bzgl. $\sigma_k = 1.00$: oben $\Delta \sigma_k = -0.5$, unten $\Delta \sigma_k = 1.5$

Die Verläufe der Sensitivitätenfelder von C_μ (Abbildung 4.18) sind komplizierter. Durch einen Anstieg von C_μ kommt es zu einem Anstieg von k in Bereichen vor der Kugel und im unteren Teil des Nachlaufes. Im oberen Bereich des Nachlaufes sinkt k mit höher werdenden C_μ ab. Dieses Verhalten ist näherungsweise unabhängig von Schrittweite und Schrittrichtung. C_μ eignet sich also für eine qualitative Anpassung von k .

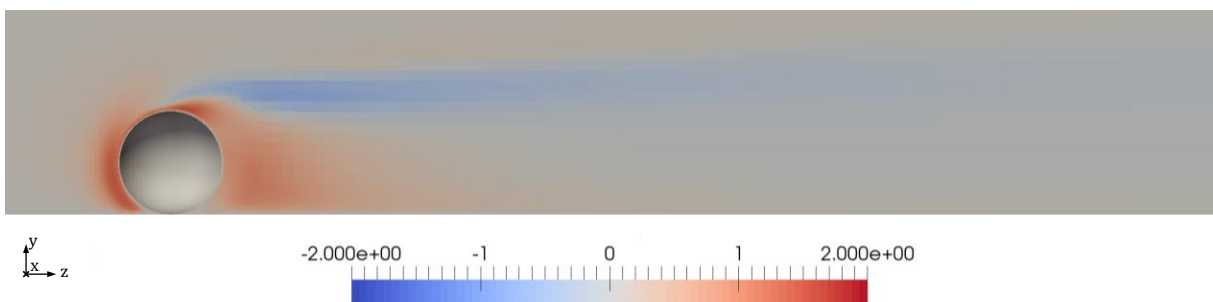


Abbildung 4.18 k - C_μ -Sensitivitätenfeld in m^2/s^2 bzgl. $C_\mu = 0.09$ mit $\Delta C_\mu = 0.81$

Auch der Verlauf der Sensitivitäten von σ_ε (Abbildung 4.19) weist unterschiedliche Vorzeichen auf. Die Bereiche, in welchen die einzelnen Vorzeichen liegen, verformen sich mit Änderung der Schrittrichtung. Der Einfluss von σ_ε auf k ist für einen Optimierer insofern schwerer handhabbar.

4.1 Strömungssimulation

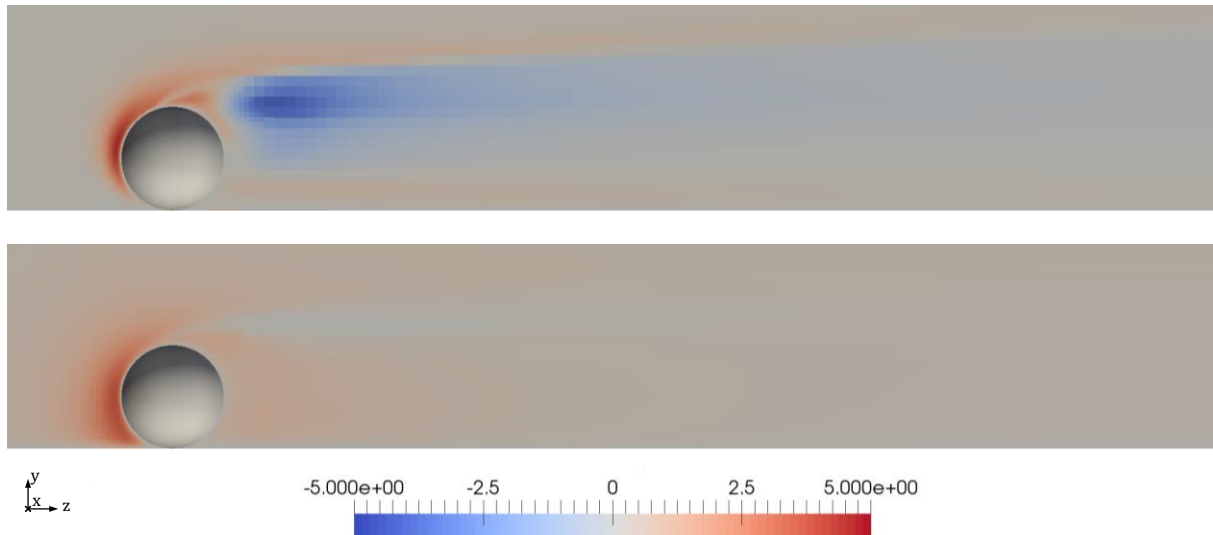


Abbildung 4.19 $k\text{-}\sigma_\varepsilon$ -Sensitivitätenfelder in m^2/s^2 bzgl. $\sigma_\varepsilon = 1.30$: oben $\Delta\sigma_\varepsilon = -0.75$, unten $\Delta\sigma_\varepsilon = 11.70$

Sensitivitäten - ε

Die Verläufe der Sensitivitäten von ε sind schwieriger zu interpretieren und weisen weitaus stärkere Gradienten auf als die Felder von k . Die Parameter beeinflussen nur kleine Bereiche der Strömung in denen hohe Geschwindigkeitsgradienten auftreten. Die mit unterschiedlichen Parametern gebildeten ε -Sensitivitätenfelder sind einheitlicher als jene von k . Sie unterscheiden sich im qualitativen Verlauf in manchen Fällen nur durch ein Vorzeichen. Ihre Werte liegen in den Größenordnungen von $1E+3$ bis $1E+4$. Der Parameter σ_k bildet eine Ausnahme. Er hat mit Sensitivitäten in der Größenordnung von $1E+1$ den geringsten Einfluss.

Die Sensitivitäten sind näherungsweise unabhängig von der Schrittweite. Des Weiteren hat in den meisten Fällen die Schrittrichtung keinen Einfluss auf die Sensitivitätenfelder. Einzig bei σ_k kommt es wieder zu einem Vorzeichenwechsel der Regionen, bei Überschreiten eines Wertes nahe des Standardwertes.

4.1 Strömungssimulation

Abbildung 4.20 zeigt zwei exemplarische ε -Sensitivitäten Verläufe:

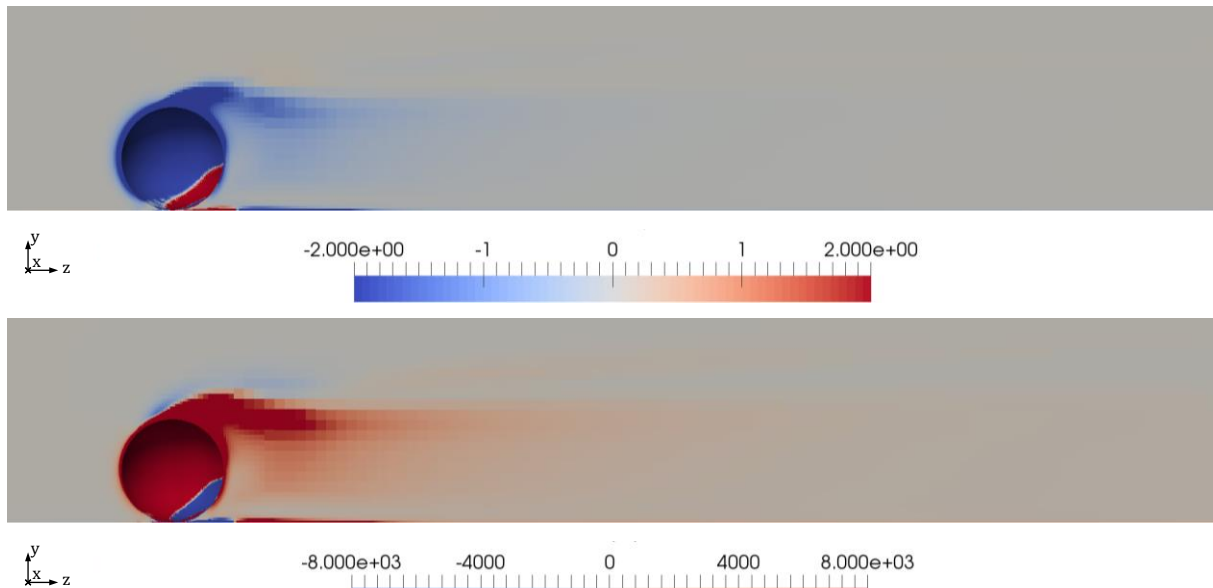


Abbildung 4.20 ε - $C_{\varepsilon i}$ -Sensitivitätenfelder in m^2/s^3 : oben bzgl. $C_{\varepsilon 1} = 1.44$ mit $\Delta C_{\varepsilon 1} = 0.03$, unten bzgl. $C_{\varepsilon 2} = 1.92$ mit $\Delta C_{\varepsilon 2} = -0.10$

Der folgende Detailplot (Abbildung 4.21), der ein freigeschnittenes Achtel des Partikels zeigt, lässt auch erkennen, dass Unstetigkeitsstellen in den Feldern vorliegen.

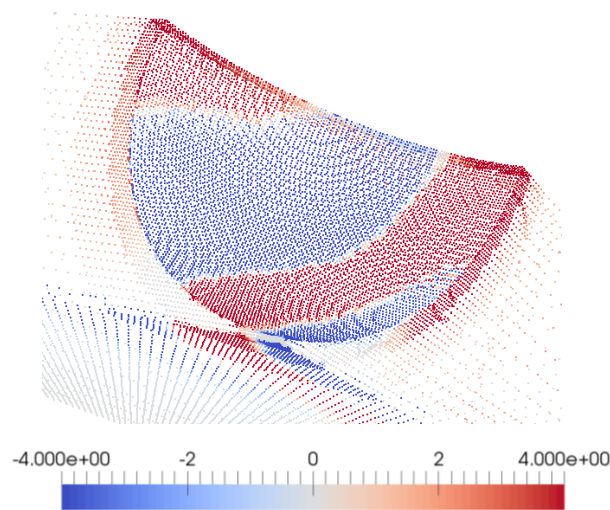


Abbildung 4.21 ε -Sensitivitätenfeld - Detailplot

Deswegen ist das Verhalten von ε bei Parametervariation vermutlich weniger gut geeignet als das Verhalten von k , denn die erhaltenen ε -Sensitivitäten weisen ein geringes Einflussgebiet, steile Gradienten und Unstetigkeitsstellen auf.

4.1 Strömungssimulation

Sensitivitäten - U

Die Sensitivitäten der Geschwindigkeitsfelder weisen relativ komplizierte Verläufe auf. Wenngleich sich die Felder für unterschiedliche Parameter, Schrittweiten und Schrittrichtungen unterscheiden und sich die Anzahl der Plots nur schwer auf wenige repräsentative Abbildungen reduzieren lässt, soll hier nur ein Auszug gezeigt werden, da die Verläufe nur bedingt aufschlussreich und schwer zu interpretieren sind.

In den meisten Fällen weisen die Plots mehrere Regionen mit unterschiedlichen Vorzeichen auf (Abbildung 4.22). Außerdem liegen oftmals Unstetigkeitsstellen vor (Abbildung 4.22, oberer und mittlerer Plot). Sie entstehen, wenn sich ein Gebiet mit starken Gradienten bei der Parametervariation über ein Gebiet mit einheitlichen Werten legt. Dies ermöglicht bei der Differenzenbildung (siehe Gleichung 4.1.1) unterschiedliche Vorzeichen in angrenzenden Zellen.

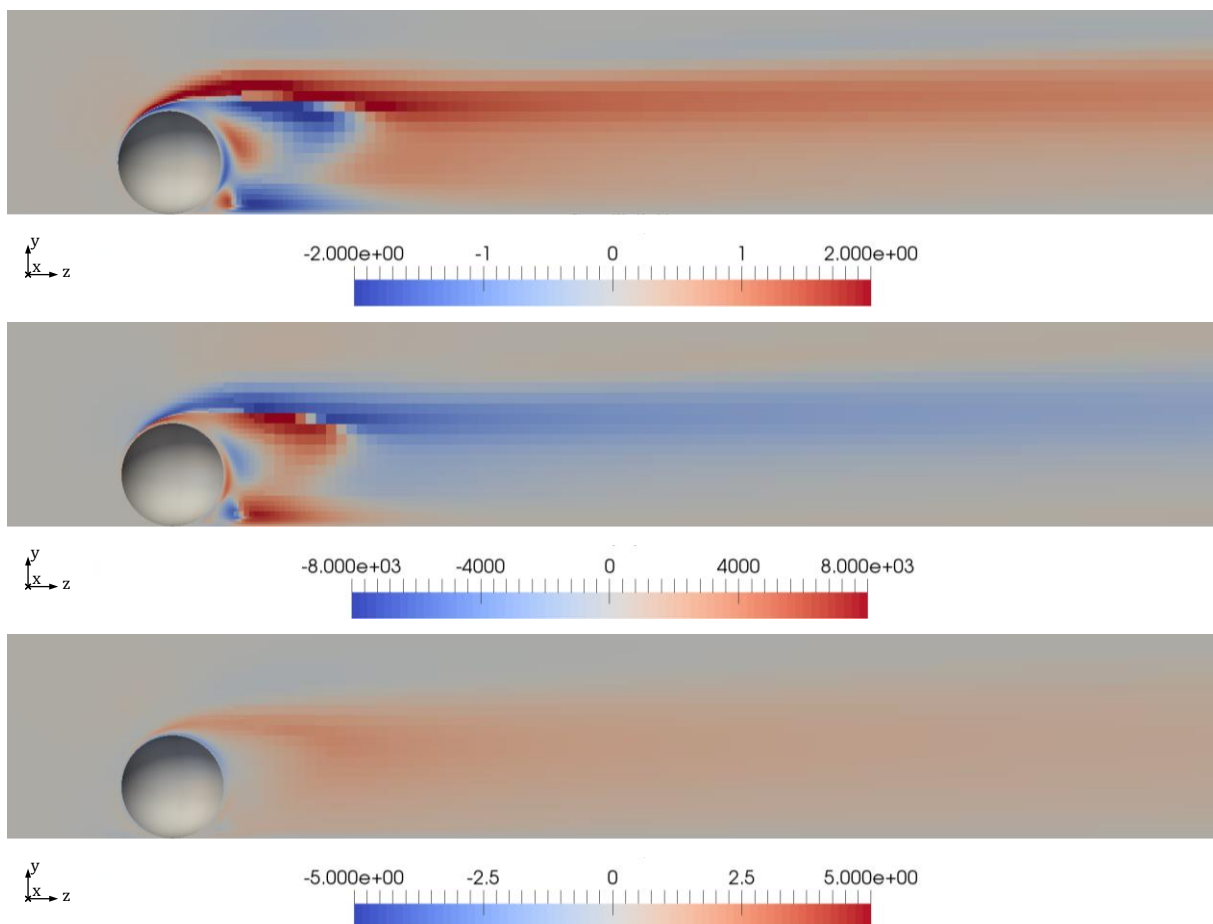


Abbildung 4.22 $\langle \mathbf{U} \rangle$ - C_i -Sensitivitätenfelder in m/s : oben bzgl. $C_\mu = 0.09$ mit $\Delta C_\mu = -0.05$, Mitte bzgl. $C_{\varepsilon 1} = 1.44$ mit $\Delta C_{\varepsilon 1} = 0.06$, unten bzgl. $C_{\varepsilon 2} = 1.92$ mit $\Delta C_{\varepsilon 2} = 17.28$

Das Verhalten von $\langle \mathbf{U} \rangle$ bei Parametervariation ist wesentlich komplexer als das von k und deswegen für einen Optimierer vermutlich weniger gut handhabbar .

4.2 Optimierung

Sensitivitäten – Größenordnungen

Tabelle 4.2 fasst die Größenordnungen der berechneten Sensitivitäten für die fünf Parameter zusammen. Hierbei sind zum Vergleich in der ersten Spalte von rechts die Größenordnungen der Felder angegeben.

Tabelle 4.2 Größenordnung der unterschiedlichen Sensitivitäten.

Sensitivität	$C_{\varepsilon 1}$	$C_{\varepsilon 2}$	C_{μ}	σ_{ε}	σ_k	Feld
$\langle \mathbf{U} \rangle$	1e+0	1e+0	1e+1	1e-1	1e-3	1e+0
k	1e+0	1e+0	1e+1	1e-1	1e-3	1e+0
ε	1e+4	1e+3	1e+4	1e+3	1e+1	1e+4

C_{μ} hat den größten Einfluss auf die untersuchten Felder. Gefolgt von $C_{\varepsilon 1}$ und $C_{\varepsilon 2}$. Die turbulenten Prandtlzahlen σ_{ε} und σ_k weisen, im Vergleich zu C_{μ} , $C_{\varepsilon 1}$ und $C_{\varepsilon 2}$, geringere Sensitivitäten auf. Die untersuchten Felder sind bezüglich σ_k am wenigsten sensitiv.

4.2 Optimierung

Die Sensitivitätsanalyse gab Aufschluss über das komplexe Verhalten der Felder bei Variation eines Modellparameters. Es konnte kein trivialer oder verwertbarer Zusammenhang gefunden werden. Wechselwirkungen durch Variationen mehrerer Parameter wurden nicht untersucht. Es ist davon auszugehen, dass diese zusätzlich zur Komplexität der Problemstellung beitragen. In Anbetracht dessen ist eine Aufbereitung der Ergebnisse bzw. eine Vereinfachung des Systems für den Optimierer notwendig.

4.2.1 Summe der Fehlerquadrate

Das zu optimierende System kann mit der Summe der Fehlerquadrate (SFQ) erheblich vereinfacht werden. Sie wird wie folgt zwischen den k- ε - und den DNS-Ergebnissen gebildet:

$$SFQ = \sum [\phi(Zelle_i)_{k-\varepsilon} - \phi(Zelle_i)_{DNS}]^2 \quad 4.2.1$$

Die SFQ reduziert die Felder mit 1.6Millionen Einträgen auf einen Wert und liefert dem Optimierer eine gut handhabbare Bewertung der k- ε -Ergebnisse. Ist die Summe der Fehlerquadrate 0, so sind k- ε - und DNS-Feld ident.

Aufgrund der Definition mit Gleichung 4.2.1 werden Regionen mit feinerer Zellauflösung stärker gewichtet. Dies sind vor allem Regionen in Wandnähe. Da schlussendlich mit den

4.3 Framework

gewonnenen Strömungsfeldern der Wärmeübergang zwischen Partikel und Platte untersucht werden soll, ist diese Gewichtung zweckmäßig.

Die SFQ wird mit dem OpenFOAM-Post-Processing-Tool *SumSqrErr* bewerkstelligt. Dabei handelt es sich um eine Weiterentwicklung des Programmes *Sensitivity*. Auch *SumSqrErr* ist Teil dieser Arbeit.

Das DNS-Gitter beinhaltet mehr Zellen als das $k-\epsilon$ -Gitter. Um die Berechnung nach 4.2.1 gewährleisten zu können, müssen die DNS-Ergebnisse auf das $k-\epsilon$ -Gitter übertragen werden. Hierfür wurde das OpenFOAM-Pre-Processing-Tool *mapFields* verwendet.

4.2.2 Optimierer - `scipy.optimize.minimize()`

Die Aufgabe des Optimierers ist die Minimierung der SFQ. Die Optimierung wird durch das Python Open-Source Programm `scipy.optimize.minimize()` bewerkstelligt, welches unterschiedliche Optimierer zur Verfügung stellt (The Scipy Community, 2016).

Im Zuge dieser Arbeit wurden einige dieser Optimierer getestet (BFGS, SLQP, TNC). Aufgrund seiner Stabilität und hinreichend hohen Konvergenzgeschwindigkeit, wurde schlussendlich das L-BFGS-B-Verfahren herangezogen. Mit L-BFGS-B können die Grenzen der Optimierungsvariablen gesetzt und somit nicht konvergierende oder divergierende Lösungen vermieden werden. Letzteres ist essentiell für eine effiziente Optimierung (siehe Kapitel 5.2.4). Andere `scipy.optimize.minimize()` Optimierungsverfahren sollen hier für künftige Untersuchungen jedoch nicht ausgeschlossen werden.

4.3 Framework

Das Framework wurde als Erweiterung der Ordner- und Filestruktur des OpenFOAM-Case-Files implementiert (Siehe Kapitel 4.1.1.1). Details zum Aufbau des Frameworks können aus dem Appendix entnommen werden.

4.3.1 Framework - lokale Optimierer

Die grundsätzliche Funktionsweise des Frameworks für die lokale Optimierung wird in folgender Abbildung 4.23 illustriert:

4.3 Framework

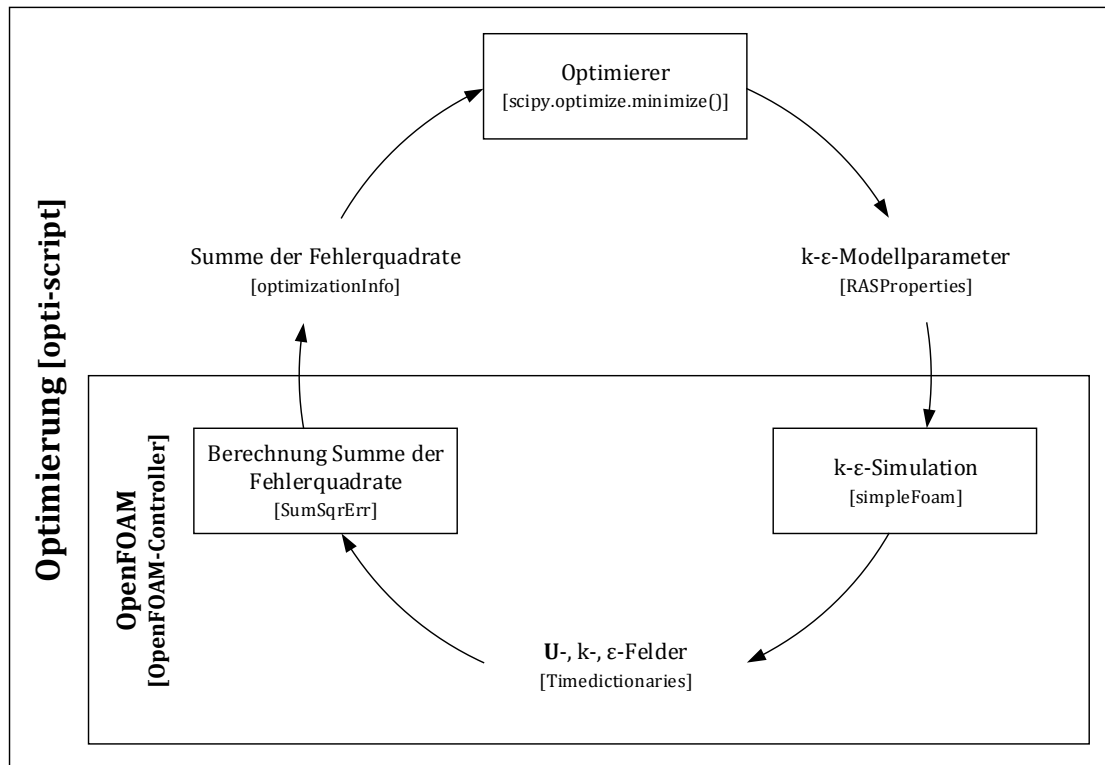


Abbildung 4.23 Schematische Darstellung des Frameworks

Das Framework, *opti-script* genannt, ermöglicht den Informationsaustausch zwischen dem Optimierer (`scipy.optimize.minimize()`) und dem Strömungssimulator (OpenFOAM). Dabei wird die SFQ dem Optimierer übertragen (Textfile: *optimizationInfo*). Dieser bestimmt verbesserte k-ε-Modellparameter. Die Parameter werden wiederum an die Strömungssimulation übergeben (Textfile: *RASProperties*). Diese berechnet die dazugehörigen Felder (*simpleFoam*). Die SFQ wird bestimmt (*SumSqrErr*) und erneut an den Optimierer übergeben. Der Kreis schließt sich, eine iterative Optimierung der k-ε-Modellparameter ist gewährleistet. Da davon ausgegangen werden kann, dass die iterativen Parameteränderungen durch den Optimierer klein sind, zieht *simpleFoam*, als Startfelder einer neuen Berechnung, die Lösungsfelder der letzten konvergierten Berechnung heran.

OpenFOAM ist in dem *OpenFOAM-Controller* (OC) eingebettet. Dieser ist ein integraler Bestandteil des Frameworks. Als eigenständiges Programm kann er für andere Optimierer herangezogen werden. Er steuert und startet die notwendigen OpenFOAM-Programme je nach Berechnungsausgang und wartet das Case-File.

Abbildung 4.24 stellt das Framework detaillierter dar. Die grundlegende Hierarchie ist nach wie vor zu erkennen. Das *opti-script* verbindet den Optimierer und den OC.

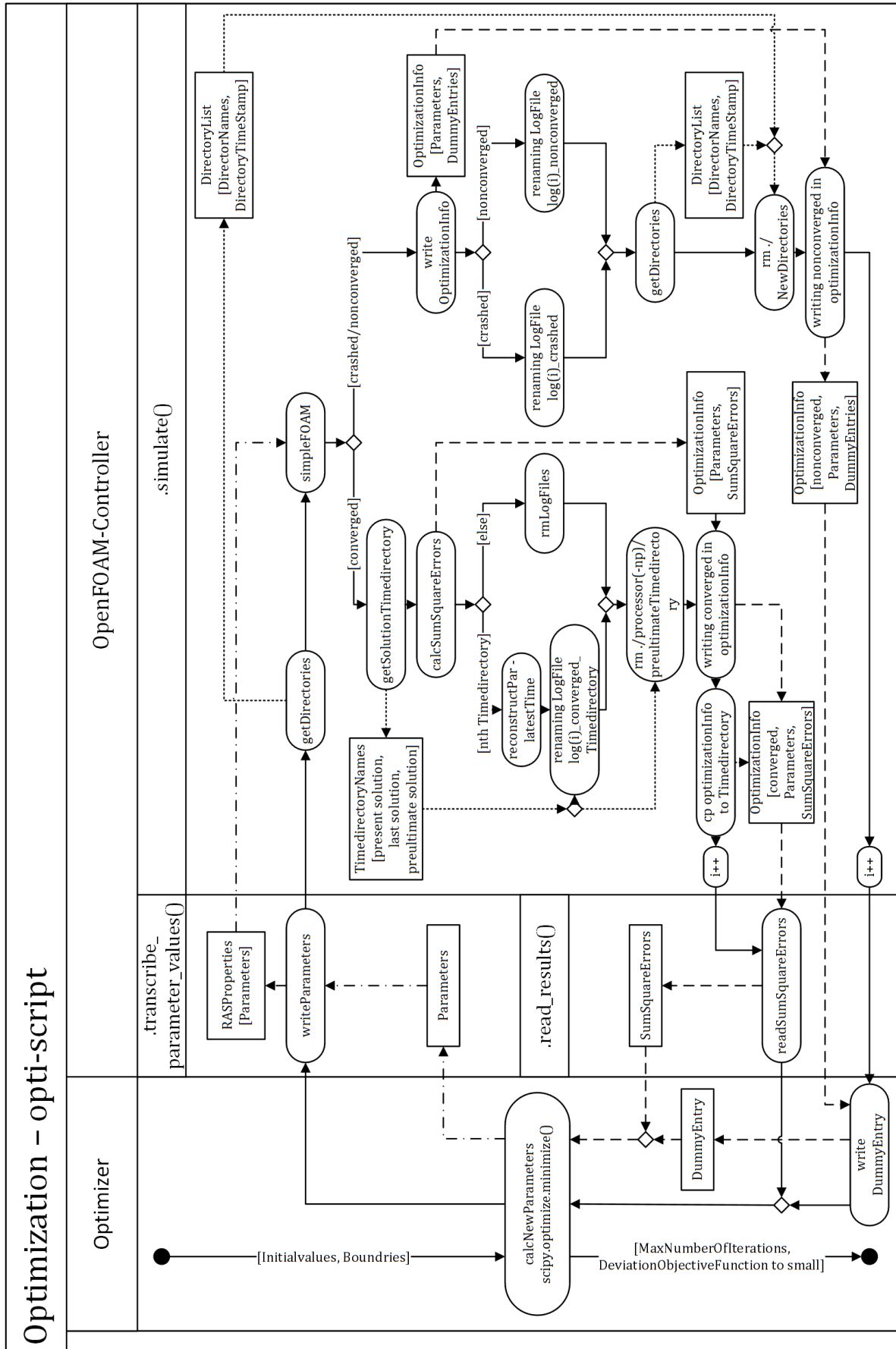


Abbildung 4.24 UML-Aktivitätsdiagramm des Frameworks: Aktivitäten: eckige Flächen, Informationen: eckige Flächen, Aktivitätenablauf: durchgehende Linien, Informationsflüsse: SFQ; strichliert, Parameter: strichpunktirt, Ordnerinformationen: kurz strichliert.

4.3 Framework

Der strukturelle Aufbau des OC ist ebenfalls ersichtlich. Dieser untergliedert sich in drei Methoden. Zwei davon (*transcribe_parameter_values()*) und *read_results()*) bilden das Interface zwischen dem Optimierer und der k- ϵ -Simulation. Sie sind für das Schreiben der Parameter in RASProperties bzw. das Lesen der SFQ aus optimizationInfo zuständig (siehe Abbildung 4.23). Die Übergabe der Parameter vom Optimierer an *transcribe_parameter_values()*, sowie die Abnahme der Fehlerquadrate von *read_results()* erfolgt durch das opti-script.

Mit der Initialisierung des OC, durch das opti-script beim erstmaligen Aufruf, wird die Domain für eine effizientere Parallelberechnung auf mehreren Prozessoren (*mpi-run*) aufgeteilt (*decomposed*). Dabei wird für jeden Prozessor ein Ordner angelegt. In ihnen liegen die zur Berechnung herangezogenen und aufgeteilten Lösungen/Felder. Der OC ermöglicht auch eine Änderung der Anzahl der Prozessoren während einer laufenden Optimierung. Der Übersichtlichkeit halber sind diese Funktionen im UML-Aktivitätsdiagramm nicht eingezeichnet.

Hauptbestandteil des OC ist die Methode (*simulate()*). Durch sie wird im Wesentlichen die k- ϵ -Simulation gestartet und die Berechnung der SFQs veranlasst. Das Programm läuft dabei wie folgt ab:

- i. Zunächst wird der Ausgangszustand des OpenFOAM-Case-Files (siehe Kapitel 4.1.1.1) durch die darin enthaltenen Ordnernamen und die dazugehörigen Änderungszeitpunkte festgehalten.
- ii. Die k- ϵ -Simulation wird mit *simpleFOAM* gestartet. Sie greift standardmäßig auf das Übergabefile RASProperties zu und schreibt zusätzliche Berechnungsinformationen in ein log-File. Die Berechnung wird beendet, wenn die Lösung konvergiert oder wenn die maximale Anzahl an Iterationen erreicht wird. Auch eine Beendigung der k- ϵ -Simulation durch Absturz aufgrund ungeeigneter Parameterwahl ist möglich und kann vom OC erkannt werden.

Ab diesem Zeitpunkt ist der weitere Ablauf von *simulate()* vom Berechnungsausgang abhängig. Dabei wird zwischen konvergiert und allen anderen Fällen (nicht konvergiert, abgestürzt) unterschieden.

4.3 Framework

Berechnung konvergiert:

- iii. Der Name des Timedirectorys (siehe Kapitel 4.1.1.1) der aktuellen Lösung wird aus dem log-File der Berechnung ermittelt.
- iv. Die SFQs werden mit SumSqrErr berechnet und diese samt den aktuellen Parametern in optimizationInfo ausgegeben.
- v. Um Speicherplatz zu sparen wird nur ein Teil der Lösungen und das dazugehörige log-File behalten. Das log-File wird mithilfe von iii. log'Laufnummer'_converged_'NameTimedirectory' benannt.
- vi. Aus iii. ist der Name des Timedirectories mit der vorletzten Lösung ermittelbar. Letztere wird zur Reduktion des Speicherplatzes aus den Prozessorordnern gelöscht.
- vii. Die Information „Lösung konvergiert“ wird in optimizationInfo geschrieben. Diese kann dann von read_results() gelesen werden.
- viii. Die Laufnummer i wird erhöht.

Berechnung nicht konvergiert/abgestürzt:

- iii. Das entwickelte OpenFOAM-Postprocessing-Tool *optimizationInfoNC*⁸ legt das Übergabefile optimizationInfo an. Dieses enthält die Parameter und hohe Strafwerte für die einzelnen SFQs.
- iv. Die Information „Lösung nicht konvergiert“ wird in optimizationInfo geschrieben.
- v. Das log-File wird je nach Berechnungsausgang (nicht konvergiert/abgestürzt) log'Laufnummer'_nonconverged/crashed benannt.
- vi. Analog zu i. wird die Ordnerstruktur und der Zustand des Case-Files festgehalten.
- vii. Die aus i. und vi. gewonnenen Informationen werden abgeglichen. Alle bearbeiteten oder neu angelegten Ordner und Files werden gelöscht, das Case-File wird auf seinen Ausgangszustand vor der k- ϵ -Simulation zurückgeführt.
- viii. Die Laufnummer i wird erhöht.

Da eine Änderung der Strafwerte eine neue Kompilierung von optimizationInfoNC erfordern würde, kann auch mit dem opti-script ein Strafwert der Zielfunktion festgelegt werden.

⁸ Dieses Programm entspricht SumSqrErr ohne Berechnung der Summe der Fehlerquadrate. Stattdessen übergibt es hohe, vor der Kompilierung festgelegte, Strafwerte.

4.3 Framework

4.3.2 Framework - Hammersley-Verteilung

Das Framework zur automatisierten Berechnung der Felder von Hammersley verteilten Parametern, zieht nur die Methode `simulate()` des OC heran. Jeder Hammersleypunkt ist dabei in einer Zeile des Textfiles Hammersley eingetragen. Aus dem File wird eine Zeile ausgelesen, die Parameter in `RASProperties` übertragen, die Felder berechnet, die SFQ gebildet und in das entsprechende Timedirectory abgelegt. Dieser Vorgang wird durch das Programm `Script.py` in einer Schleife für jede Zeile ausgeführt.

Anstelle von Hammersley-verteilten Punkten, kann in das Textfile Hammersley eine beliebige Ansammlung von Punkten geschrieben und somit dem Hammersley-Framework übergeben werden.

5 Ergebnisse und Diskussion

Zur Abschätzung der Güte von unterschiedlichen Parametersätzen wurden in dieser Arbeit die SFQs der zugehörigen Felder $\langle U_x \rangle, \langle U_y \rangle, \langle U_z \rangle, k$ und ε untersucht. Die SFQ ist in den folgenden Abbildungen jeweils durch die SFQ für den Standardparametersatz dividiert. Diese Normierung lässt eine Verbesserung (Werte < 1) bzw. Verschlechterung (Werte > 1) der Felder schnell erkennen.

5.1 Hammersley-Verteilung

Hammersley-Verteilungen dienen zur Generierung geeigneter Startparameter für die lokale Optimierung. Im Zuge dieser Masterarbeit wurden die SFQs mit Hilfe von zwei Hammersley-Sätzen, mit jeweils 100 Hammersley-Punkten untersucht. Die beiden Untersuchungen unterschieden sich dabei in der Größe des Raumes, in dem die Punkte verteilt wurden. Tabelle 5.1 fasst die erhaltenen Ergebnisse zusammen:

Tabelle 5.1 Vom Framework bestimmte Zielfunktionswerte einer Optimierungsiteration.

Unter- suchung	Untergrenze	Obergrenze	Konvergiert	Besser als Standard- Par.-Satz	nicht konvergiert	abgestürzt	Anzahl der Punkte
1	$P_{i,Standard}/100$	$P_{i,Standard} \cdot 100$	16	0	38	46	100
2	$P_{i,Standard} \cdot 0.5$	$P_{i,Standard} \cdot 1.5$	48	5	52	0	100

In Untersuchung 1 wurden die Hammersley-Punkte weit verteilt. Die Obergrenze beträgt das Hundertfache und die Untergrenze entspricht einem Hundertstel der Standardparameter. Bei derart großen Abweichungen vom Standardparametersatz, kommt es oftmals zum Absturz von simpleFOAM. In dem untersuchten Bereich lag die Absturzquote bei 46%. In 38% der Fälle konvergierten die Residuen nicht auf den geforderten Wert von $1E-5$. Insofern weisen nur 16% der Parametersätze gültige, konvergierte Lösungen auf. Deren SFQs sind in Abbildung 5.1 dargestellt. Die Parametersätze wurden hierbei nach SFQ- k geordnet.

5.1 Hammersley-Verteilung

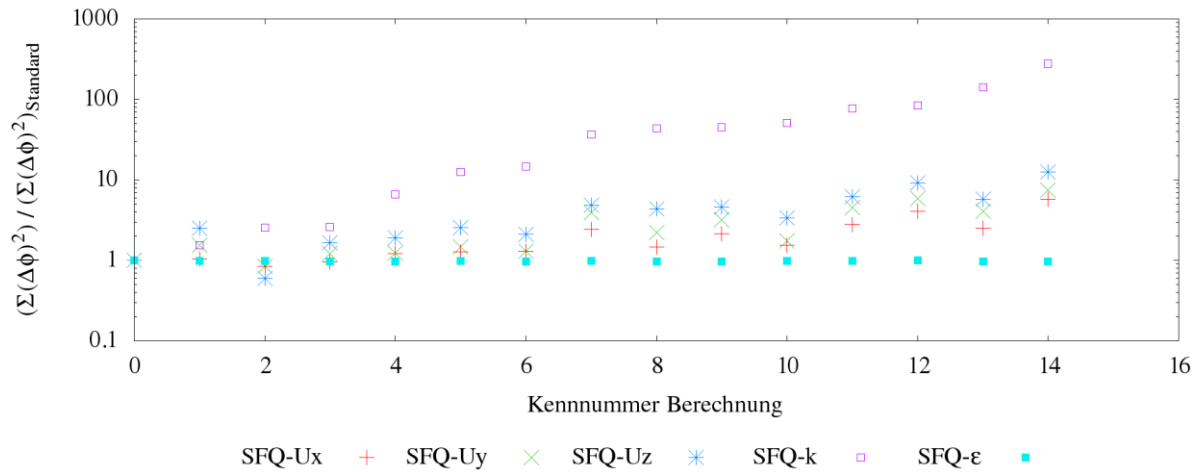


Abbildung 5.1 SFQs Hammersley-Untersuchung 1, Parametersätze geordnet nach SFQ-k

Aus Abbildung 5.1 ist ersichtlich, dass keiner der 100 untersuchten Parametersätze besser als der Standardparametersatz für die betrachtete Strömung geeignet ist.

In Untersuchung 2 wurde der Bereich reduziert. Die Ober- und Untergrenze beträgt $\pm 50\%$ des Standardparametersatzes. Durch enger gesetzte Grenzen, können Abstürze von simpleFoam verhindert werden (Absturzquote 0%). Außerdem konnte der Anteil der konvergierenden Lösungen auf 48% angehoben werden. Abbildung 5.2 zeigt die SFQs der konvergierten Parametersätze. 5% der untersuchten Parametersätze weisen geringere SFQs als der Standardparametersatz auf. Sie stellen daher geeignete Startparameter für eine lokale Optimierung dar.

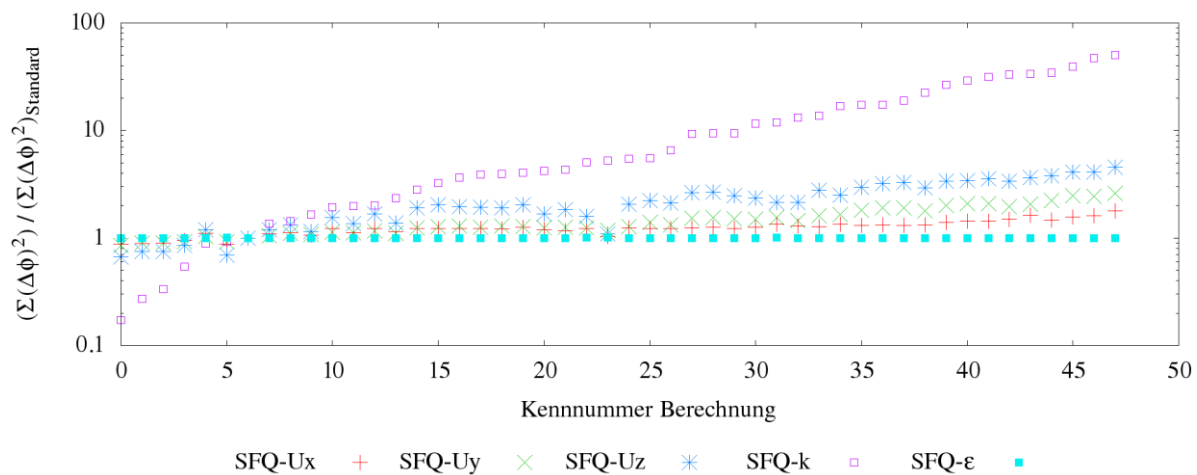


Abbildung 5.2 SFQs Hammersley-Untersuchung 2, Parametersätze geordnet nach SFQ-k

Tabelle 5.2 zeigt diese 5 Parametersätze. Aus ihr ist ersichtlich, dass sowohl Erhöhungen als auch Senkungen desselben Parameters Verbesserungen der Felder bewirken konnten. Wesentlich scheint daher die Kombination der Parameter zu sein. Aufgrund der

5.1 Hammersley-Verteilung

komplizierten Wechselwirkung der einzelnen Parameter zueinander, lässt sich aus den Ergebnissen in Tabelle 5.2 jedoch kein für die Optimierung nützlicher Zusammenhang ableiten.

Tabelle 5.2 Parametersätze mit niedrigeren SFQs als der Standardparametersatz.

$(SFQ - k)/(SFQ - k)_{standard}$	$C_{\varepsilon 1}$	$C_{\varepsilon 2}$	C_{μ}	σ_{ε}	σ_k
0.171	2.069	2.540	0.126	0.715	0.607
0.270	2.400	3.040	0.135	1.950	1.500
0.334	2.161	2.840	0.099	1.608	0.657
0.542	1.766	2.391	0.080	1.062	0.979
0.923	0.995	1.126	0.073	0.908	1.409

Der Berechnungsaufwand zur Bestimmung der SFQs von Hammersley-verteilten Punkten ist abhängig von der Größe des untersuchten Bereiches. Untersuchung 1 mit weiteren Grenzen nahm 18 Tage und somit 12 Tage mehr als Untersuchung 2 in Anspruch. Dies kann wie folgt erklärt werden. Das Framework zieht als Startfelder für die Berechnung eines Punktes die Lösungsfelder des vorangegangenen Punktes heran. Aufgrund engerer Grenzen sind die Startfelder bei kleinen Parameteränderungen bessere Abschätzungen der Lösungen. Die Berechnungsdauer ist insofern kürzer und der Berechnungsaufwand sinkt mit kleineren Grenzen. Wird die Berechnungsreihenfolge so gewählt, dass der nächstberechnete Punkt immer der nächstgelegene Punkt ist, kann die Berechnung effizienter gestaltet werden. Die dafür notwendige Sortierung der Hammersley-Punkte kann durch `OrderHammersleyPoints.py` vorgenommen werden. Dieses Tool wurde im Rahmen dieser Masterarbeit entwickelt.

Unter Berücksichtigung des hohen Berechnungsaufwandes (18 Tage) und im Hinblick auf die erhaltenen Ergebnisse (keine geeigneten Startparameter für die lokale Optimierung, Instabilität in weiten Teilen des Untersuchungsbereiches), waren die für Untersuchung 1 gewählten Grenzen zu weit.

Die Grenzen für Untersuchung 2 scheinen geeigneter zu sein. Sie sind enger gesetzt. Hierdurch liegen die Hammersley-verteilten Parametersätze näher am Standardparametersatz. Dies ist physikalisch durchaus sinnvoll. Denn wie in Kapitel 4.1.5.1 diskutiert, verhält sich die untersuchte Strömung gegenüber den Benchmark-Strömungen in Kapitel 2.5.2.2 qualitativ ähnlich. Werden Letztere herangezogen um einen

5.2 Lokale Optimierung

k - ε -Modellparametersatz zu bestimmen, so liegt dieser nahe am Standardparametersatz und innerhalb der eng gesetzten Grenzen von Untersuchung 2.

Auch in Hinblick auf die numerische Stabilität ist die Wahl der Grenzen geeignet. Eine Konvergenzwahrscheinlichkeit von knapp 50% stellt einen guten Kompromiss zwischen einem großen Suchbereich und einer hohen Anzahl an potentiellen Startwerten (konvergierten Lösungen) dar. Um eben diese Anzahl zu erhöhen, könnte auch eine Verdichtung der Hammersley-Verteilung vorgenommen werden. Hierbei würden bei gleichbleibenden Bereich die Anzahl der Hammersley-Punkte erhöht werden. In Anbetracht der geringen Ausbeute an geeigneten Startparametern, kann die Verdichtung der Verteilung auf Gebiete nahe der besten Hammersley-Punkte, mit den geringsten SFQs, konzentriert werden. Dies könnte den Anteil der geeigneten Startparametersätze, an den insgesamt berechneten Parametersätzen, erhöhen. Gleichzeitig würde der Untersuchungsbereich kleiner werden und insofern die Berechnungsdauer sinken, sowie die Effizienz steigen.

Abschließend muss festgestellt werden, dass die Auswertung von Hammersley-Punkten für den Erhalt besserer Parametersätze oder Startparametersätze relativ aufwendig ist. Auch die oben beschriebenen Maßnahmen, zur Steigerung der Effizienz und Ausbeute, werden das Kosten/Nutzen-Verhältnis nicht wesentlich verbessern können. Die Verwendung von Hammersley-Punkten ist daher nur sinnvoll und verhältnismäßig, wenn die Zielfunktion viele lokale und unterschiedlich stark ausgeprägte Minima aufweist. Ist die Zielfunktion glatt, erzielen lokale Optimierer effizienter bessere Ergebnisse.

5.2 Lokale Optimierung

Zur Anpassung der Einstellungen des Optimierers und zur Überprüfung der Funktion des Frameworks für die lokale Optimierung, wurden umfangreiche Untersuchungen durchgeführt. Das Vorgehen soll im Folgenden beschrieben werden:

5.2.1 Reduktion Optimierungsvariablen

Wird die Anzahl der Optimierungsvariablen auf zwei reduziert, lässt sich die Zielfunktion dreidimensional darstellen. Dadurch kann das Verhalten des Optimierers illustriert und somit gut nachvollzogen werden. Die Wahl der Variablen fiel dabei mit folgender Begründung auf $C_{\varepsilon 1}$ und $C_{\varepsilon 2}$:

5.2 Lokale Optimierung

- i. Der Einfluss der Parameter $C_{\varepsilon 1}$ und $C_{\varepsilon 2}$ auf die Strömungsfelder kann theoretisch am besten nachvollzogen werden (Kapitel 2.5.2.2).
- ii. $C_{\varepsilon 1}$ und $C_{\varepsilon 2}$ befinden sich in einer komplett modellierten Transportgleichung. Ihre Anpassung ist daher durchaus sinnvoll (Kapitel 2.5.3.3, 2.5.3.6).
- iii. Die Sensitivitäten der Parameter $C_{\varepsilon 1}$ und $C_{\varepsilon 2}$ scheinen für die Optimierungsaufgabe am besten geeignet zu sein (Kapitel 4.1.5).

5.2.2 Zielfunktion

Zunächst wurden die SFQs für ein zweidimensionales Feld, aufgespannt durch $C_{\varepsilon 1}$ und $C_{\varepsilon 2}$, berechnet. Hierzu wurde das Hammersley-Framework herangezogen, welches die im Input-File vorgegebenen Punkte in einer Schleife berechnet. Anstatt einer Hammersley-Verteilung wurde hierfür ein rechteckiges Raster vorgegeben. Die Berechnungsreihenfolge lief dabei, aus Effizienzgründen, das Raster schlangenlinienförmig ab. Dadurch entstehen die periodischen Verläufe in Abbildung 5.3:

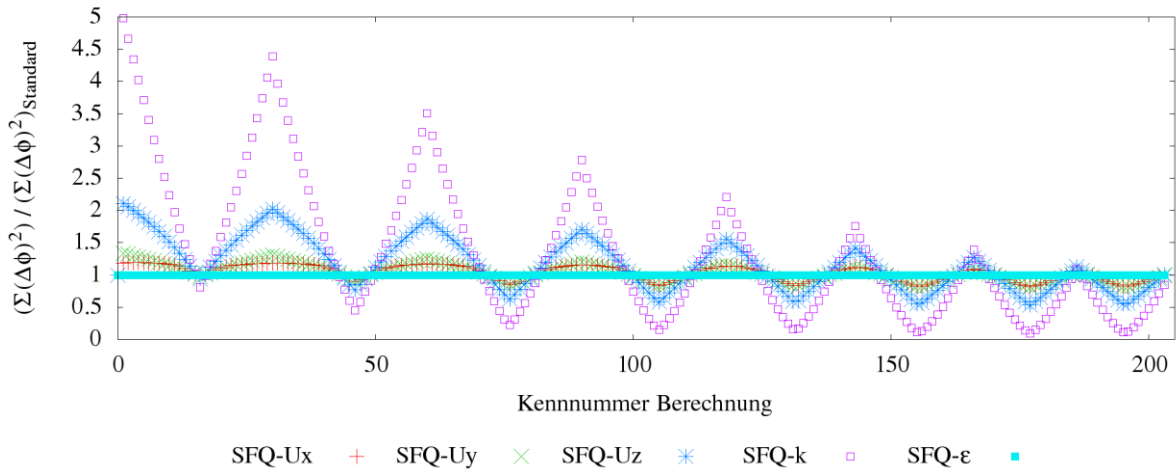


Abbildung 5.3 Überprüfung Framework - SFQ des Feldes $\phi = \langle \mathbf{U} \rangle, \varepsilon, k$ für einen Parametersatz, normiert durch die SFQ des Standardparametersatzes.

Offensichtlich korrelieren die SFQs der unterschiedlichen Felder, mit Ausnahme der SFQs von ε . Diese werden durch die Parametervariation nicht beeinflusst. Die geringsten Sensitivitäten werden für die SFQs von $\langle U_x \rangle$ und $\langle U_y \rangle$ verzeichnet. Sie sind in etwa gleich groß. Für die SFQ von $\langle U_z \rangle$ lässt sich ein Rückgang um bis zu 50% und ein Anstieg um mehr als 100% verzeichnen. $\langle U_z \rangle$ weist die sensibelsten SFQs der Geschwindigkeitsfelder auf. Die SFQ von k ist von allen Feldern am sensitivsten. Der niedrigste Wert liegt bei etwa 0.1, ein Anstieg auf 5.0 ist ohneweiters möglich.

5.2 Lokale Optimierung

Nach obigen Ergebnissen hat der Rückgang einer SFQ eine Absenkung aller SFQs zur Folge. Dieses grundlegende Verhalten kann auch in Abbildung 5.1 und Abbildung 5.2 beobachtet werden und war für alle getätigten Berechnungen zu verzeichnen. Die Wahl der Zielfunktion fiel insofern auf die sensitivste normierte SFQ, die des Feldes k .

Die Zielfunktion wird in Abbildung 5.4 für das berechnete Raster dreidimensional dargestellt. Zur Überprüfung der Oberflächenglätte erfolgten Kontrollberechnungen, mit dichter gesetzten Berechnungspunkten. Für diese wurde ebenfalls das Hammersley-Framework herangezogen. Die dargestellte Oberfläche gibt hierbei den Verlauf der Kontrollberechnungen gut wieder. Bei derart glatten Zielfunktionsverläufen, ohne lokale Minima, ist die Auswertung der SFQs durch eine Hammersley-Verteilung nicht von Nöten und daher unverhältnismäßig aufwendig.

Die Farbskalierung wurde so gewählt, dass eine Verbesserung (blau) bzw. Verschlechterung (rot) der Zielfunktion gut ersichtlich ist. Wie durch die Ergebnisse der Sensitivitätsanalyse (Kapitel 4.1.5) zu erwarten war, ist eine Verbesserung der Felder durch Erhöhung von $C_{\varepsilon 1}$ und Senkung von $C_{\varepsilon 2}$ möglich.

Im Bereich für hohe $C_{\varepsilon 1}$ und niedrige $C_{\varepsilon 2}$ treten Instabilitäten der Lösungen auf. Es konnte für diese Punkte keine Zielfunktion berechnet werden. Der unstetige Übergang von Weiß zu Blau stellt also approximativ die Stabilitätsgrenze dar. Für das hier vorliegende Ergebnis, liegt sie am Minimum.

5.2 Lokale Optimierung

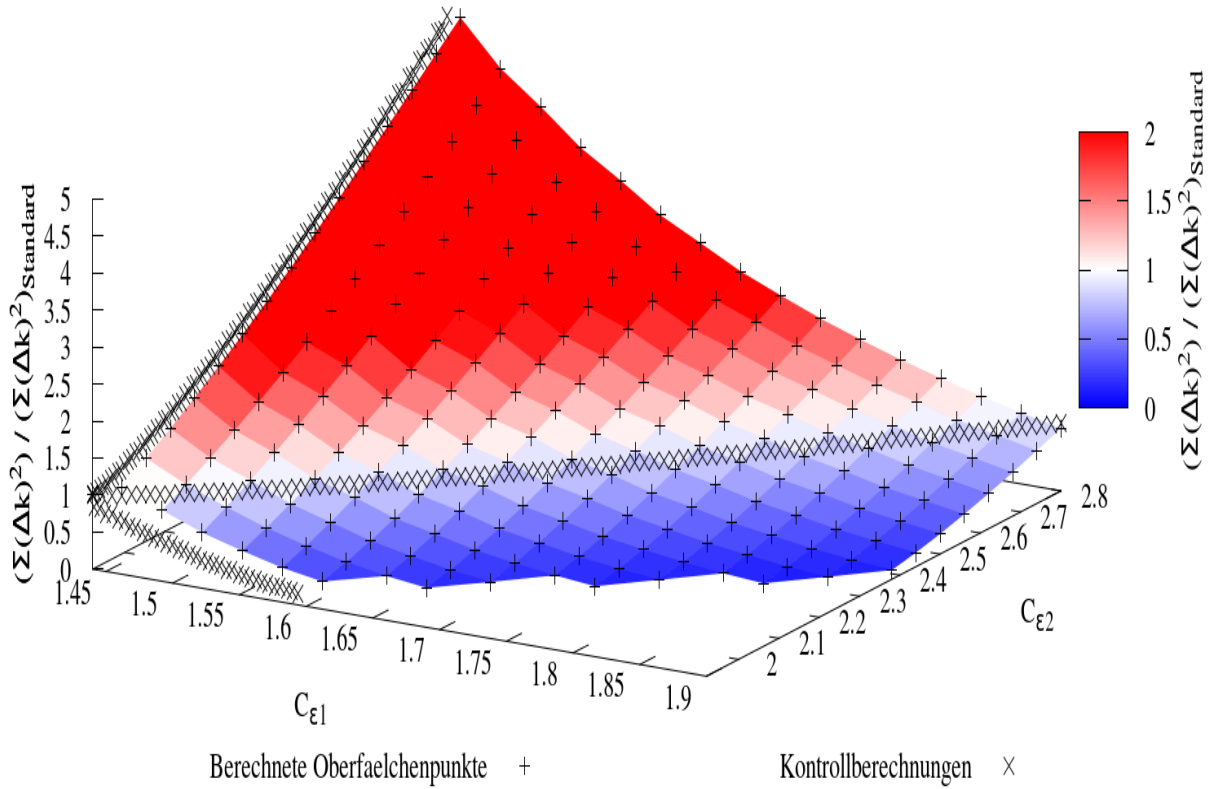


Abbildung 5.4 Dreidimensionale Darstellung der Zielfunktion für einen ausgewählten Bereich.

Abbildung 5.5 zeigt eine zweidimensionale Projektion der Zielfunktion. In ihr sind zwei Geraden eingezeichnet. Die strichlierte Gerade beschreibt nach Gleichung 2.5.58 ein konstantes Verhältnis von $\mathcal{P}/\varepsilon = 2.09$. Sie setzt die Konstanten $C_{\varepsilon 1}$ und $C_{\varepsilon 2}$ mit \mathcal{P}/ε in Beziehung und ist strenggenommen nur für homogen turbulente Scherströmungen gültig. Dennoch vermag diese Gerade jenen Bereich abzuschätzen, in dem es zu keiner Verbesserung oder Verschlechterung der Zielfunktion kommt. Eine Erhöhung von \mathcal{P}/ε nach Gleichung 2.5.58 führt insofern zur Verschlechterung der Zielfunktion. Eine Senkung von \mathcal{P}/ε geht mit einer Verbesserung einher. Die Produktion \mathcal{P} der turbulenten kinetischen Energie k wird daher von den Standardparametern überschätzt. Dieses Verhalten ist für eine Scherströmung mit homogener Turbulenz bekannt (siehe Kapitel 2.5.2.2). Ein Vergleich der k -Felder zwischen DNS und dem k - ε -Modell, unter Verwendung der Standardparameter, lässt diesen Rückschluss ebenfalls zu (Abbildung 4.3). Die am Eintritt nicht turbulente Strömung weist beim k - ε -Modell, in weiten Bereichen des Nachlaufes, weitaus höhere k -Werte als die DNS auf. Hierfür ist eine überschätzte \mathcal{P} eine naheliegende Erklärung.

Die zweite Gerade (durchgezogene Linie) kann durch Gleichung 2.5.69 erhalten werden. Sie wurde für eine turbulente Scherströmung in Wandnähe hergeleitet und geht in eine Geradengleichung über, wenn C_μ und σ_ε konstant gehalten werden. In Abbildung 5.5

5.2 Lokale Optimierung

beschreibt die Gerade eine Van Kármán-Konstante von 0.43, unter Verwendung der Standardwerte für C_μ und σ_ε . Sie verläuft nahe der Stabilitätsgrenze. Diese Lage ist theoretisch plausibel. Nach Pope (2000) ist σ_ε für die Stabilität der Lösung maßgeblich. Denn durch ihren Einfluss auf den diffusiven Transport von ε ermöglicht sie einen Ausgleich von Quell- und Senkenterm und somit eine Glättung der Felder.

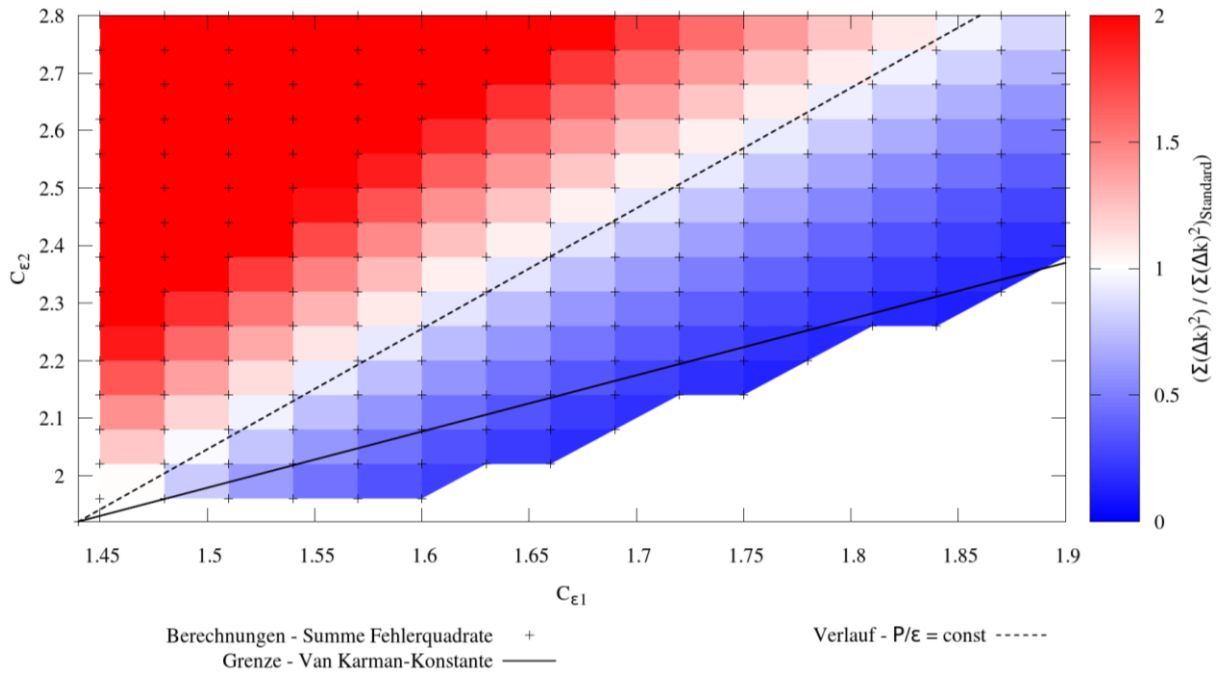


Abbildung 5.5 Zweidimensionale Darstellung der Zielfunktion für einen ausgewählten Bereich

5.2.3 Rauschen der Zielfunktion

Mit der in Kapitel 5.2.2 beschriebenen Zielfunktion wurde das Framework für die lokale Optimierung getestet. Abbildung 5.6 zeigt einen kleineren Ausschnitt von Abbildung 5.5. In ihr eingezeichnet sind die Berechnungswege zweier Optimierungen. Anhand der zweiten Farbskalierung kann der Fortschritt der Optimierung, bzw. die Abfolge der Berechnungsschritte, nachvollzogen werden.

5.2 Lokale Optimierung

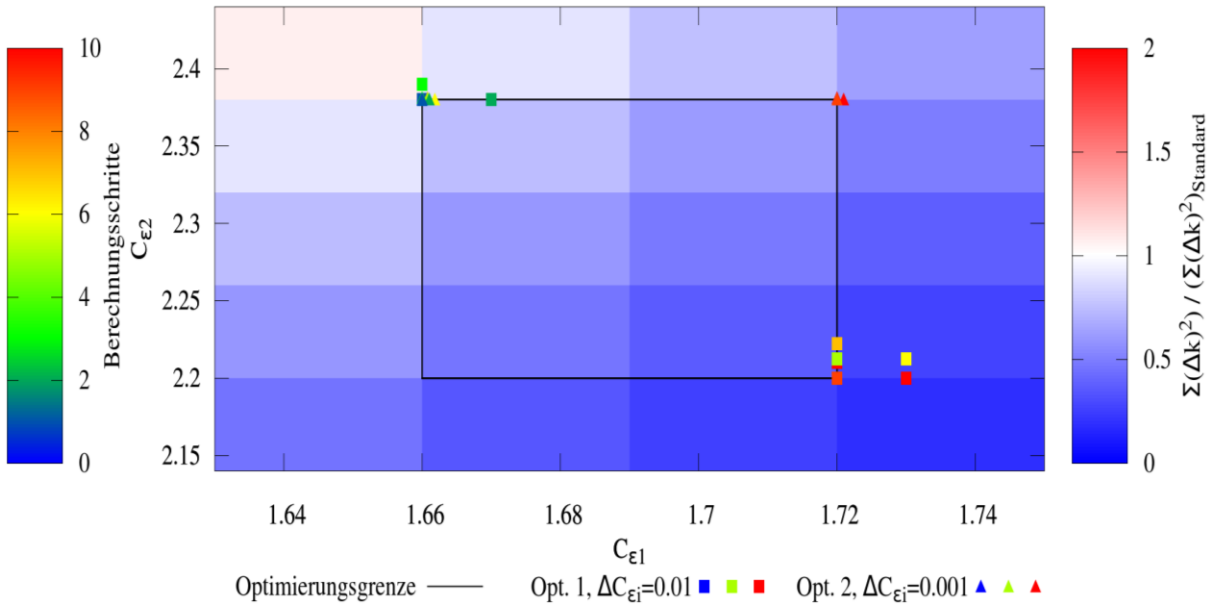


Abbildung 5.6 Berechnungsverläufe der Optimierer mit unterschiedlichen Schrittweiten der Sensitivitäten (eps)

Wie in Kapitel 3.1.3 beschrieben, berechnet das L-BFGS-B-Verfahren zur Minimierung der Zielfunktion f die Gradienten der Zielfunktion nach den Optimierungsvariablen $\Delta f / \Delta C_{\varepsilon i}$. Obige Abbildung zeigt, dass der Optimierungserfolg in dem entwickelten Framework von der gewählten Gradientenschrittweite $\Delta C_{\varepsilon i}$ abhängig ist (Quadrate: $\Delta C_{\varepsilon i} = 0.01$, Dreiecke: $\Delta C_{\varepsilon i} = 0.001$). Beide Optimierungen starten am Punkt (1.66, 2.38), jedoch nur eine Optimierung (Quadrate) findet das Minimum bei (1.72, 2.2), für die gegebenen und eingezeichneten Grenzen.

Um das Versagen bei kleinen Gradientenschrittweiten zu erklären, soll zunächst der korrekte Ablauf einer Optimierungsiteration mit $\Delta C_{\varepsilon i} = 0.01$ beschrieben werden. Dieser ist in Abbildung 5.6 am Punkt (1.72, 2.21) (Quadrat, grün) nachvollziehbar:

- i. Die Zielfunktion für (1.72, 2.21) (grün) wird berechnet.
- ii. Die Zielfunktion für (1.72, 2.21) (grün) wird erneut berechnet (In Abbildung nicht zu sehen).
- iii. Die Zielfunktion für (1.73, 2.21) (gelb) wird berechnet um den Gradienten in $C_{\varepsilon 1}$ -Richtung zu bilden.
- iv. Die Zielfunktion für (1.72, 2.22) (orange) wird berechnet um den Gradienten in $C_{\varepsilon 2}$ -Richtung zu bilden.
- v. (Die Zielfunktion wird für jeden weitere Optimierungsvariable einmal berechnet, um den entsprechenden Gradienten zu bilden)
- vi. Es erfolgt ein Schritt entgegen dem Gradienten (1.72, 2.20)(rot-orange)

5.2 Lokale Optimierung

Bei der Optimierung mit Gradientenschrittweiten von 0.001 (Dreiecke) konnte mit diesem Ablauf nicht die richtige Schrittrichtung für $C_{\varepsilon 2}$ bestimmt werden. Die Optimierung wird an der Grenze am Punkt (1.66, 2.20) abgebrochen. Dieser Punkt stellt kein Minimum für die gegebenen Grenzen dar. Das Fehlverhalten kann anhand der gebildeten Zielfunktionswerte, exemplarisch angeführt in Tabelle 5.3, erklärt werden.

Tabelle 5.3 Vom Framework bestimmte Zielfunktionswerte einer Optimierungsiteration.

$C_{\varepsilon 1}$	$C_{\varepsilon 2}$	$\Sigma \Delta k^2 / \Sigma \Delta k_{Standard}^2$
1.720	2.380	0.70582
1.720	2.380	0.70581
1.721	2.380	0.70579
1.720	2.381	0.70574

Ein Rauschen der Zielfunktion kann bei Vergleich der ersten zwei Einträge festgestellt werden. Die Werte variieren trotz der exakt gleichen Parameter um 1E-5. Dieses Rauschen suggeriert auch dem Optimierer einen Abfall der Zielfunktion in der Höhe von ca. 7E-5 bei einer Erhöhung von $C_{\varepsilon 2}$ um 1E-3. Dies ist jedoch nach Abbildung 5.4 nicht der Fall. Nach dieser ist qualitativ ein Anstieg der Zielfunktion bei Erhöhung von $C_{\varepsilon 2}$ zu erwarten.

Das Rauschen wird durch die Beschaffenheit des Frameworks verursacht. In diesem sind die Startfelder der k- ε -Simulation, die Lösungsfelder der vorangegangenen k- ε -Simulation. Die dadurch bestimmten Ergebnisse sind insofern von der Berechnungsvorgeschichte abhängig. Dieses numerische Rauschen muss durch hinreichend große Gradientenschrittweiten überlagert werden.

Abbildung 5.7 illustriert die Charakteristik des Rauschens. Sie zeigt die Verläufe der SFQs bei der abwechselnden und wiederholten Berechnung der Punkte (1.44, 1.92) (Standardparameter) in Farbe und (1.45, 1.92) in Schwarz. Der Abstand zwischen den beiden Punkten entspricht dabei der Gradientenschrittweite 0.01. Jeder farbige Datenpunkt entspricht einer Berechnung des Standardparametersatzes.

5.2 Lokale Optimierung

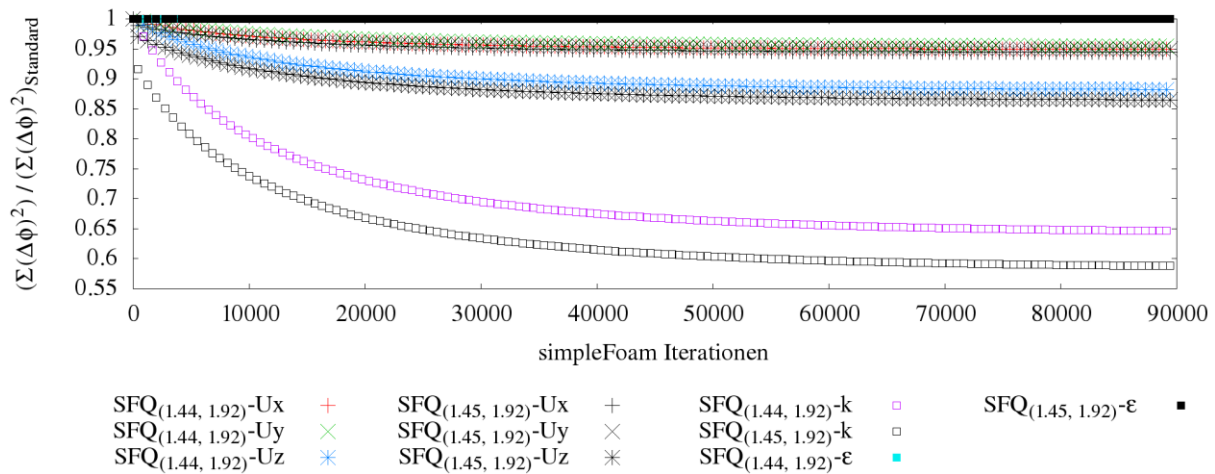


Abbildung 5.7 Einfluss des Rechenweges auf SFQs, 280faches hin- und retour-Rechnen zwischen den Punkten (1.44, 1.92) und (1.45, 1.92)

Es ist gut zu erkennen, dass die SFQs erst im Zuge der Berechnungen gegen einen konstanten Wert konvergieren. Die SFQ für k reduziert sich dabei stark um bis zu 30%. Die vom Framework bestimmten Zielfunktionswerte sind daher quantitativ ungenau. Qualitativ ist kaum eine Änderung zu verzeichnen. Der Wert für SFQ- k am Punkt (1.45, 1.92) liegt unabhängig vom Berechnungsfortschritt um etwa 5%-Punkte unter den Werten von (1.44, 1.92). Die Bildung korrekter Gradienten und insofern die Funktionsweise des Optimierers, ist durch das obige Verhalten daher nicht gefährdet.

Ursache für die Konvergenz der SFQs sind vermutlich die Residuen der berechneten Felder. Diese liegen bei $1E-5$. Bei der Bildung der SFQs, durch Aufsummierung der quadratischen Differenzen in 1.6 Millionen Zellen, gewinnen die Residuen Einfluss im Bereich der signifikanten Nachkommastellen. Bei der wiederholten Berechnung zweier nah aneinander liegenden Punkte, können die Felder sukzessive besser auskonvergieren, da immer bessere Startfelder vorliegen. Damit könnte bei Eignung des Modells der Rückgang der SFQs erklärt werden.

Abbildung 5.8 zeigt die Verläufe der Residuen in folgender Reihenfolge:

- i. Erste Berechnung des Punktes (1.45, 1.92) ausgehend vom Punkt (1.44, 1.92).
- ii. Erste Berechnung des Punktes (1.44, 1.92) ausgehend vom Punkt (1.45, 1.92).
- iii. Letzte Berechnung des Punktes (1.45, 1.92) ausgehend vom Punkt (1.44, 1.92).
- iv. Letzte Berechnung des Punktes (1.44, 1.92) ausgehend vom Punkt (1.45, 1.92).

5.2 Lokale Optimierung

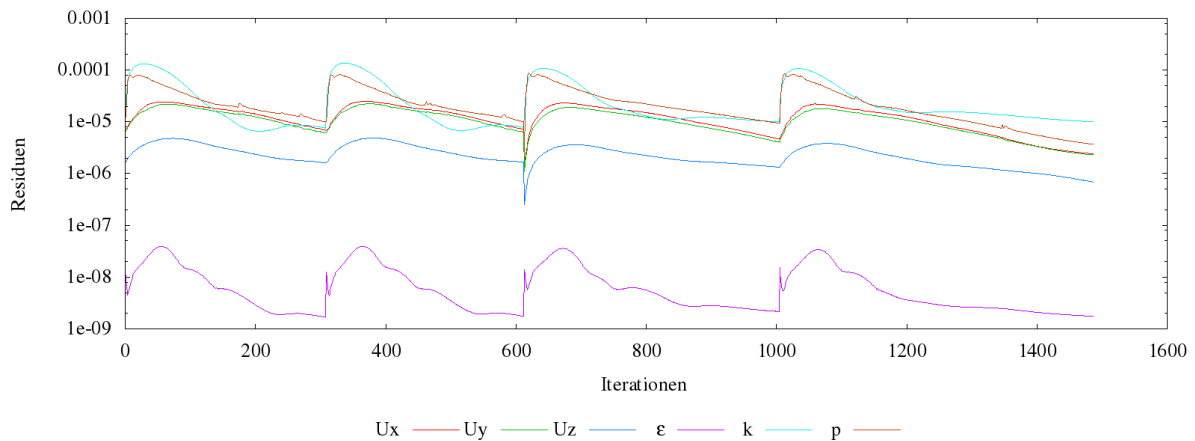


Abbildung 5.8 Änderung der Residuenverläufe bei wiederholter Berechnung zweier nah aneinander liegender Punkte.

Nach Abbildung 5.8 nimmt die Anzahl der Iterationen bis zur Konvergenz der Lösung, mit steigender Anzahl an Wiederholungen zu. Außerdem werden die Verläufe glatter und fallen langsamer. Die Residuen von k , die bei den ersten Berechnungen überschwingen, sinken bei den letzten Berechnungen monoton. Die Höhe der Werte der Residuen bleiben jedoch ungefähr gleich. Dies liegt daran, dass die Berechnung abgebrochen wird, wenn alle Residuen Werte kleiner $1\text{E-}5$ aufweisen.

Bei Verschärfung des Abbruchkriteriums durch Senkung der maximalen Residuen auf $1\text{E-}6$, kommt es ebenfalls zu einer Verringerung der SFQs. Abbildung 5.9 zeigt dies für eine Nachrechnung des Standardparametersatzes. Mit fallenden Residuen und daher exakteren Feldern sinken die SFQs. Dabei treten ähnliche Verläufe der SFQs wie in Abbildung 5.7 auf. In beiden Fällen ist nach etwa 50000 Iterationen eine Reduktion von SFQ- k von 30% zu verzeichnen.

5.2 Lokale Optimierung

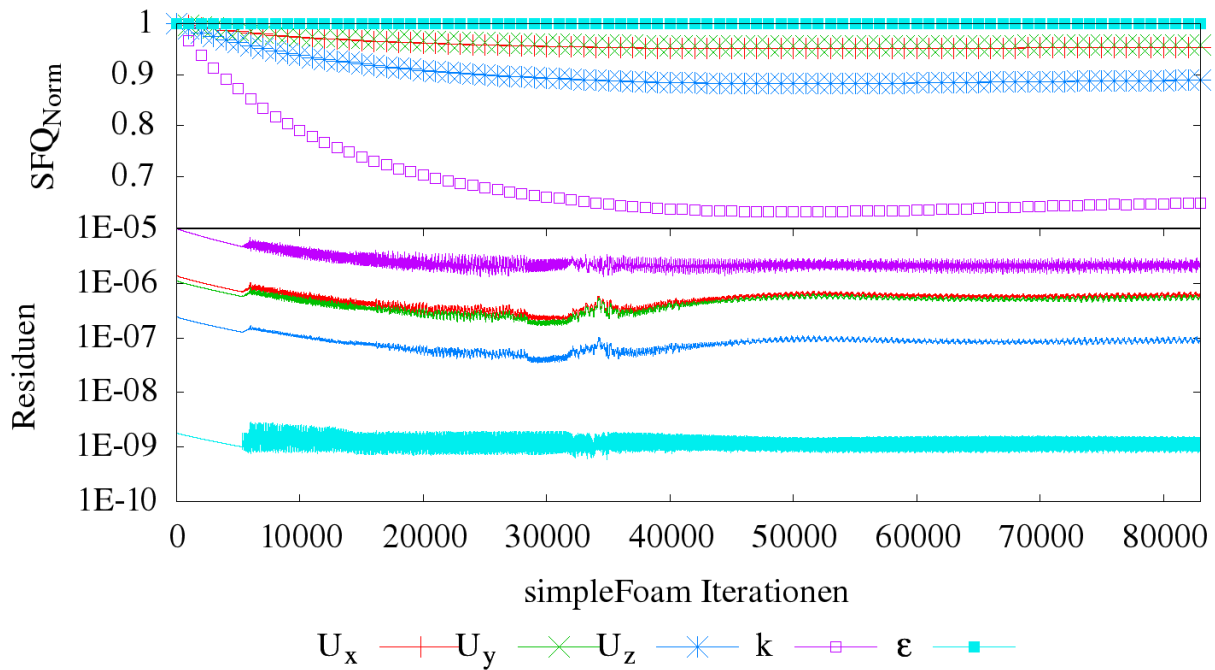


Abbildung 5.9 Verläufe der SFQs für den Standardparametersatz bei Reduktion der Residuen

In Abbildung 5.9 beginnen die Residuen nach etwa 5000 Iterationen zu oszillieren und nach 35000 Iterationen wieder leicht anzusteigen. Dies macht sich schlussendlich auch in einem Anstieg der SFQs ab etwa 50000 Iterationen bemerkbar. Die Residuen von k können das Abbruchkriterium von $1e-6$ nicht erreichen. Dieses Verhalten könnte mit einem feineren Gitter umgangen werden. Berechnungen mit dem k - ϵ -Modell auf dem DNS-Rechengitter weisen nicht oszillierende Residuenverläufe auf. Es bedarf jedoch ca. 18000 zusätzlicher Iterationen, damit alle Residuen Werte kleiner $1E-6$ erreichen. Die Genauigkeit der Felder kann also nur durch erheblichen Rechenaufwand (Vervielfachung der Rechengitterzellen, Verdopplung der Iterationen) um eine Größenordnung reduziert werden. Eine Verschärfung des Abbruchkriteriums stellt also keine geeignete Vorgehensweise zur Unterbindung des Rauschens dar.

Für den Optimierungserfolg ist daher die gewählte Gradientenschrittweite maßgeblich. Die Erstellung von dreidimensionalen Zielfunktionsplots (Abbildung 5.4) ist jedoch aufwendig. Liegen derartige Plots nicht vor, können zu kleine Gradientenschrittweiten an den SFQs mit vergleichbaren Verläufen, ähnlich denen in Abbildung 5.10, erkannt werden. Hierbei konvergiert der Optimierer nicht gegen das Minimum, sondern bewegt sich von diesem aufgrund verrauschter Gradienten weg.

Abschließend ist zu sagen, dass sich die Schrittweite von $1E-2$ für alle im Zuge dieser Arbeit vorgenommenen Optimierungen bewährt hat. Sie kann im File `opti-script.py` unter

5.2 Lokale Optimierung

der Option eps eingetragen werden. Hohe Schrittweiten gehen jedoch, aufgrund der längeren Berechnungsdauer von weiter auseinanderliegenden Punkten, mit einem Effizienzverlust einher.

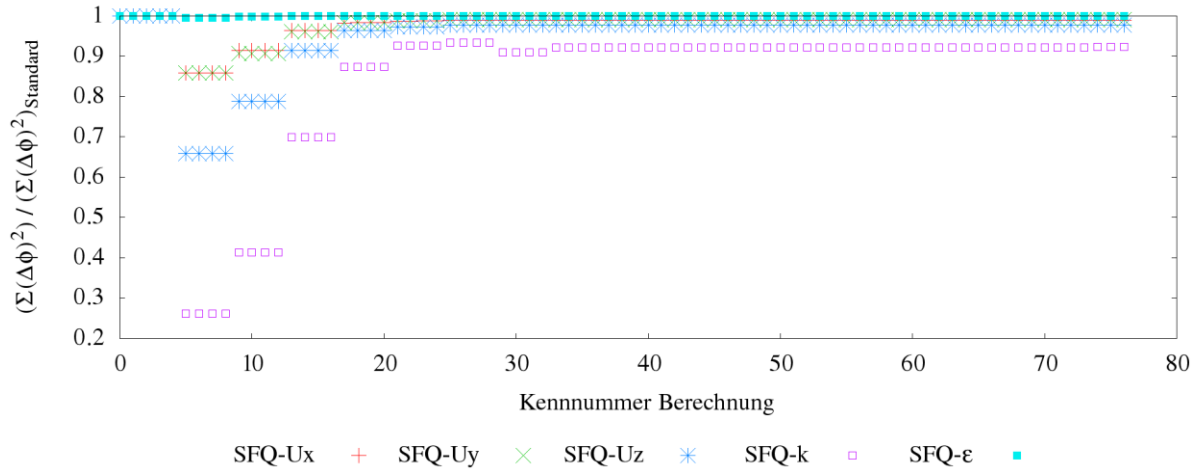


Abbildung 5.10 Verlauf der SFQs bei ungeeigneter Schrittweitenwahl.

5.2.4 Verhalten bei Instabilität

Da nach den vorliegenden Berechnungen das Minimum an der Stabilitätsgrenze liegt, wurde das Verhalten des Frameworks auch in der Nähe des instabilen Bereichs getestet (Abbildung 5.11).

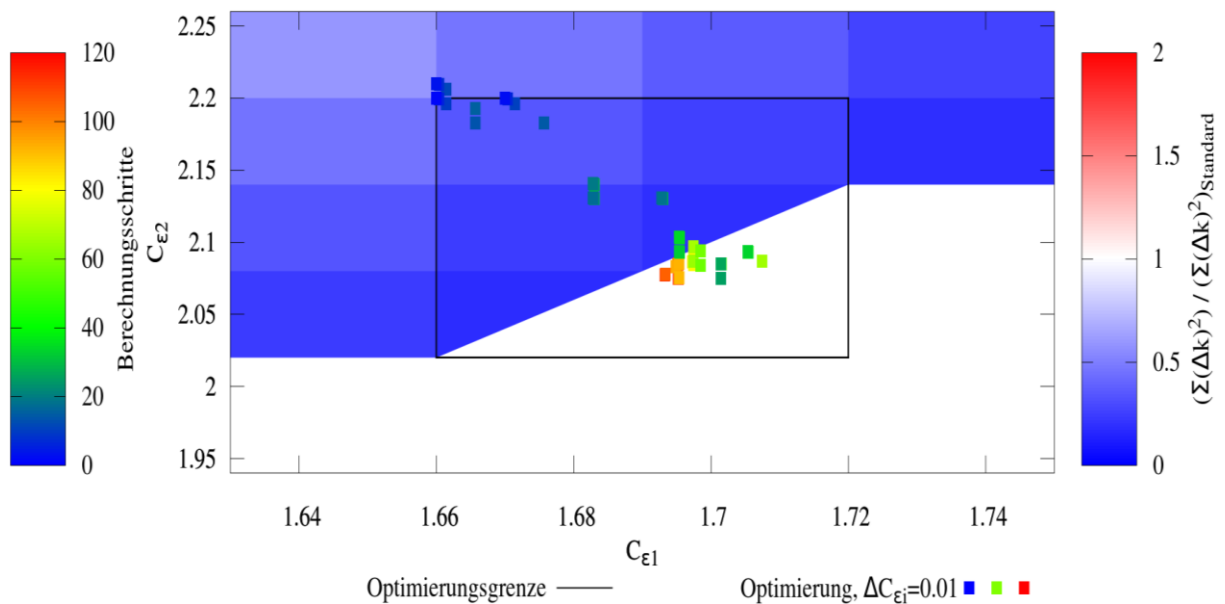


Abbildung 5.11 Test des Frameworks in Bereichen nahe der Stabilitätsgrenze

Auch im instabilen Bereich ist eine Optimierung mit dem entwickelten Framework möglich. Der Optimierer konvergiert gegen das an der Stabilitätsgrenze liegende Minimum.

5.2 Lokale Optimierung

Für instabile Punkte nahe der Stabilitätsgrenze ist dieses aber nur mäßig effizient. Die Berechnung wird hier durch das Erreichen der Maximalanzahl an Iterationsschritten abgebrochen und die erhaltene Lösung nicht verwendet. Stattdessen werden dem Optimierer hohe, konstante Strafwerte übergeben. Das Abbruchkriterium ist also unverhältnismäßig aufwendig.

Abbildung 5.12 gibt die SFQs im Laufe der in Abbildung 5.9 dargestellten Optimierung an. Die SFQs werden dabei über der Anzahl der von OpenFOAM berechneten Zeitschritte aufgetragen. Der Übersichtlichkeit halber sind in Abbildung 5.12 die Strafwerte aller SFQs auf einen Wert von 1.2 gedämpft. Es ist gut ersichtlich, dass viele der Iterationsschritte auf instabile und daher nicht verarbeitbare Lösungen entfallen. Deren Berechnung beansprucht die maximale Berechnungsdauer.

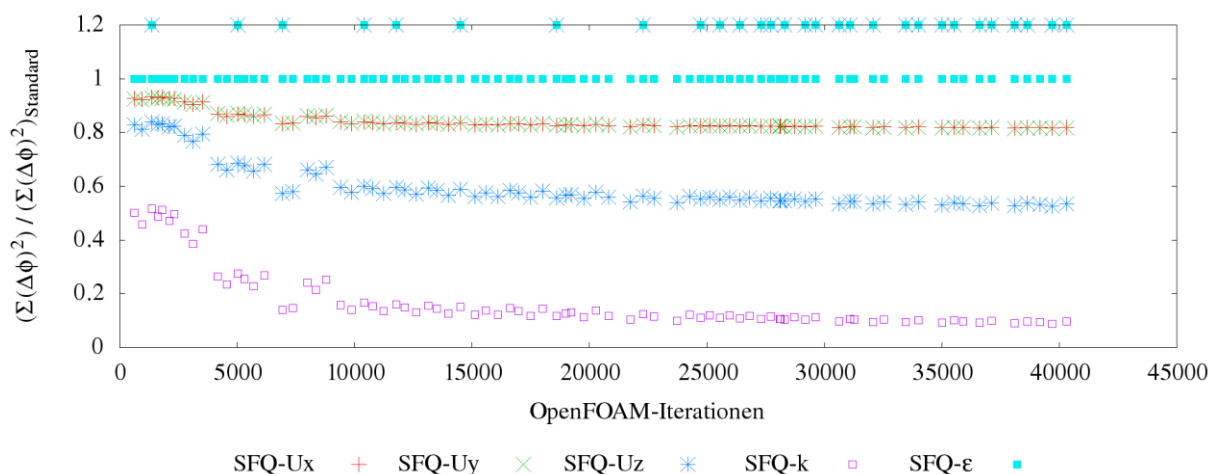


Abbildung 5.12 Verläufe der SFQs bei einer Optimierung nahe der Stabilitätsgrenze, hohe Strafwerte wurden durch den Wert 1.2 ersetzt.

Die einfachste Methode zur Effizienzsteigerung ist daher die manuelle Anpassung der maximalen Iterationsanzahl. Wie in Abbildung 4.15 zu sehen, kann frühzeitig anhand der Residuenverläufe eine instabile Lösung erkannt werden. Des Weiteren reduzieren sich im Zuge einer Optimierung die Iterationsschrittweiten, da der Optimierer an das Minimum konvergiert. Deswegen benötigen konvergente Lösungen sukzessive weniger Iterationsschritte. Daher ist bei immer schneller konvergierenden bzw. offensichtlich instabilen Lösungen eine Anpassung der maximalen Iterationsanzahl sinnvoll. Diese kann im *controlDict* unter der Einstellung *writeIntervall* vorgenommen werden.

Eine komplexere Methode zur Effizienzsteigerung ist im Framework implementiert und beruht auf folgender Annahme: Konvergiert ein Punkt nicht, so kann davon ausgegangen werden, dass die benachbarten Punkte zur Berechnung der Gradienten, ebenfalls nicht

5.2 Lokale Optimierung

konvergieren. Das Framework überprüft daher, ob die erste Zielfunktionsberechnung (i) einer Optimierungsiteration (i-vi, Seite 78) konvergiert. Ist dies nicht der Fall, werden alle weiteren Berechnungen (ii-vi) übersprungen und Strafwerte überreicht. Diese Funktion ist der Übersichtlichkeit halber nicht im UML-Diagramm in Abbildung 4.24 eingezeichnet. Durch dieses Vorgehen kann die Gradientenberechnung eines instabilen Punktes, um das 4fache bei zwei oder um das 7fache bei fünf Parametern beschleunigt werden. Problematisch wird diese Shortcut-Methode, wenn durch die Größe der Gradientenschrittweite ein Wiedereintritt in den stabilen Bereich möglich wäre. Diese Gefahr besteht vor allem bei C_μ , dessen Größenordnung der Größenordnung der gewählten Gradientenschrittweite (0.01) entspricht. Die so verlorengegangene Information fehlt dem Optimierer bei der Annäherung an die Stabilitätsgrenze. Durch die Stabilität des L-BFGS-B-Optimierers haben solche Fehlabschätzungen bis dato zu keinen bemerkbaren Problemen geführt. Außerdem überschreibt L-BFGS-B, aufgrund seiner Beschaffenheit (Kapitel 3.1.3.2), die Zielfunktionsbestimmungen weiter zurückliegender Optimierungsiterationen. In Kombination mit der Shortcut-Methode, welche Fehleinschätzungen einbringen kann, ist diese Vorgehensweise wünschenswert.

Wie in Kapitel 5.2.2 diskutiert, konnte eine mögliche theoretisch fundierte Stabilitätsgrenze gefunden werden. Nach Pope (2000) könnte eine Anpassung von σ_ε den stabilen Bereich erweitern. Abbildung 5.13 zeigt vorgenommene Kontrollberechnungen an 15 Punkten (Quadrate). An diesen wurde σ_ε an die vorliegenden $C_{\varepsilon 1}$ und $C_{\varepsilon 2}$, sowie die konstanten κ und C_μ Werte nach Gleichung 2.5.69 angepasst. Durch die vorgenommene Anpassung konnte nur ein zusätzlicher Punkt nahe der Stabilitätsgrenze zur Konvergenz gebracht werden (grün). Die Anpassung ist daher nur bedingt sinnvoll. Gleichung 2.5.69 ist infolgedessen kein allgemeingültiges Konvergenzkriterium.

5.3 Auswahl der Optimierungsvariablen

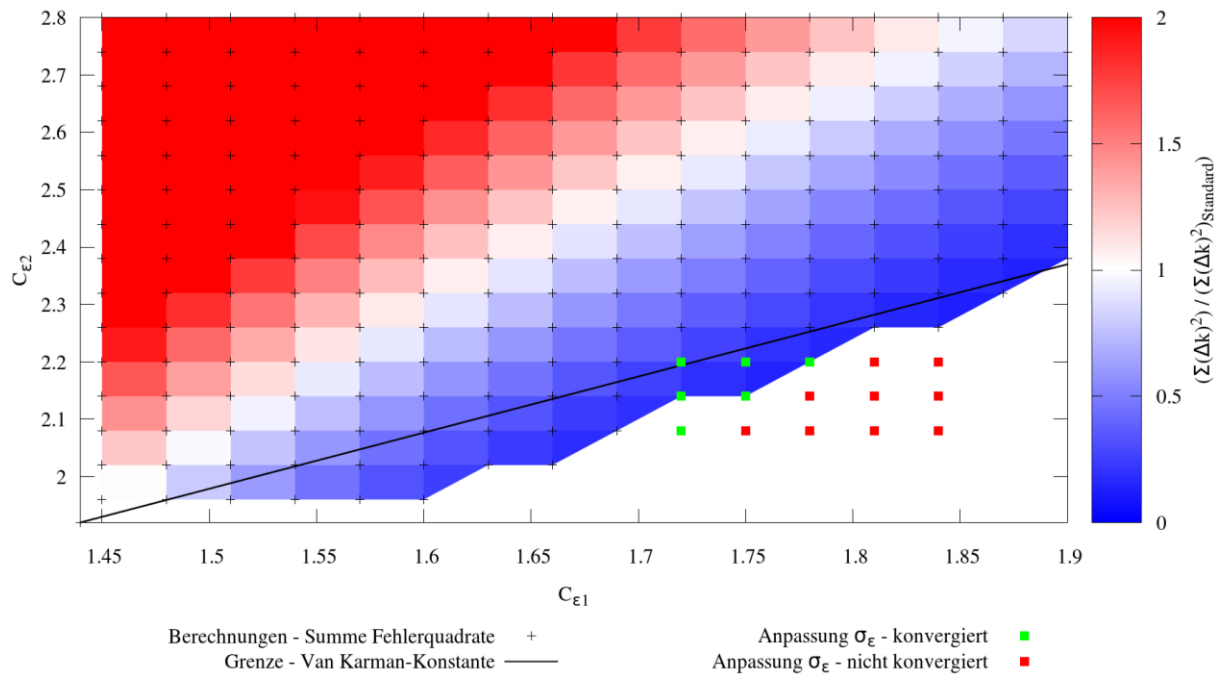


Abbildung 5.13 Anpassung von σ_ϵ zur Erweiterung des stabilen Bereiches

5.3 Auswahl der Optimierungsvariablen

In diesem Kapitel soll die Eignung der fünf k- ϵ -Modellparameter als Optimierungsvariablen diskutiert und ihr Einfluss auf das Optimierungsergebnis veranschaulicht werden.

Eine Reduktion auf zwei Optimierungsvariablen ermöglicht eine dreidimensionale Darstellung der Zielfunktion. Aus diesen können mehr Informationen als durch den Verlauf einer Optimierung gewonnen werden. Wie in Kapitel 5.2.1 diskutiert, ist die Reduktion auf die Parameter $C_{\epsilon 1}$ und $C_{\epsilon 2}$ naheliegend. Abbildung 5.14 zeigt den Verlauf der SFQs im Laufe einer Optimierung der Parameter $C_{\epsilon 1}$ und $C_{\epsilon 2}$. Die jeweilige Parameterobergrenze betrug 190% des entsprechenden Parameterstandardwertes. Die Untergrenze entsprach 10% des dazugehörigen Standardwertes. Die Optimierung nahm etwa 2 Tage in Anspruch und übergab die Berechnung der Gradienten an instabilen Punkten (siehe Kapitel 5.2.4).

5.3 Auswahl der Optimierungsvariablen

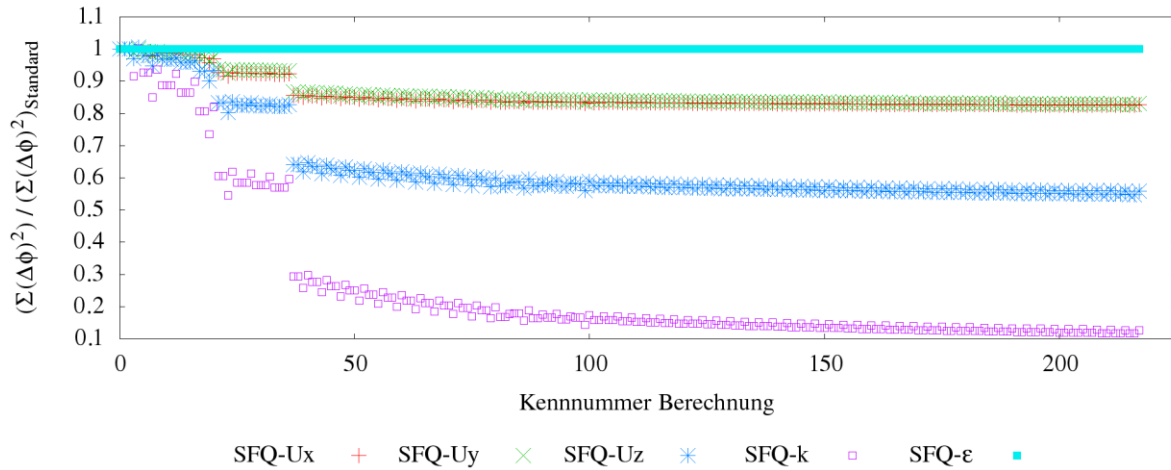


Abbildung 5.14 Optimierung von $C_{\varepsilon 1}$ und $C_{\varepsilon 2}$, Optimierungsbereich: $P_{i,Standard}(1 \pm 0.9)$

Die Zielfunktion (SFQ-k) sinkt im Zuge der Optimierung um 90%. Die SFQ von ε weist wiederum keine Sensitivität auf und verbleibt auf den normierten Wert von 1. Der Anteil am Geschwindigkeitsbetrag ist für die Geschwindigkeitskomponente U_z am höchsten. Ihre SFQ geht um etwa 45% zurück. Die SFQs der restlichen Geschwindigkeitskomponenten U_x und U_y können um etwa 20% reduziert werden.

Im Allgemeinen ist wieder die Korrelation der einzelnen SFQs zu beobachten. Die SFQs fallen die ersten 40 Berechnungen stark ab. Dieser Abfall reduziert sich sukzessive mit Fortschritt der Optimierung und geht schlussendlich ab 100 Berechnungen in einen glatten konvergenten Verlauf über. Die am Anfang dieses Bereiches vorliegenden $C_{\varepsilon 1}$ und $C_{\varepsilon 2}$ Werte unterscheiden sich, nach den log-Files, nur in der achten bzw. siebten Nachkommastelle vom Endergebnis. In diesem Bereich verursacht daher vermutlich nur noch das Rauschen den Abfall der SFQs. Die Optimierung könnte daher nach 150 Berechnungen abgebrochen werden.

Abbildung 5.15 zeigt den Verlauf der SFQs einer Optimierung aller Parameter. Für diese wurden die Parametergrenzen analog wie in der $C_{\varepsilon 1}$ - $C_{\varepsilon 2}$ -Optimierung definiert. Diese Optimierung nahm etwa 5.5 Tage in Anspruch, da die Berechnung der Gradienten an instabilen Punkten nicht übersprungen wurde. Soll die Effizienz der beiden Optimierungen diskutiert werden, ist es daher sinnvoller, die Anzahl der untersuchten instabilen Punkte, sowie die Anzahl der konvergierten Berechnungen gegenüber zu stellen.

Auch bei der Optimierung aller Parameter ist der starke Abfall der SFQs am Anfang und der darauffolgende Übergang in den konvergenten Bereich zu erkennen. Aus den log-Files

5.3 Auswahl der Optimierungsvariablen

geht für den konvergenten Bereich ebenfalls eine geringe Änderung der Parameter hervor, das Rauschen dominiert wieder den Abfall der SFQs. Der beste Parametersatz wird in etwa bei Berechnung 150 gefunden. Hier könnte die Optimierung abgebrochen werden.

Ab Berechnung 160 kommt es zur Irritation des Optimierers und einem damit einhergehenden starken Anstieg der SFQs. Die log-Files zeigen, dass der Optimierer kurz vor dem Anstieg den Parameter C_μ durchvariiert und vom gängigen Berechnungsschema (i-vi, Seite 78) abweicht. Diese Irritation dürfte durch die Größenordnung von C_μ verursacht werden. Letztere entspricht der Größenordnung der Gradientenschrittweite. Dadurch kann bei der Gradientenbildung mit der Variation von C_μ der stabile Bereich verlassen werden. Der somit berechnete Gradient zwischen einem stabilen Punkt und einem instabilen Punkt fällt daher sehr steil aus. Da in dieser Phase der Optimierung die Parameter kaum noch variiert werden, liegen C_μ -Gradienten von benachbarten Punkten, für die keine Strafwerte herangezogen werden mussten, vor. Sie sind daher weitaus kleiner. Die Diskrepanz in der Stärke der gebildeten Gradienten kann als Ursache für die Irritation des Optimierers angesehen werden. Die SFQs konvergieren jedoch nach dem Anstieg schnell wieder in Richtung Minimum. Der L-BFGS-B-Algorithmus erweist sich in diesem Fall als stabil.

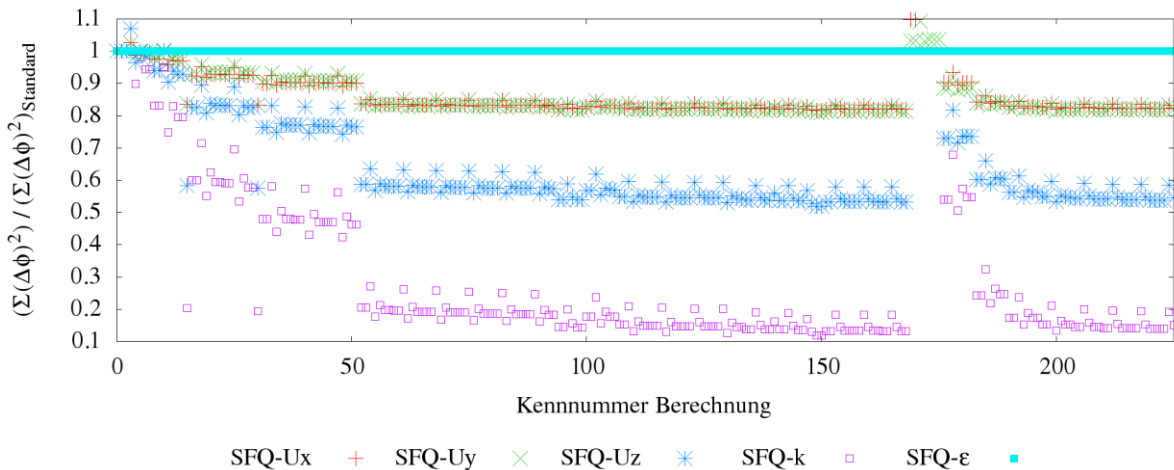


Abbildung 5.15 Optimierung aller Parameter, Optimierungsbereich: $P_{i,Standard}(1 \pm 0.9)$

Die Minimalwerte der SFQs der Optimierung aller Parameter unterscheiden sich kaum von denen der $C_{\varepsilon 1}$ - $C_{\varepsilon 2}$ -Optimierung. In beiden Optimierungen konnten, unabhängig von der Parameterwahl, gute Ergebnisse innerhalb von 150 konvergierten Zielfunktionsbestimmungen erreicht werden. Außerdem wurden in beiden Fällen 10 instabile Punkte vom Optimierer untersucht. Die Berechnung dieser Punkte ist aufgrund

5.3 Auswahl der Optimierungsvariablen

des Abbruchkriteriums sehr aufwendig. Bei der Optimierung aller Parameter ist zudem die Berechnung von drei weiteren Gradienten notwendig. Der dadurch entstehende Mehraufwand ist aufgrund der hinreichend kleinen Gradientenschrittweite jedoch vernachlässigbar. Da sich die Anzahl der instabilen Punkte in beiden Optimierungen gleicht, ist aus Effizienz Gründen eine Reduktion der Parameter nicht notwendig.

Abbildung 5.16 zeigt den Verlauf einer Optimierung aller Parameter mit enger gesetzten Grenzen. Diese beträgt anstatt 90% $\pm 5\%$ des Standardwertes. Die Verläufe der SFQs sind mit den Verläufen der Optimierungen bei weiten Grenzen, sowohl qualitativ als auch quantitativ, vergleichbar. Durch die eng gesetzten Grenzen konnte die Untersuchung instabiler Punkte und damit die Irritation des Optimierers vermieden werden. Sie ist mit einer Dauer von 1.5 Tagen und komparablen Ergebnissen die effizienteste Optimierung.

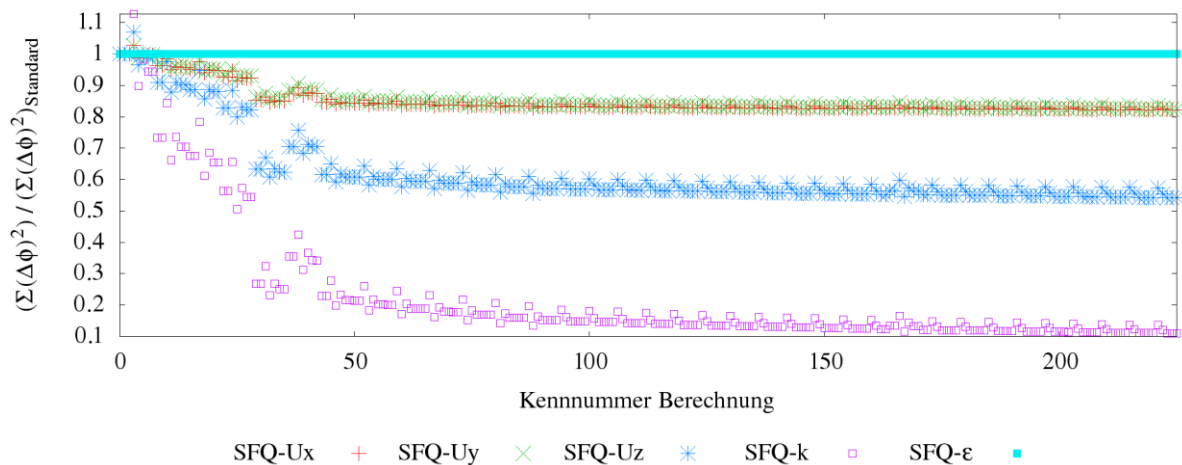


Abbildung 5.16 Optimierung aller Parameter, Optimierungsbereich: $P_{i, \text{Standard}}(1 \pm 0.05)$

Tabelle 5.4 zeigt die gewonnenen optimierten Parametersätze der drei diskutierten Optimierungen. In allen Optimierungen wurde $C_{\varepsilon 1}$ erhöht und $C_{\varepsilon 2}$ gesenkt. Dies deckt sich mit den Ergebnissen der Sensitivitätsanalyse (Kapitel 4.1.5.1), sowie mit dem Plot der Zielfunktion in Abbildung 5.5. $C_{\varepsilon 2}$ wird dabei näher an den durch Gleichung 2.5.51 bestimmten Wert von 1.77 gerückt. Bei der Optimierung aller Parameter wurde C_{μ} in beiden Fällen gesenkt. Auch dieses Ergebnis stimmt mit den in der Sensitivitätsanalyse gewonnenen Erkenntnissen überein. Eine mögliche Erklärung liefert Abbildung 2.5. Sie zeigt, dass C_{μ} in Wandnähe mit einem Wert von 0.09 überschätzt wird. Die Absenkung von C_{μ} könnte durch einen Einfluss der Wände auf große Teile der untersuchten Strömung erklärt werden, denn Punkte im turbulenten Bereich der Strömung sind maximal 1mm von der Wand entfernt. Die Parameter σ_{ε} und σ_k wurden in beiden Fällen

5.4 Optimierte Strömungsfelder

erhöht. Ihr Einfluss auf die Strömungsfelder war mittels der Sensitivitätsanalyse nur schwer abzuschätzen.

Tabelle 5.4 Vergleich der optimierten Parameter.

Parameter	Grenzen	C_μ	$C_{\varepsilon 1}$	$C_{\varepsilon 2}$	σ_k	σ_ε
Keine	—	0.09	1.44	1.92	1.00	1.30
$C_{\varepsilon 1}, C_{\varepsilon 2}$	$P_{i,Standard}(1 \pm 0.9)$	0.09	1.4923976	1.8095522	1.00	1.30
Alle	$P_{i,Standard}(1 \pm 0.9)$	0.048090761	1.4828751	1.9010110	1.0139453	1.3039629
Alle	$P_{i,Standard}(1 \pm 0.05)$	0.0855*	1.512*	1.8447824	1.050*	1.365*
Anm.:	*optimierter Parameter ist Grenzwert					

Der Einfluss der turbulenten Prandtlzahlen, mit den geringsten Sensitivitäten (siehe Kapitel 4.1.5.2), geht bei der Optimierung aller Parameter mit weiten Grenzen unter. Sie werden kaum variiert. Dennoch können sie für künftige Optimierungen nicht ausgeschlossen werden. Bei der Optimierung mit eng gesetzten Grenzen wird C_μ , der sensitivste Parameter, gesenkt und an seiner Untergrenze festgehalten. Dadurch kommt es zu stärkeren Variation der anderen, weniger sensitiven Parameter. Die bei weiten Grenzen kaum variierten Parameter σ_ε und σ_k , werden bei engen Grenzen um 5% gesteigert und erreichen ihre Obergrenze. Sie bleiben daher als potentielle Optimierungsvariablen bestehen.

5.4 Optimierte Strömungsfelder

Abschließend sollen die Plots der optimierten Strömungsfelder gezeigt werden. Da in den drei diskutierten Optimierungen die Güte der gewonnenen Felder kaum Unterschiede aufweist, beschränkt sich die Darstellung der Plots auf die Optimierung mit eng gesetzten Grenzen. Deren Felder sind am besten auskonvergiert und die SFQs insofern am niedrigsten. Sie werden in diesem Kapitel den Feldern bei Standardparametern und den Feldern der DNS gegenübergestellt.

Die Verbesserung des ε -Feldes durch Parameteranpassung ist nach den vorliegenden Ergebnissen nicht möglich. Dies geht aus den Verläufen der SFQs hervor und kann durch die Plots in Abbildung 5.17 bestätigt werden. Die durch das k- ε -Modell berechneten Felder unterscheiden sich trotz Parameteranpassung kaum. Das Feld der DNS kann daher weder qualitativ, noch quantitativ besser wiedergegeben werden.

5.4 Optimierte Strömungsfelder

Letzteres weist weitaus höhere Werte in Wandnähe auf. Die Bereiche nahe der Wände werden jedoch durch das Rechengitter hoch aufgelöst. In Kombination mit der Größenordnung des ε -Feldes, entstehen an diesen Stellen hohe Differenzen zwischen dem DNS- und dem k- ε -Feld, die schlussendlich zu den hohen Werten der nicht normierten SFQs von ε , in der Größenordnung von $E+21$, führen.

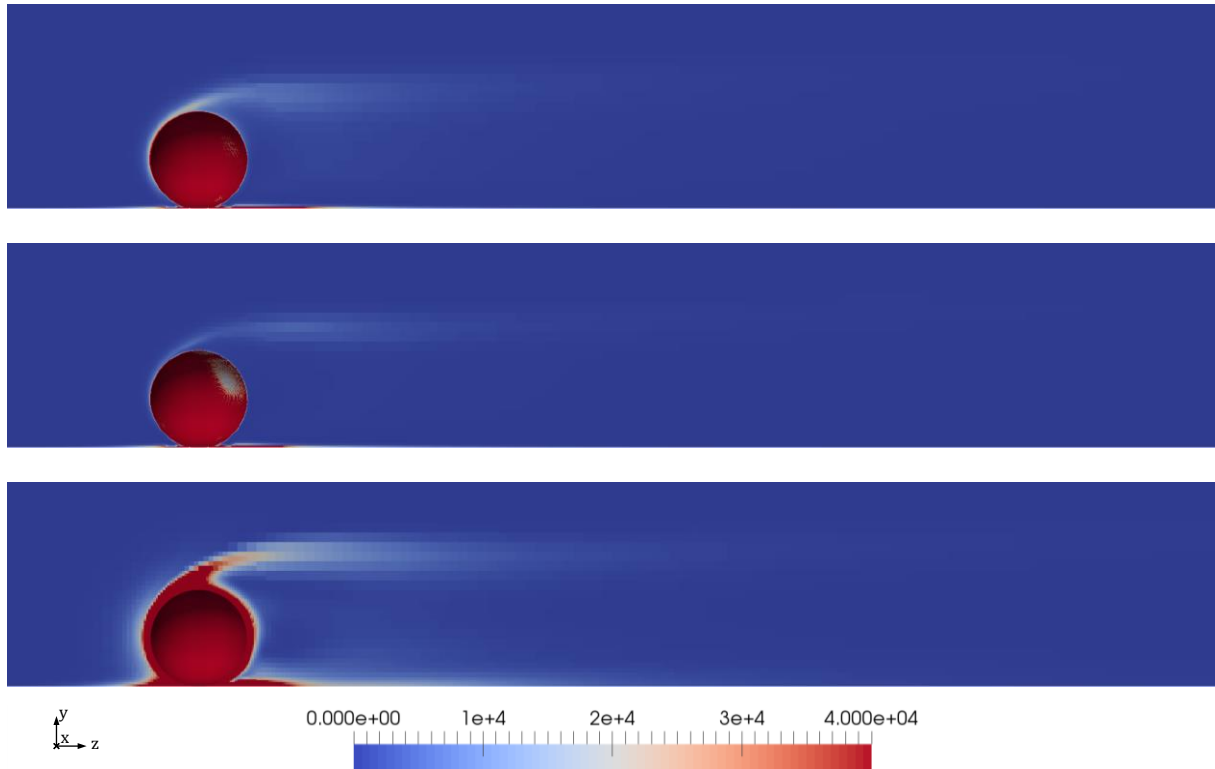


Abbildung 5.17 Optimiertes Feld von ε in m^2/s^3 : oben Standardparameter, mittig optimierte Parameter, unten DNS

Die SFQs der Felder $\langle U_x \rangle$ und $\langle U_y \rangle$ konnten durch die lokale Optimierung nur um 10% reduziert werden. Diese geringe Verbesserung ist in den Plots der Felder nur schwer ersichtlich. Deshalb beschränkt sich diese Arbeit auf die Darstellung des Geschwindigkeitsbetrages. Wie in Abbildung 5.18 ersichtlich, kann das Geschwindigkeitsfeld in seinem qualitativen Verlauf durch Parameteranpassung an die DNS-Lösung angeglichen werden. Quantitativ herrscht gute Übereinstimmung.

5.4 Optimierte Strömungsfelder

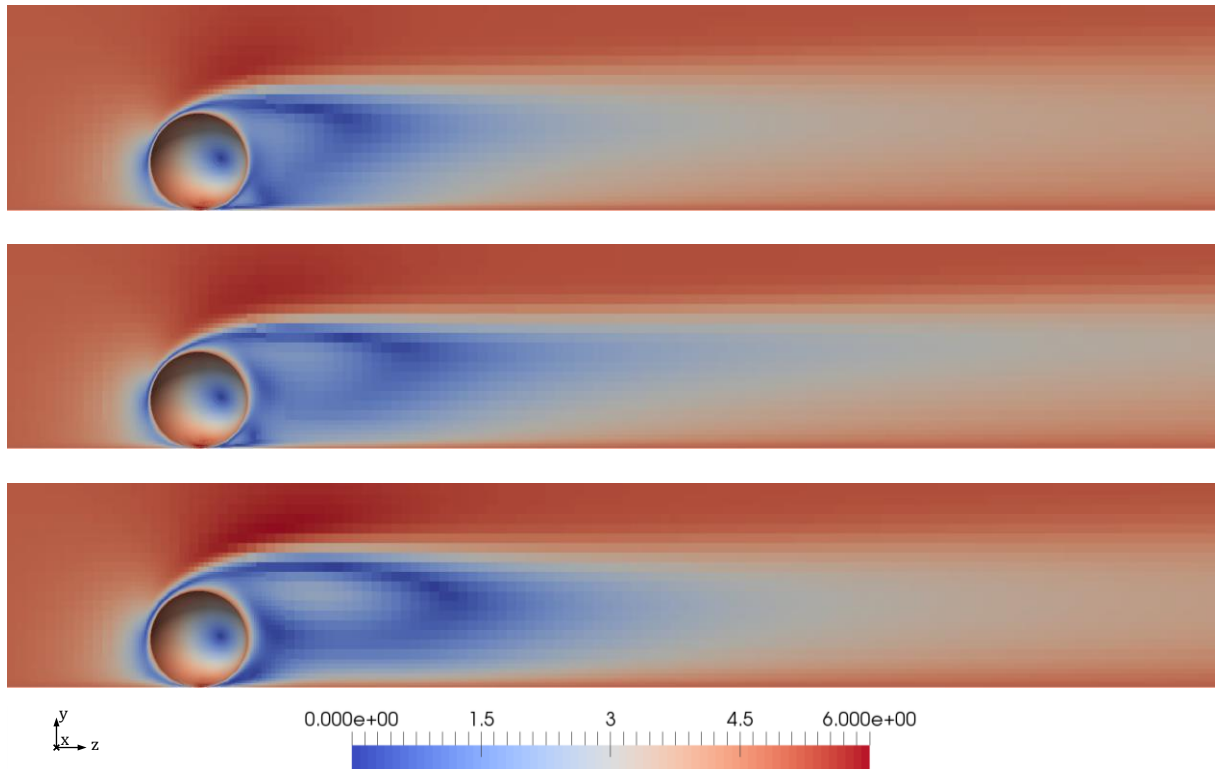


Abbildung 5.18 Optimierter Betrag des gemittelten Geschwindigkeitsfeldes $\langle \mathbf{U} \rangle$ in m/s : oben Standardparameter, mittig optimierte Parameter, unten DNS

Die SFQ von k konnte um ca. 90% und daher am stärksten reduziert werden. Die quantitative Fehleinschätzung des k -Feldes durch die Standardparameter wird durch die Parameteranpassung gut korrigiert. Aus Abbildung 5.19 geht jedoch ein verbleibender qualitativer Unterschied hervor. Der Bereich mit hohen k -Werten setzt in der DNS-Lösung nicht direkt an der Kugel an. Dies ist hingegen beim optimierten Feld der Fall.

5.5 Übertragbarkeit

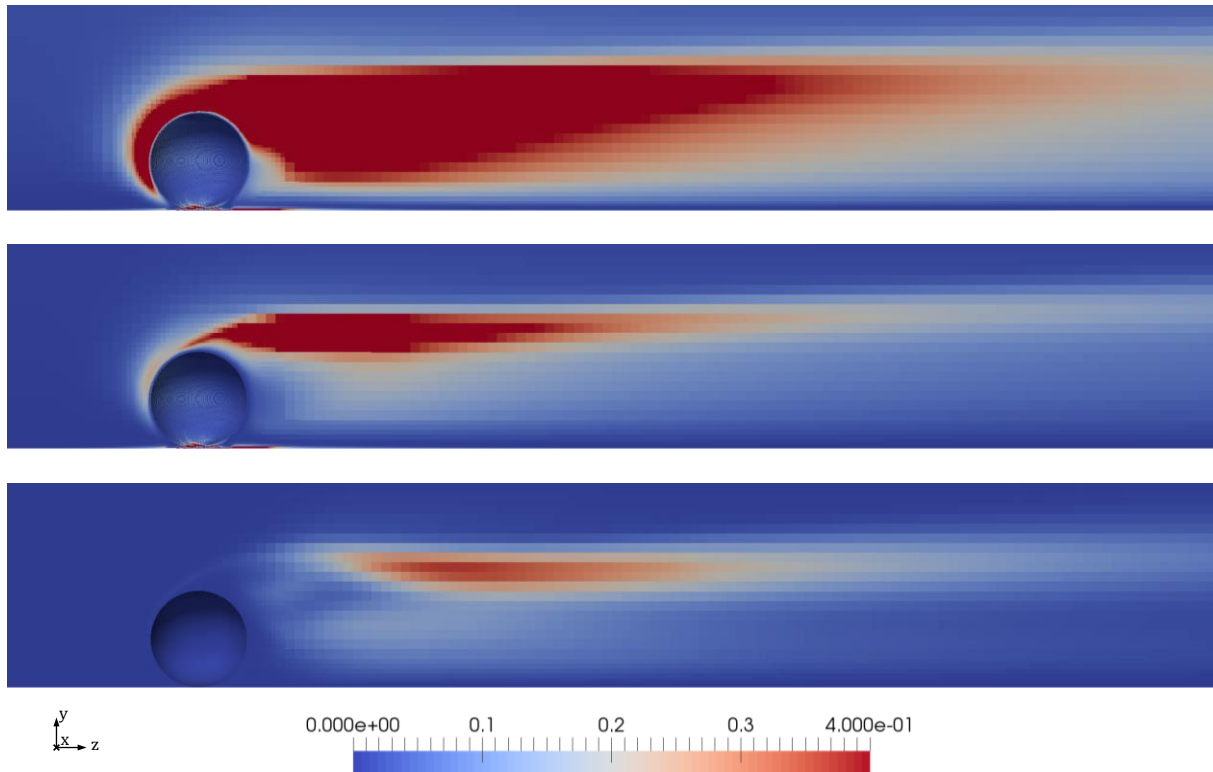


Abbildung 5.19 Optimiertes Feld der turbulenten kinetischen Energie k in m^2/s^2 : oben Standardparameter, mittig optimierte Parameter, unten DNS

5.5 Übertragbarkeit

Wie in Kapitel 2.5.3.6 diskutiert, sind die Standardparameter für den universellen Einsatz am besten geeignet. Anpassungen der Parameter an eine bestimmte Strömung, zur besseren Wiedergabe der Strömungsfelder, gehen immer mit einer Spezialisierung auf die Strömung einher. Die angepassten Parameter sind dann nur für ähnliche Strömungen besser geeignet als die Standardparameter. Die Kenntnis des Übergangs, an dem die Standardparameter wieder eine geeignetere Abschätzung der Felder bereitstellen, ist daher wesentlich, wenn eine verbesserte Wiedergabe der Strömungsfelder garantiert werden soll. Die Ermittlung dieses Gültigkeitsüberganges ist jedoch aufwendig, da hierbei DNS-Lösungen für die sukzessiv stärker abgewandelte Strömung bereitliegen müssen. Sie sprengt daher den Rahmen dieser Masterarbeit.

In Zuge dieser Arbeit konnte die Übertragbarkeit von optimierten Parametern deshalb nur stichprobenartig an einer leicht abgewandelten Strömung überprüft werden. Sie unterscheidet sich im Wesentlichen vom Standard-Case anhand zweier Parameter. Der Kugeldurchmesser d_K wurde um 0.5mm erhöht und beträgt 1.5mm. Die Eintrittsgeschwindigkeit U_{ein} wurde von 5m/s auf 10m/s gesteigert. Die Kugel-Reynoldszahl Re_K wurde insofern verdreifacht und die Strömung damit turbulenter.

5.5 Übertragbarkeit

$$Re_K = U_{ein} d_K / \nu \quad 5.5.1$$

Eine Analyse der Geschwindigkeitsplots der DNS-Lösungen bestätigt dies. Die DNS-Lösung des Standard-Cases lässt noch leichte Regelmäßigkeiten im zeitlichen Verlauf des Geschwindigkeitsfeldes erkennen. Die Strömung befindet sich daher vermutlich im turbulenten Übergangsbereich. Die Geschwindigkeitsplots des abgewandelten Cases weisen zeitlich unregelmäßige Wirbelstrukturen auf. Die Strömung scheint einen höheren Grad an aufkommender Turbulenz aufzuweisen. Da für diese Beobachtung die Begutachtung vieler Geschwindigkeitsplots notwendig ist, soll von einer Darstellung der Plots abgesehen werden.

Der höhere Grad an aufkommender Turbulenz kann auch anhand eines Vergleiches der k- ϵ -Modell- mit den DNS-Lösungen nachvollzogen werden. Tabelle 5.5 stellt die SFQs der beiden Cases gegenüber. Sie wurden mit den Standardparametern gebildet und zur Vergleichbarkeit, durch die Anzahl der Zellen des jeweilig verwendeten Gitters, dividiert. Nach Tabelle 5.5, kann das Geschwindigkeitsfeld im turbulenten Fall durch das k- ϵ -Modell besser wiedergegeben werden. Die SFQs für die Komponenten U_y und U_z sinken um eine Größenordnung und SFQ- U_x halbiert sich. Die SFQs der zwei modellierten Transportgrößen k und ϵ nehmen im turbulenten Fall erheblich zu. SFQ- ϵ steigt um eine, SFQ- k um drei Größenordnungen.

Tabelle 5.5 Vergleich der SFQs bei Standardparametern, normiert durch die Anzahl der Rechengitterzellen

Case	SFQ- U_x	SFQ- U_y	SFQ- U_z	SFQ- k	SFQ- ϵ
Re_K niedrig	3.86E-02	1.22E-01	2.80E-01	4.61E-02	2.98E+15
Re_K hoch	1.35E-02	1.66E-02	4.20E-02	1.87E+01	1.72E+16

Die Diskrepanz der SFQs zwischen den beiden Fällen um Größenordnungen lässt vermuten, dass sich beide Strömungen grundlegend anders verhalten. Berechnungen einzelner $C_{\epsilon 1}$ - $C_{\epsilon 2}$ -Kombinationen für den Re_K -hoch-Fall bestätigen diese Vermutung. Zur Überprüfung der Übertragbarkeit wurden die SFQs der berechneten Kombinationen mit den SFQs der Standardparameter des turbulenten Falles normiert. Für alle stichprobenartig untersuchten Kombinationen lagen die normierten SFQs bei Werten größer 1. Die untersuchten $C_{\epsilon 1}$ - $C_{\epsilon 2}$ -Kombinationen gaben daher im Re_K -hoch-Fall die Felder schlechter als der Standardparametersatz wieder. Letzteres gilt sowohl für $C_{\epsilon 1}$ - $C_{\epsilon 2}$ -Kombinationen die im Re_K -niedrig-Fall Verbesserungen als auch Verschlechterungen der Strömungsfelder bewirkten. Dies deutet für den Re_K -hoch-Fall auf ein Optimum nahe

5.6 Eignung k - ε -Model

- i. die SFQs korrelieren (Abbildung 5.3)
- ii. die SFQs fallen beim Auskonvergieren der Strömungsfelder (Abbildung 5.9)
- iii. für die Benchmark-Strömungen hergeleitete theoretische Zusammenhänge finden sich in den SFQs wieder (Abbildung 5.5)
- iv. die untersuchte Strömung weist qualitativ ähnliche Sensitivitäten wie die Benchmark-Strömungen auf (Kapitel 4.1.5.1)

6 Ausblick

6.1 Hammersley-Verteilung

Die Berechnung von Hammersley-Punkten, besonders mit weit gesetzten Parametergrenzen, ist sehr aufwendig (Kapitel 5.1). Sie sollte daher nur dann vorgenommen werden, wenn die Zielfunktion mehrere, unterschiedlich stark ausgeprägte lokale Minima aufweist. Bei Reduktion auf zwei Optimierungsvariablen können lokale Minima durch eine dreidimensionale Darstellung der Zielfunktion ausgeschlossen werden. Durch mehrmalige, lokale Optimierung mit unterschiedlichen Startwerten ist eine Detektion lokaler Minima ebenfalls möglich. Diese Option besteht auch bei Benutzung von mehr als zwei Optimierungsvariablen.

6.2 Verhalten bei Instabilität

Zur Verbesserung des Verhaltens bei Instabilität kann sowohl am Framework als auch am Optimierer angesetzt werden.

Wie in Abbildung 4.15 ersichtlich, kann relativ eindeutig anhand der Residuenverläufe eine instabile Lösung erkannt werden. Ein Programm, das die Residuen ausliest, Instabilität erkennt und in diesem Fall die Berechnung stoppt, würde die Effizienz des Frameworks signifikant steigern.

Außerdem könnte der Optimierer auf andere Art und Weise über die Instabilität informiert werden. In der derzeitigen Version des Frameworks wird dem Optimierer ein Strafwert übergeben. Mit diesem wird indirekt die Instabilität der Lösung vermittelt. Dabei wird jedoch der Verlauf der Zielfunktion verfälscht. Der Einfluss der Strafwerthöhe auf den Ausgang der Optimierung müsste deshalb untersucht werden. Um einen etwaigen Einfluss der Strafwerthöhe zu vermeiden, könnte die Information über die Instabilität der Lösung direkt an einen entsprechenden Optimierer weitergegeben und von diesem verarbeitet werden. Die Optimierer von `scipy.optimize.minimize()` sind dafür jedoch nicht konzipiert.

Eine weitere Möglichkeit wäre, zu verhindern, dass der Optimierer den stabilen Bereich verlässt. Die dafür notwendigen Grenzen könnten als *constraints* für den `scipy.optimize.minimize()` Optimierer SLSQP vorgegeben werden. Hierfür würde sich die in Kapitel 5.2.4 und 5.5 diskutierte Grenze nach Gleichung 2.5.69 anbieten.

6.3 Auswahl der Optimierungsvariablen

Nach Kapitel 5.3 ist eine Reduktion der Optimierungsvariablen aus Effizienzgründen nicht von Nöten. Jedoch führte der Parameter C_μ , welcher die gleiche Größenordnung wie die Gradientenschrittweite aufweist, bei weiten Grenzen zur Irritation des Optimierers. Des Weiteren überzeichnet der Einfluss des relativ sensitiven Parameters C_μ die Wirkung der anderen Parameter auf die SFQs. Deren Einfluss auf die SFQs ist dann für den Optimierer nur bedingt identifizierbar. Künftige Optimierungen ohne C_μ als Optimierungsvariable wären daher durchaus sinnvoll.

Die Felder der Benchmark-Strömungen und der untersuchten Strömung verhalten sich bei Parametervariation qualitativ ähnlich. Dies kann sowohl an den Ergebnissen in Kapitel 4.1.5.1 und 5.2.2 als auch an den, für die Benchmark-Strömung hergeleiteten, Gleichungen 2.5.58 und 2.5.69 nachvollzogen werden. Die Unabhängigkeit der in diesen Gleichungen enthaltenen Modellparameter ist daher nicht unbedingt gewährleistet. Edeling et al. (2014) schlagen insofern die Anpassung von $C_{\varepsilon 1}$ und σ_ε nach Gleichung 2.5.58 bzw. 2.5.69 vor.

6.4 Überprüfung der Übertragbarkeit

Wie in Kapitel 5.5 diskutiert, ist die Übertragbarkeit der optimierten Parameter von einer Strömung im turbulenten Übergangsbereich zu einer Strömung mit signifikant höheren Turbulenzgrad nicht gewährleistet. Die Übertragbarkeit zwischen zwei vollturbulenten Strömungen wäre aber grundsätzlich denkbar. Weitere Untersuchungen sollten diese daher überprüfen. Um die Verbesserung der Strömungsfelder durch die optimierten Parameter garantieren zu können, müssten außerdem die Grenzen der Übertragbarkeit erhoben werden. Eine Formulierung dieser Grenzen wäre möglicherweise mit der Kugel-Reynoldszahl Re_K möglich.

Die Übertragbarkeit von Parametersätzen könnte a priori sichergestellt werden, wenn der Optimierer die SFQs mehrerer vollturbulenter Strömungen parallel minimiert. Dies könnte durch leichte Abwandlung des Frameworks bewerkstelligt werden. Hierfür müssten als Zielfunktion nur die SFQs der unterschiedlichen Strömungen des Feldes k aufsummiert an den Optimierer übergeben werden. Derartige Parallel-Optimierungen wurden von Souza et al. (2017) vorgenommen.

6.5 Anwendung anderer Turbulenzmodelle

Parametersätze in der Nähe der Standardparametersätze sind in Bezug auf die Übertragbarkeit vermutlich besser geeignet. Denn die untersuchte Strömung ist in vielen Belangen den Benchmark-Strömungen ähnlich (siehe Kapitel 5.6). Da durch die Benchmark-Strömungen eine grobe Abschätzung der Standardparameter möglich ist, sollten die angepassten Parameter auch in deren Nähe liegen.

6.5 Anwendung anderer Turbulenzmodelle

Finales Ziel der Anpassung der k - ε -Modellparameter ist eine effizientere Berechnung von korrekteren Strömungsfeldern zur Untersuchung des Wärmeübergangs zwischen einem kugelförmigen Partikel und einer ebenen Platte. Durch die Verwendung anderer Turbulenzmodelle könnte dies ebenfalls bewerkstelligt werden. Wie in Kapitel 2.5.3.5 diskutiert, könnte das k - ω -Modell dafür geeignet sein. Die Ergebnisse in Kapitel 5.3 verdeutlichen außerdem die wichtige Rolle der Anpassung von C_μ zur Verbesserung der Strömungsfelder. Die Verwendung des k - ε -realizable-Modells ist daher naheliegend. In diesem wird C_μ vom Modell selbst angepasst (siehe Kapitel 2.5.3.4).

7 Konklusion

Im Rahmen dieser Masterarbeit wurde das k - ε -Modell rigoros hergeleitet, erklärt und diskutiert, sowie die Herkunft der Modellparameter einschließlich deren Werten im Detail erläutert. Auf Basis dieser Ausführungen soll Folgearbeiten zu einer theoretisch fundierten und physikalisch sinnvollen Anpassung der k - ε -Modellparameter verholffen werden.

Durch die Ergebnisse einer umfangreichen Sensitivitätsanalyse konnten theoretische Zusammenhänge konkret mit der untersuchten Strömung verknüpft und die praktische Funktionsweise des k - ε -Modells besser nachvollzogen werden.

Ein Framework zur lokalen Optimierung der k - ε -Modellparameter wurde erfolgreich entwickelt und eingehend auf Funktion überprüft. Außerdem ist durch Einbettung des Hammersley-Sequence-Samplings die Generierung geeigneter Startwerte und die Findung eines globalen Optimums möglich.

Anhand einer dem turbulenten Übergangsbereich zuordenbaren Strömung wurden Optimierungen der k - ε -Modellparameter durchgeführt. Diese unterschieden sich durch die Anzahl der Optimierungsvariablen und den gewählten Grenzen für die Variablen. Die erhaltenen Ergebnisse konnten sowohl mit der Theorie als auch mit den Erkenntnissen der Sensitivitätsanalyse verbunden werden. Durch diese ist es in Zukunft möglich, effizient an optimierte und physikalisch legitime Parametersätze zu gelangen.

Die Übertragbarkeit der angepassten k - ε -Modellparameter wurde stichprobenartig mittels Berechnungen einer Strömung mit signifikant höherem Turbulenzgrad überprüft. Diese ist jedoch nicht gewährleistet. Eine mögliche Ursache könnte die sowohl qualitativ, als auch quantitativ unterschiedlich ausgeprägte Turbulenz in den beiden Strömungen sein.

Die Übertragbarkeit zwischen zwei ähnlich turbulenten Strömungen ist jedoch nicht ausgeschlossen und sollte daher in künftigen Arbeiten untersucht werden. Um die Verbesserung der Strömungsfelder durch die optimierten Parameter garantieren zu können, müssten außerdem die Grenzen der Übertragbarkeit erhoben werden.

Mit derart gewonnen Parametern ist die effiziente Generierung korrekter Strömungsfelder einer turbulenten Strömung zur Untersuchung des Wärmeübergangs zwischen einem kugelförmigen Partikel und einer ebenen Platte möglich.

8 Literaturverzeichnis

- Biegler, L. T. (2010). *Nonlinear programming: concepts, algorithms, and applications to chemical processes*. Philadelphia.
- Blom, J., & de Vries, D. A. (1970). *An experimental determination of the turbulent Prandtl number in a developing temperature boundary layer* (Proefschrift.). Eindhoven: Technische Hogeschool Eindhoven.
- Boussinesq, J. (1897). *Théorie de l'écoulement tourbillonnant et tumultueux des liquides dans les lits rectilignes a grande section*. Paris, Gauthier-Villars et fils. Retrieved from <http://archive.org/details/thbeoriedelbeco01bousrich>
- Bradshaw, P., Launder, B. E., & Lumley, J. L. (1996). Collaborative Testing of Turbulence Models. *Journal of Fluids Engineering*, 118(2), 243–247. <https://doi.org/10.1115/1.2817369>
- Brösigke, G., Herter, A., Rädle, M., & Repke, J.-U. (2017). Fundamental investigation of heat transfer mechanisms between a rolling sphere and a plate in OpenFOAM® for laminar flow regime. *International Journal of Thermal Sciences*, 111, 246–255. <https://doi.org/10.1016/j.ijthermalsci.2016.09.007>
- Broyden, C. G. (1970). The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations. *IMA Journal of Applied Mathematics*, 6(1), 76–90. <https://doi.org/10.1093/imamat/6.1.76>
- Comte-Bellot, G., & Corrsin, S. (1966). The use of a contraction to improve the isotropy of grid-generated turbulence. *Journal of Fluid Mechanics*, 25(4), 657–682. <https://doi.org/10.1017/S0022112066000338>
- Diwekar, U. M., & Kalagnanam, J. R. (1997). Efficient sampling technique for optimization under uncertainty. *AIChE Journal*, 43(2), 440–447. <https://doi.org/10.1002/aic.690430217>
- Edeling, W. N., Cinnella, P., Dwight, R. P., & Bijl, H. (2014). Bayesian estimates of parameter variability in the k- ϵ turbulence model. *Journal of Computational Physics*, 258, 73–94. <https://doi.org/10.1016/j.jcp.2013.10.027>
- Ferziger, J. H., & Peric, M. (2008). *Numerische Strömungsmechanik*. Springer Berlin Heidelberg.
- Fletcher, R. (1970). A new approach to variable metric algorithms. *The Computer Journal*, 13(3), 317–322. <https://doi.org/10.1093/comjnl/13.3.317>
- Goldfarb, D. (1970). A family of variable-metric methods derived by variational means. *Mathematics of Computation*, 24(109), 23–26. <https://doi.org/10.1090/S0025-5718-1970-0258249-6>
- Hanjalić, K., & Launder, B. E. (1972). A Reynolds stress model of turbulence and its application to thin shear flows. *Journal of Fluid Mechanics*, 52(4), 609–638. <https://doi.org/10.1017/S002211207200268X>
- Jones, W. P., & Launder, B. E. (1972). The prediction of laminarization with a two-equation model of turbulence. *International Journal of Heat and Mass Transfer*, 15(2), 301–314. [https://doi.org/10.1016/0017-9310\(72\)90076-2](https://doi.org/10.1016/0017-9310(72)90076-2)

- Jones, W. P., & Launder, B. E. (1973). The calculation of low-Reynolds-number phenomena with a two-equation model of turbulence. *International Journal of Heat and Mass Transfer*, 16(6), 1119–1130. [https://doi.org/10.1016/0017-9310\(73\)90125-7](https://doi.org/10.1016/0017-9310(73)90125-7)
- Kármán, T. V. (1930). *Mechanische Ähnlichkeit und Turbulenz*. Weidmannsche Buchh.
- Kelley, C. (1999). *Iterative Methods for Optimization*. Society for Industrial and Applied Mathematics. <https://doi.org/10.1137/1.9781611970920>
- Kim, J., Moin, P., & Moser, R. (1987). Turbulence statistics in fully developed channel flow at low Reynolds number. *Journal of Fluid Mechanics*, 177, 133–166. <https://doi.org/10.1017/S0022112087000892>
- Kolmogorov, A. N. (1942). Equations of Turbulent Motion in an Incompressible Fluid, *Izv. Akad. Nauk SSSR Ser Phys*(6), 328–330. https://doi.org/10.1007/978-94-011-3030-1_48
- Kolmogorov, A. N. (1991). The Local Structure of Turbulence in Incompressible Viscous Fluid for Very Large Reynolds Numbers. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 434(1890), 9. <https://doi.org/10.1098/rspa.1991.0075>
- Launder, B. E., & Sharma, B. I. (1974). Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc. *Letters in Heat and Mass Transfer*, 1(2), 131–137. [https://doi.org/10.1016/0094-4548\(74\)90150-7](https://doi.org/10.1016/0094-4548(74)90150-7)
- Mansour, N. N., Kim, J., & Moin, P. (1988). Reynolds-stress and dissipation-rate budgets in a turbulent channel flow. *Journal of Fluid Mechanics*, 194, 15–44. <https://doi.org/10.1017/S0022112088002885>
- Menter, F. R. (1994). Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, 32(8), 1598.
- Mohamed, M. S., & Larue, J. C. (1990). The decay power law in grid-generated turbulence. *Journal of Fluid Mechanics*, 219, 195–214. <https://doi.org/10.1017/S0022112090002919>
- Papageörgiu, M. (2015). *Optimierung : statische, dynamische, stochastische Verfahren für die Anwendung* (4., korrigierte Auflage). Berlin.
- Paschedag, A. R. (2004). *CFD in der Verfahrenstechnik: Allgemeine Grundlagen und mehrphasige Anwendungen* (1st ed.). Weinheim: Wiley-VCH Verlag GmbH & Co. KGaA.
- Pope, S.B. (2000). *Turbulent Flows*. Cambridge University Press. Retrieved from <https://books.google.de/books?id=HZsTw9SMx-0C>
- Pope, Stephen B. (2013, January 22). S.B. Pope: TURBULENT FLOWS. Retrieved March 8, 2017, from <https://pope.mae.cornell.edu/TurbulentFlows.html>
- Prandtl, L. (1931). Über die Entstehung der Turbulenz. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift Für Angewandte Mathematik Und Mechanik*, 11(6), 407–409. <https://doi.org/10.1002/zamm.19310110602>
- Prandtl, L. (1945). *Über ein neues Formelsystem für die ausgebildete Turbulenz*. Göttingen: Vandenhoeck & Ruprecht.

- Reynolds, O. (1895). On the Dynamical Theory of Incompressible Viscous Fluids and the Determination of the Criterion. *Philosophical Transactions of the Royal Society of London. (A.)*, 186, 123. <https://doi.org/10.1098/rsta.1895.0004>
- Richardson, L. F. (1922). *Weather Prediction by Numerical Process*. Cambridge: Cambridge University Press.
- Shanno, D. F. (1970). Conditioning of quasi-Newton methods for function minimization. *Mathematics of Computation*, 24(111), 647–656. <https://doi.org/10.1090/S0025-5718-1970-0274029-X>
- Souza, L. M. D., Janiga, G., & Thévenin, D. (2017). Multi-objective optimisation of the model parameters for the realisable k- ϵ turbulence model. *Progress in Computational Fluid Dynamics, an International Journal*, 17(2), 90–101. <https://doi.org/10.1504/PCFD.2017.082523>
- Speight, J. G. (2014). The chemistry and technology of petroleum. *Chemical Industries*.
- Tavoularis, S., & Corrsin, S. (1981). Experiments in nearly homogenous turbulent shear flow with a uniform mean temperature gradient. Part 1. *Journal of Fluid Mechanics*, 104, 311–347. <https://doi.org/10.1017/S0022112081002930>
- Tebianian, S., Dubrawski, K., Ellis, N., Cocco, R. A., Hays, R., Reddy Karri, S. B., ... Grace, J. R. (2015). Investigation of particle velocity in FCC gas-fluidized beds based on different measurement techniques. *Chemical Engineering Science*, 127, 310–322. <https://doi.org/10.1016/j.ces.2015.01.049>
- The Scipy Community. (2016, September 19). `scipy.optimize.minimize` — SciPy v0.18.1 Reference Guide. Retrieved April 6, 2017, from <https://docs.scipy.org/doc/scipy-0.18.1/reference/generated/scipy.optimize.minimize.html#scipy.optimize.minimize>
- Tong, C., & Warhaft, Z. (1995). Passive scalar dispersion and mixing in a turbulent jet. *Journal of Fluid Mechanics*, 292, 1–38. <https://doi.org/10.1017/S0022112095001418>
- Treese, S. A., Pujadó, P. R., & Jones, D. S. J. (2015). *Handbook of Petroleum Processing* (Vol. 1). Cham.
- Ulbrich, M. (2012). Nichtlineare Optimierung. *Mathematik Kompakt*.
- Van Dyke, M. (1982). *An album of fluid motion*. Stanford, Calif.: Parabolic Press.
- Wilcox, D. C. (1993). *Turbulence Modeling for CFD*. DCW Industries, Incorporated.
- Wong, T.-T., Luk, W.-S., & Heng, P.-A. (1997). Sampling with Hammersley and Halton Points. *Journal of Graphics Tools*, 2(2), 9–24. <https://doi.org/10.1080/10867651.1997.10487471>
- Zhu, C., Byrd, R. H., Lu, P., & Nocedal, J. (1997). Algorithm 778: L-BFGS-B: Fortran Subroutines for Large-scale Bound-constrained Optimization. *ACM Trans. Math. Softw.*, 23(4), 550–560. <https://doi.org/10.1145/279232.279236>

Anhang

Anhang A Allgemeiner Aufbau Framework

Das Framework kann als eine Erweiterung der OpenFOAM-Case-Filestruktur (siehe Kapitel 4.1.1.1) angesehen werden. Es wurde für die OpenFOAM-Version 2.4.0 entwickelt. Die Ordnerstruktur des Frameworks enthält daher die default Ordner *constant*, *system* sowie allfällige *Timedirectories*.

A.1 system

In *system* liegt das default File *controlDict*. Da davon ausgegangen werden kann, dass die Parameteränderungen durch den Optimierer klein sind, werden als Startfelder einer neuen Berechnung, die Lösungsfelder der letzten konvergierten Berechnung herangezogen. Die Option *startTime* muss daher *latestTime* lauten.

Konvergiert eine Berechnung nicht, wird die Anzahl der maximalen Iterationsschritte ebenfalls durch das *controlDict* vorgegeben. Sie wird durch die Angabe eines geeigneten Wertes für *writeIntervall* hinterlegt. Die Option *stopAt* muss hierbei *nextWrite* lauten.

Der Ordner *system* enthält auch das *decomposeParDict*. Dieses File ist notwendig, da die Domain auf mehrere Prozessoren aufgeteilt und die Berechnung auf diesen parallel stattfinden soll. Das Framework veranlasst die dafür notwendigen Schritte automatisch und erlaubt auch eine Änderung der Anzahl der verwendeten Prozessoren während der Optimierung.

A.2 constant

In *constant* liegt das bei Verwendung von RANS-Modellen notwendige default File *RASProperties*. In diesem File kann das k- ϵ -Modell als RANS-Modell ausgewählt und die zu verwendenden Modellparameter vorgegeben werden. Des Weiteren können die Parameter der Vollständigkeit halber in das log-File der Berechnung (*./timelog/logi*) geschrieben werden (mit der Option: *printCoeffs on*).

Ebenfalls in *constant* liegt das Textfile *optimizationInfo*. In dieses gibt das Programm *SumSqrErr* die Summe der Fehlerquadrate (SFQ), sowie die aktuellen k- ϵ -Parameter aus.

A.3 Timedirectories

Vor dem Start der Optimierung müssen zwei *Timedirectories* im Case-File liegen.

0: enthält die gemittelte Lösung der DNS

1: enthält die Startfelder der ersten Berechnung

Die Namen (0, 1) sind dabei wichtig. Das Programm *SumSqrErr* geht immer davon aus, dass die DNS Ergebnisse in *0* gespeichert sind. Die Benennung *1* ist für das Python Programm *printlogResults* (siehe Kapitel C.4.1) wichtig. Dieses unterscheidet *Timedirectories* von anderen Ordnern anhand ihres Namens der Form „1.*“.

A.4 processorN

Auf diese Ordner wird die Domain aufgeteilt. Sie werden im Framework automatisch angelegt. Die in ihnen enthaltenen Felder werden tatsächlich zur Berechnung herangezogen. Insofern wird auch die Lösung der letzten Iteration in diese Ordner geschrieben. Lösungen deren Residuen nicht konvergieren, müssen daher vom Framework aus den Ordnern gelöscht werden, da ansonsten die nächste Berechnung auf ihnen aufbaut. Vom Framework der lokalen Optimierung werden aus Effizienz- und Speicherplatzgründen nicht alle konvergierten Lösungen zu *Timedirectories* zusammengesetzt.

A.5 timelog

In diesem automatisch angelegten Ordner befinden sich die log-Files der Berechnungen. Die log-Files enthalten sowohl alle wesentlichen Informationen zur k- ϵ -Simulation, als auch zur Ermittlung der SFQs. Anhand der Namen der log-Files kann der ordnungsgemäße Verlauf der Optimierung gut nachvollzogen werden. Dieser setzt sich wie folgt zusammen:

log_(i ... Laufnummer)_(Berechnungsausgang)_(Timedirectory der Lösung)

Da für die nicht konvergierten Lösungen kein *Timedirectory* vorliegt, reduziert sich der Name für diese auf *i* und *Berechnungsausgang*.

A.6 gnuplot

Dieser Ordner enthält Files zum Monitoring der Berechnungen. Sie werden mit dem Konsolenbefehl „*gnuplot Filename*“ gestartet.

<i>leastSquares:</i>	plottet den Verlauf der Zielfunktion
<i>residuals:</i>	plottet den Verlauf der Residuen aller OpenFOAM-Berechnungen
<i>residualsConverged:</i>	plottet den Verlauf der Residuen aller konvergierten OpenFOAM-Berechnungen
<i>residualsNonconverged:</i>	plottet den Verlauf der Residuen aller nicht konvergierten OpenFOAM-Berechnungen
<i>residualsLogi:</i>	plottet den Verlauf der Residuen der OpenFOAM-Berechnungen mit dem log-File <i>./timelog/logi</i>

Der Suffix „*PNG*“ gibt an, wenn die Ausgabe in ein PNG-File erfolgt.

A.7 src

Dieser Ordner ist ein evolutionäres Überbleibsel. Das Framework selbst und einige zur Entwicklung notwendige Visualisierungs- und Datenbearbeitungsprogramme sind Python-Programme. Sie können mit dem Programmierungswerkzeug *Eclipse* bearbeitet werden. Dieses legt einen Source-Ordner (*src*) an. Viele der entwickelten Programme sind durch das lange problematische und erst später zufriedenstellend entwickelte Ausgabefile *logResults* obsolet geworden. Nach wie vor wichtige in *src* enthaltene Programme sind in den Kapiteln B.4 und C.4 beschrieben.

A.8 Weitere Ordner und Files

Der Ordner *templates* enthält wichtige OpenFOAM Kopf- und Fußzeilen, die vom Framework herangezogen werden. *Cleaner.py* ist ein Programm welches das Case-File in den Ausgangszustand zurücksetzt und sollte daher vor jeder Optimierung gestartet werden (Achtung löscht Lösungen!). Das File *iterate.dat* enthält zusätzliche Information bezüglich dem Ablauf der Optimierung.

A.9 Framework

Das Framework für die lokale Optimierung ist in Form der Textfiles *opti-script.py* und *OpenFoamController.py* in dem Case-File abgelegt. Das Hammersley-Framework ist in Form des Textfiles *Script.py* im Case-File hinterlegt.

Anhang B Aufbau Framework lokale Optimierung

B.1 opti-script.py

Das File *opti-script.py* ist der Überbau des Frameworks. Zum Start der Optimierung wird nur das Programm *opti-script.py* über die Konsole („*python opti-script.py*“) aufgerufen. Dieses greift dann auf *OpenFoamController.py* zu. Wesentliche Einstellungen sind im *opti-script.py* unter der Sektion *USER INPUT* vorzunehmen:

initialValues: Die Variable ist ein Array mit den Startwerten der k-ε-Modellparameter für die lokale Optimierung ($\text{array}([C_\mu, C_{\varepsilon 1}, C_{\varepsilon 2}, \sigma_\varepsilon, \sigma_k])$). Durch die Anzahl der Einträge im Array wird die Anzahl der Optimierungsvariablen definiert. Welcher Arrayeintrag welchem k-ε-Modellparameter zugewiesen wird, muss im *OpenFOAM-Controller* gesteuert werden (siehe B.2).

multiPlyer: Die Ober-/Untergrenzen der lokalen Optimierung werden wie folgt festgelegt: ($\text{initialValues} \pm \text{initialValues} * \text{multiPlyer}$).

Auch eine individuelle Anpassung ist möglich. Hierfür muss in Zeile 86 (außerhalb *USER INPUT*) die variable *parameter_boundaries* wie folgt neu definiert werden: $\text{array}([[C_{\mu,UG}, C_{\mu,OG}], [C_{\varepsilon 1,UG}, C_{\varepsilon 1,OG}], [C_{\varepsilon 2,UG}, C_{\varepsilon 2,OG}], [C_{\sigma_\varepsilon,UG}, C_{\sigma_\varepsilon,OG}], [C_{\sigma_k,UG}, C_{\sigma_k,OG}]])$, mit Untergrenze (UG) und Obergrenze (OG)

dummyEntry: Mit dieser Variable kann die Höhe des Strafwerts festgelegt werden.

opt: Durch die Variable *opt* können Optimierungseinstellungen geändert werden. Mögliche Optionen können folgender Website entnommen werden: <https://docs.scipy.org/doc/scipy-0.18.1/reference/optimize.minimize-lbfgsb.html>

Wichtige Optionen sind dabei *disp* (wenn *TRUE* wird das log-File *iterate.dat* ausgegeben) und *eps* (Gradientenschrittweite). Sie sind daher standardmäßig hinterlegt.

Method: Hier kann das gewünschte Optimierungsverfahren ausgewählt werden. Die Auswahl für *scipy.optimize.minimize()* steht auf folgender Website bereit: <https://docs.scipy.org/doc/scipy-0.18.1/reference/optimize.html>. Das

Anhang

Framework wurde jedoch nur eingehend für *L-BFGS-B* getestet. Es erfolgten jedoch auch Berechnungen mit *BFGS*, *TNC*, *SLSQP*.

Zur Anpassung der Zielfunktionsberechnung aus den SFQs muss die Definition der Variablen *weight* und *leastSquares* außerhalb der *USER INPUT* Sektion (Zeilen 53, 54) geändert werden.

Außerdem soll hier eingeräumt werden, dass die Definition von Strafwerten an drei Orten (*optimizationInfoNC*, *OpenFoamController.py* – ca. Zeile 274, *opti-script.py* – Zeile 57) im Framework stattfindet. Diese Unsinnigkeit ist historisch gewachsen und kann in dem hier vorliegenden Anhang für Verwirrung sorgen. Aufgrund der Programm-Hierarchie ersetzt *opti-script.py* die Strafwert Definition von *OpenFoamController.py*, welche wiederum die Strafwerte von *optimizationInfoNC* in *optimizationInfo* überschreibt (siehe Kapitel D.2). Letzten Endes ist für den Optimierer mit der derzeitigen Version des Frameworks nur die Definition von *opti-script.py* wesentlich.

B.2 OpenFoamController.py

Auch in *OpenFOAM-Controller.py* sollten vor dem Start des Frameworks einige Einstellungen überprüft werden. Auch sie befinden sich in der Sektion *USER INPUT*.

pathCaseFile: Dateipfad des Ordners in dem die *Timedirectories* der gemittelten DNS-Lösung und des k - ε -Startfeldes sowie die Ordner *system* und *constant* liegen.

openFOAMalias: Alias zum Starten von OpenFOAM hinterlegt in *~/.bashrc*.

numberparameters: Anzahl der Optimierungsvariablen, notwendig für das Überspringen der Gradientenberechnung an instabilen Punkten.

reconstructme: Gibt die Anzahl der Lösungen an, welche reconstructed also wieder zusammengesetzt werden sollen. (1: alle, 2: jede zweite, ...)

Welche der k - ε -Parameter als Optimierungsvariable herangezogen werden soll, muss außerhalb von *USER INPUT* definiert werden. Im digitalen Anhang sind daher beispielhaft sowohl eine Optimierung von $C_{\varepsilon 1}$ und $C_{\varepsilon 2}$ als auch eine Optimierung aller Parameter zu finden. In deren *OpenFoamController.py* Files ist die entsprechende Implementierung zw. Zeile 54 und 59 nachzuvollziehen.

Grundsätzlich wird die Auswahl in der OpenFoam-Controller Methode *transcribe_parameter_values()* vorgenommen. Diese erhält vom Optimierer die Parameter in dem Array *p*. In diesem stehen die Parameter mit der durch den Array *initialValues* festgelegten Reihenfolge. Die Methode *transcribe_parameter_values()* liest die Parameter aus *p* und trägt sie in *RAS-Properties* ein. Daher muss in ihr festgelegt werden, welcher Eintrag von *p*, welchem Parameter in *RASProperties* zugeordnet wird.

B.3 logResults

WICHTIG!!! Im *logResults* Textfile werden die Namen der *Timedirectories* aller Lösungen, deren *Parameter* und die dazugehörigen *SFQs* gespeichert. Es ist daher essentiell um den Verlauf und die Ergebnisse der Optimierung illustrieren und auswerten zu können.

Ist ein anderes Format gewünscht, müssen Änderungen in *opti-script.py* in Zeile 62-74 vorgenommen werden. Auch eine kleine und unkomplizierte Anpassung der OpenFoamController-Methode *transcribe_parameter_values()* kann notwendig sein, da dort der Eintrag der Parameter für *logResults* formatiert wird.

Die *SFQs* für *logResults* werden aus dem Textfile *optimizationInfo* entnommen. Im Falle von nicht konvergenten Lösungen entsprechen die Strafwerte daher der Definition von *OpenFoamController.py* und nicht der von *opti-script.py*.

B.4 src

B.4.1 plotlogResults

Kombination aus einem python und gnuplot Programm. Greift auf *logResults* zu und normiert die darin eingetragenen *SFQs* nach den *SFQs* der Standardparameter (Standard*SFQs*) und den *SFQs* der ersten Lösung (in der Regel die *SFQs* der Startwerte, Initial*SFQs*). Die normierten *SFQs* werden dann in die Textfiles *StandardOptimizationResume* bzw. *InitialOptimizationResume* geschrieben, wo sie eingesehen werden können. Die Resümees sind beide im *src* Ordner zu finden.

Bei Aufruf von *plotlogResults* (Konsole „*python plotlogResults*“) wird auch automatisch der Plot der normierten *SFQs* in die Files *StandardSFQs.png* und *InitialSFQs.png* veranlasst. Dort werden die *SFQs* über der benötigten Anzahl der OpenFOAM-Iterationen dargestellt. Hohe normierte *SFQ*-Werte werden der Übersichtlichkeit halber von *gnuplot* auf einen konstanten Wert hinab gesetzt.

Anhang

Mit der Funktion *order* in der Sektion *USER INPUT* kann die Ordnung der Parametersätze nach ansteigenden SFQs veranlasst werden (*order = True*). Welcher der SFQs zur Sortierung herangezogen werden soll, wird ebenfalls in *USER INPUT* durch den Integer *choose* bestimmt (0: U_x , 1: U_y , 2: U_z , 3: k , 4: ε). Die Sortierung betrifft nur die Resümees. Der beste Parametersatz bzgl. eines SFQs kann dann aus den Resümees als erster Eintrag ausgelesen werden.

Anhang C Aufbau Framework Hammersley-Verteilung

C.1 Script.py

Script.py enthält das Hammersley Framework. Dieses wird in der Konsole mit „python *Script.py*“ gestartet. Es benötigt im Case-File das Textfile *Hammersley* mit der Auflistung der Hammersley-Punkte. Diese sollten davor durch das Programm *OrderHammersleyPoints* geordnet werden. Es kann aber auch zur Berechnung beliebiger Punkte herangezogen werden. Da das Hammersley-Skript dafür ausgelegt wurde, gute Startwerte zu finden und die dazugehörigen Felder der lokalen Optimierung zur Verfügung zu stellen, wird in *Script.py* jede konvergierte Lösung zusammengesetzt. Daher muss genügend Speicherplatz zur Verfügung gestellt werden.

C.2 Generierung der Hammersley-Punkte

Die Generierung der Hammersley-Punkte ist mit der, dem Hammersley-Framework beigelegten, *Hammersley_Sequence_Library* möglich. Dafür muss zunächst das in der Library vorliegende File *hammersley_dataset.m* mit Matlab geöffnet und das darin enthaltene Programm gestartet werden. Bei Befolgung der klaren Anweisungen werden die Hammersley-Punkte in Form eines Textfiles in die *Hammersley_Sequence_Library* ausgegeben. Der Name des Textfiles wird dabei durch *hammersley_dataset.m* angegeben.

Da die ausgegebenen Hammersley-Punkte h_i nur Werte zwischen eins und null aufweisen, müssen sie noch bezüglich der gewünschten Ober- und Untergrenzwerte H_{OG} und H_{UG} in die tatsächlich benötigten Werte H_i umgerechnet werden. Die einfachste Umrechnungsmethode ist linear:

$$H_i = (H_{OG} - H_{UG})h_i + H_{UG}$$

Unterscheiden sich H_{OG} und H_{UG} um Größenordnungen, so ist eine logarithmische Umrechnung vorzuziehen:

$$\log(H_i) = (\log(H_{OG}) - \log(H_{UG}))h_i + \log(H_{UG})$$

Die Umrechnung kann in *Excel* erfolgen. Beispielhaft liegt hierfür dem Hammersley-Framework das *Excel*-File *Hammersley.xlsx* bei.

C.3 Hammersley

Das Textfile, in dem die Hammersley-Punkte aufgelistet sind, muss *Hammersley* benannt werden. Die erste Zeile ist dabei ein beliebiger Header. Ab der zweiten Zeile folgen dann die Punkte der Form:

$$C_\mu \backslash tC_{\varepsilon 1} \backslash tC_{\varepsilon 2} \backslash t\sigma_\varepsilon \backslash t\sigma_k \backslash n$$

C.4 src

C.4.1 printlogResults

Da im Hammersley-Framework *opti-script.py* nicht vorliegt, fehlt *logResults* als Ausgabefile. Die Ausgabe von *logResults* kann manuell durch *printlogResults* (Konsolenbefehl: „*python printlogResults*“) veranlasst werden. Das Programm greift auf die in den *Timedirectories* abgelegten OpenFoam-Dictionaries *info* zu, um die k- ε -Modellparameter und die SFQs auszulesen. Diese werden dann samt den Namen der *Timedirectories* in *logResults* zusammengefasst.

C.4.2 plotlogResultsHammersley

Eine leichte Abwandlung des Programmes *plotlogResults* (siehe Kapitel B.4.1) ist das Programm *plotlogResultsHammersley*. Der entscheidende Unterschied besteht darin, dass die SFQs über einer Berechnungskennnummer aufgetragen und hohe SFQ-Werte von *gnuplot* nicht auf einen konstanten Wert hinabgesetzt werden. Es eignet sich daher auch für den Einsatz im Framework der lokalen Optimierung.

C.4.3 OrderHammersleyPoints

Zur Effizienzsteigerung zieht das Framework zur Berechnung der Lösung eines Punktes als Startfelder die Lösung des vorhergehenden Punktes heran. Liegt daher Letzterer nahe am aktuellen Punkt, sind die Startfelder bessere Abschätzungen der Lösung, die Rechendauer verkürzt sich. *OrderHammersleyPoints* versucht daher die Abstände zweier in der Berechnungsreihenfolge aufeinanderfolgende Punkte zu minimieren. Es wählt daher den ersten Punkt in der Auflistung der Hammersleypunkte und sucht für diesen den Punkt mit dem geringsten Euklidischen Abstand. Für letzteren wird wiederum der Punkt mit dem geringsten Euklidischen Abstand aus den verbleibenden Punkten ermittelt und so weiter. Hierfür entnimmt *OrderHammersleyPoints* eine Liste von Hammersley-Punkten aus dem Textfile *Hammersley.org* und schreibt die geordnete Liste in das Textfile

Hammersley. Beide Textfiles sind dabei auf gleiche Weise formatiert und müssen im selben Ordner wie `OrderHammersleyPoints` liegen.

C.4.4 **makeGrid**

Das Programm *makeGrid* generiert eine Liste an (k - ϵ -Modellparameter-)Punkten für ein zweidimensionales, rechteckiges Raster. Die Reihenfolge der Liste entspricht einem schlangenlinienförmigen Durchlauf der Punkte. Dadurch ist es möglich, effizient Datensätze für dreidimensionale Zielfunktionsplots mit Hilfe des Hammersley-Frameworks zu berechnen. Das Format des Ausgabefiles *grid* entspricht dabei dem des Textfiles *Hammersley*.

Anhang D OpenFOAM-Postprocessing-Tools

Die entwickelten OpenFOAM-Postprocessing-Tools werden wie alle anderen OpenFOAM-Programme durch ihren Namen mit der Konsole aufgerufen. Mit Letzterer musste zuvor in den entsprechenden OpenFOAM-Case navigiert werden („cd“). Die Postprocessing-Tools ziehen zur Berechnung jene *Timedirectories* heran, welche im *controlDict* festgelegt wurden (z.B.: *latestTime*). Um sie anwenden zu können, müssen die im digitalen Anhang als Ordner hinterlegten Tools in den Ordner `~/OpenFOAM/Username-OpenFOAM-version/applications` verschoben und mit den Konsolenbefehlen „wclean“ und „wmake“ kompiliert werden.

D.1 Sensitivity

Sensitivity ist ein OpenFOAM-Postprocessing-Tool und berechnet die partiellen Ableitungen S der Felder $\phi = U, k, \varepsilon$ nach den k- ε -Modellparametern P_i :

$$S(x, P_i, \Delta P_i) = \frac{\phi(x, P_i + \Delta P_i) - \phi(x, P_i)}{\Delta P_i}$$

Das Programm benötigt daher die entsprechenden Felder $\phi(x, P_i + \Delta P_i)$ bzw. $\phi(x, P_i)$ und die herangezogenen Parameter $P_i + \Delta P_i$ sowie P_i . Die Felder müssen insofern korrekt, durch Variation eines Parameters, zur Verfügung gestellt werden. Ihre Berechnung kann automatisiert durch das Hammersley-Framework erfolgen.

Die Felder $\phi(x, P_i)$ müssen aufgrund der Beschaffenheit von *Sensitivity* in das Timedirectory „0“ gelegt werden. Durch das *controlDict* wird definiert, aus welchem Timedirectory die Felder $\phi(x, P_i + \Delta P_i)$ entnommen werden.

Die zur Bildung der Sensitivitäten notwendigen Parameter müssen in den entsprechenden *Timedirectories* durch das Textfile log vermerkt werden. Dieses hat den folgenden Aufbau (die Parameter müssen noch entsprechend angepasst werden):

```

1  FoamFile
2  {
3      version      2.0;
4      format       ascii;
5      class        dictionary;
6      object       log;
7  }
8
9  kEpsilonCoeffs
10 {
11     Cmu           0.09;
12     C1            1.44;
13     C2            1.92;
14     sigmaEps      1.30;
15     sigmaK        1.00;
16 }
```


Wurden die Felder durch das Hammersley-Skript berechnet, so wäre auch ein Auslesen der Parameter aus den OpenFOAM-Dictionary *info* möglich. Dafür müsste jedoch *Sensitivity* bearbeitet und neu kompiliert werden. Steht erneut eine umfangreiche Sensitivitätsanalyse bevor, würde sich der dafür nötige Aufwand schnell rechnen. Sind die SFQs nicht von Interesse, könnte zur Effizienzsteigerung der Berechnungen der Felder ϕ in Script.py der Aufruf von *SumSqrErr* durch den Aufruf von *optimizationInfoNC* ersetzt werden (siehe Kapitel D.2.1).

D.2 SumSqrErr

SumSqrErr berechnet die Summe der Fehlerquadrate (SFQ) der Felder $\phi_i = \mathbf{U}_i, k_i, \varepsilon_i$ mit denen der Felder $\phi_0 = \mathbf{U}_0, k_0, \varepsilon_0$:

$$SFQ = \sum \left[\phi(\text{Zelle}_j)_i - \phi(\text{Zelle}_j)_0 \right]^2$$

Für ϕ_0 werden aufgrund der Beschaffenheit von *SumSqrErr* jene Felder herangezogen, welche im *Timedirectory* „0“ liegen. Das *Timedirectory* der Felder ϕ_i wird durch das *controlDict* bestimmt.

Die Ergebnisse werden in Form von zwei OpenFOAM-Dictionaries ausgegeben. Das Dictionary *optimizationInfo* wird im Ordner *constant* und das Dictionary *info* im *Timedirectory* der Felder ϕ_i hinterlegt. Beide sind ident und enthalten:

- den Berechnungsausgang (daher *converged*)
- die k- ε -Modellparameter (ausgelesen aus *./constant/RASProperties*)
- die SFQs der einzelnen Felder

Da die Differenzen der SFQs zwischen den Zellenwerten gebildet werden, müssen die Rechengitter der Felder ϕ_0 und ϕ_i ident sein. Die Felder ϕ_0 müssen daher auf das Gitter von ϕ_1 gemappt werden. Dies kann durch das OpenFOAM-Programm *mapFields* erfolgen. Das dafür notwendige Konsolenkommando lautet beispielhaft für das mappen der DNS-Lösung auf das grobe Gitter:

```
„mapFields 'Pfad-Source-Case-File' -fields '(UMean, kMean, epsilonMean)' -sourceRegion 'Gas' -sourceTime 'Timedirectory' -consistent“
```

Aus Effizienzgründen wird dazu geraten, die Felder jedoch einzeln zu mappen. So erfolgt eine Arbeitsteilung auf drei Prozessoren. Ein *mpirun* ist mit *mapFields* grundsätzlich möglich aber nicht unproblematisch.

D.2.1 **optimizationInfoNC**

Bei *optimizationInfoNC* handelt es sich um eine Abwandlung von *SumSqrErr*. In ihr werden jedoch nicht die SFQs berechnet. Stattdessen wird nur das Directory *optimizationInfo* ausgegeben. In diesem sind neben dem Berechnungsausgang (*nonconverged*) und den k- ϵ -Modellparametern hohe, konstante Strafwerte für die SFQs eingetragen.

Das Framework der lokalen Optimierung ruft *optimizationInfoNC* nur dann auf, wenn eine Lösung nicht konvergiert. Aus Effizienzgründen könnten die Aufgaben von *optimizationInfoNC* vom Framework übernommen werden.

D.2.2 **AllSumSqrErr**

AllSumSqrErr ist eine weitere Abwandlung von *SumSqrErr*. Durch *AllSumSqrErr* wird die Berechnung der SFQs mit allen Feldern ϕ_i vorgenommen, welche sich innerhalb eines *Timedirectorys* im Case-File befinden. Die SFQs werden dabei immer mit den Feldern ϕ_0 gebildet. Da sowohl *SumSqrErr* als auch *AllSumSqrErr* die Parameter aus *./constant/RASProperties* beziehen, sind die Parametereinträge in *info* dann jedoch fehlerhaft.

Anhang E Bedienungsanleitung lokale Optimierung

1. Lektüre der Kapitel 4.1.1.1, 4.2, 4.3, Anhang A und 0 der Masterarbeit von Benjamin Lukitsch.
2. Mappen der gemittelten DNS-Lösung auf das grobe Rechengitter (*OpenFOAM – mapFields*)
3. Anlegen eines Case-Files mit den Ordnern bzw. Files:
 - DNS-Lösung „0“
 - auskonvergiertes $k-\epsilon$ -Startfeld „1“
 - constant
 - system
 - src
 - gnuplot
 - templates
 - OpenFoamController.py
 - opti-script.py
 - Cleaner.py
4. Überprüfung der Einstellungen in *system* (A.1) und *constant* (A.2) sowie in *opti-script.py* (B.1) und *OpenFOAM-Controller.py* (B.2)
5. Mit Konsole in das Case-File-Directory navigieren „cd“
6. Aufräumen des Case-Files mit Konsolenbefehl: „Cleaner.py“ – Achtung löscht Lösungen
7. Starten der Optimierung mit Konsolenbefehl: „python opti-script.py“
8. Warten, ob nach der Decomposition die Optimierung startet (output in Konsole beachten, die Berechnung der ersten Felder geht bei auskonvergierten $k-\epsilon$ -Startfeldern relativ schnell vonstatten)
9. WARTEN
10. Residuen überprüfen (*gnuplot residuals*) und die maximale Anzahl an Iterationen anpassen (*system/controlDict/writeIntervall*), auf Ungereimtheiten achten
11. SFQ-Verläufe auf Konvergenz kontrollieren (*python plotlogResults*)
12. WARTEN
13. Residuen überprüfen (*gnuplot residuals*) und die maximale Anzahl an Iterationen anpassen (*system/controlDict/writeIntervall*), auf Ungereimtheiten achten
14. SFQ-Verläufe auf Konvergenz kontrollieren (*python plotlogResults*)
15. Wenn die SFQs in glatte, nur noch gering absinkende Verläufe übergegangen sind, *logResults* auf Änderung der Parameter untersuchen. Sind diese klein, suggeriert nur noch das Rauschen dem Optimierer eine Absenkung der Zielfunktion.
16. Optimierung abbrechen (Konsole: „strg. + C“)
17. Mit *plotlogResults* und der Option *order = True* die Parametersätze nach ihrer Güte sortieren und den Besten auslesen.

Anhang F Bedienungsanleitung Hammersley-Verteilung

1. Lektüre der Kapitel 4.1.1.1, 4.2, 4.3, Anhang A und Anhang C der Masterarbeit von Benjamin Lukitsch.
2. Mappen der gemittelten DNS-Lösung auf das grobe Rechengitter (*OpenFOAM – mapFields*)
3. Generierung von Hammersley-Punkten
4. Anlegen eines Case-Files mit den Ordnern bzw. Files:
 - DNS-Lösung „0“
 - auskonvergiertes k- ϵ -Startfeld „1“
 - constant
 - system
 - src
 - gnuplot
 - templates
 - Script.py
 - Cleaner.py
5. Überprüfung der Einstellungen in *system* (A.1) und *constant* (A.2)
6. Mit Konsole in das Case-File-Directory navigieren „cd“
7. Aufräumen des Case-Files mit Konsolenbefehl: „Cleaner.py“
8. Starten der Berechnungen mit Konsolenbefehl: „Script.py“
9. Warten, ob nach der Decomposition die Berechnung startet (Blick in *timelog/logi* werfen)
10. WARTEN
11. Residuen auf Instabilitäten überprüfen („gnuplot residuals“) und die maximale Anzahl an Iterationen zu einem sinnvollen Abbruchkriterium anpassen (*./system/controlDict/writeIntervall*)
12. *logResults* erstellen (*python printlogResults*)
13. SFQ-Verläufe plotten (*python plotlogResultsHammersley*)