



TECHNISCHE
UNIVERSITÄT
WIEN

Vienna University of Technology

Diplomarbeit

Modellierung und Simulation von thermischen Speichern

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines

Diplom-Ingenieurs

unter der Leitung von

Ao.Univ.Prof. Dipl.-Ing. Dr.techn. KARL PONWEISER

und Projektass. Dipl.-Ing. DOMINIK BOTHE BSc.

am Institut für

Energietechnik und Thermodynamik / E302

eingereicht an der Technischen Universität Wien

Fakultät für Maschinenwesen und Betriebswissenschaft

von

STEFAN FALKNER

0826830

Seibersdorf 15, 4131 Kirchberg/D.

Wien, am 29. Sept. 2014

eigenhändige Unterschrift

Danksagung

Ich möchte mich in aller Form bei Herrn Professor Karl Ponweiser bedanken, der mich bei meiner Diplomarbeit unterstützt, mir mit wertvollen Hinweisen und Tipps zur Seite gestanden und als Gutachter dieser Arbeit fungiert hat. Mein Dank geht ebenfalls an Herrn Dipl.-Ing. Dominik Bothe, der mir durch sein Fachwissen und seine Anregungen oft weitergeholfen hat.

Mein ganz besonderer und tiefer Dank gilt meinen Eltern Christine und Bernhard, die mir mein Studium nicht nur durch finanzielle sondern auch durch ihre moralische Unterstützung überhaupt erst ermöglicht haben.

Bei meiner Freundin Patricia möchte ich mich recht herzlich bedanken. Sie hat mich während der gesamten Diplomarbeitsphase unterstützt und auch in hektischen Zeiten immer großes Verständnis gezeigt.

Abschließend bedanke ich mich bei all meinen Freunden und Mitbewohnern, die mir die gemeinsame Zeit in Wien und zu Hause in Oberösterreich in einen Hochgenuss verwandelten.

Kurzfassung

Schwindende Vorräte an fossilen Energieträgern und steigende Kohlendioxidkonzentrationen in der Atmosphäre, fordern immer mehr, einen schonenderen und umweltfreundlicheren Umgang mit vorhandenen Ressourcen. Eine effizientere Nutzung dieser Energieträger und der Ausbau regenerativer Energieanlagen ist nur in Verbindung mit der Speicherung von Energie, über kurze und längere Zeiträume, möglich bzw. sinnvoll. Das zeitliche Missverhältnis zwischen Energieangebot und -nachfrage muss ausgeglichen werden.

In dieser Arbeit werden die thermodynamischen Vorgänge in thermischen Schichtspeichern untersucht. Durch Einführen einer effektiven Wärmeleitfähigkeit, können auftretende freie Konvektionen im Speichermedium, in ein eindimensionales Wärmeleitungsproblem übergeführt werden. Zur Beurteilung der Strömungsform, während eines Be- bzw. Entladevorgangs, ist die Richardson-Zahl gut geeignet. Für hinreichend große Werte kann von einer idealen Kolbenströmung im gesamten Speicher ausgegangen werden. Als Modellgleichung resultiert die Transportgleichung der thermischen Energie. Mit Hilfe der Finite-Volumen-Methode, werden auf Basis dieser Erkenntnisse, numerische Modelle zur instationären Simulation solcher Speicher erstellt.

Unter vorgegebener Betriebsführung, ist dadurch eine Vorhersage der Temperaturverteilung und Effizienz in Abhängigkeit von der Zeit möglich. Integriert man das Modell beispielsweise, in ein bereits modelliertes thermisches Netz oder in ein häusliches Heizsystem, so kann das Verhalten des Gesamtsystems analysiert und beurteilt werden. Dadurch ist es möglich, die Investitionskosten des realen Speichers, den durch die Integration zu erwartenden Energieeinsparungen, gegenüberzustellen.

Abstract

Due to decreasing supplies of fossil fuels and rising carbon dioxide concentrations in the atmosphere, milder and more environmentally friendly use of these resources are becoming more and more important. A more efficient use of energy sources and the development of renewable energy systems is only in conjunction with the storage of energy, over short and long periods of time, possible and useful. The temporal mismatch between energy supply and demand have to be balanced.

In this paper, the thermodynamic processes in thermally stratified storages are studied. Occurring free convections of the storage medium, can be converted into a one-dimensional heat conduction problem, by establishing an effective thermal conductivity. The Richardson number can be used, to assess the type of flow during a loading or unloading period. For sufficiently large values, an ideal plug flow for the entire storage can be assumed. The model equation results as the transport equation of thermal energy. Based on the findings, the development of numerical models, for the transient simulation of such storages, is carried out by using the finite volume method.

With these models, a prediction of the temperature distribution and efficiency, as a function of time, can be made, for predefined conditions of operation. Integration of the storage model, for example in an already modeled thermal network or in a domestic heating system, makes it possible, to analyze the behavior of the overall system. Because of that, the anticipated cost reductions due to energy savings, can be contrasted with the investment costs of the real storage.

Nomenklatur

Symbole

Bezeichnung	SI-Einheit	Beschreibung
a	-	Koeffizient
a	m^2/s	Temperaturleitfähigkeit
A	m^2	Fläche
A	-	Hilfskoeffizient
b	-	Exponent
B	-	Hilfskoeffizient
c	J/kgK	spezifische Wärmekapazität
\bar{c}	J/kgK	spezifische integrale Wärmekapazität
C	-	Courant-Zahl
D	m	Speicherdurchmesser
f	-	Faktor
\mathbf{f}	m/s^2	Volumenkraft pro Masse
Fo	-	Fourier-Zahl
\mathbf{g}	m/s^2	Erdbeschleunigung
g	m/s^2	Erdbeschleunigung in z-Richtung
Gr	-	Grashof-Zahl
h	J/kg	spezifische Enthalpie
H	J	Enthalpie
H	m	Speicherhöhe
\dot{H}	W	Enthalpiestrom
i	-	Koordinate im Indexraum
k	$\text{W}/\text{m}^2\text{K}$	Wärmedurchgangskoeffizient
L	m	charakteristische Länge
m	kg	Masse
\dot{m}	kg/s	Massenstrom
n	-	Knotenanzahl

n	-	Anzahl
nt	-	Anzahl der Zeitschritte
p	Pa	Druck
Pe	-	Péclet-Zahl
Pr	-	Prandtl-Zahl
q	kg/m ³ s, kg/m ² s ² , W/m ³	Quelldichte
Q	J	Wärmemenge
\dot{Q}	W	Wärmestrom
R	m	Speicherradius
Ra	-	Rayleigh-Zahl
Re	-	Reynolds-Zahl
Ri	-	Richardson-Zahl
s	J/kgK	spezifische Entropie
S	J/K	Entropie
T	K	Temperatur
t	s	Zeit
\mathbf{u}	m/s	Geschwindigkeitsvektor
U	J	innere Energie
v	m ³ /kg	spezifisches Volumen
V	m ³	Volumen
w	m/s	Geschwindigkeit in z-Richtung
\mathbf{x}	m	Ortsvektor
x	-	reelle od. komplexe Variable
z	m	z-Koordinate

Bezeichnung	SI-Einheit	Beschreibung
α	W/m ² K	Wärmeübergangskoeffizient
β	1/K	räumlicher Wärmeausdehnungskoeffizient
δ	m	Wandstärke
ϵ	-	Verwirbelungsfaktor
ϵ	kg/m ³ , kg/m ² s, J/m ³	Dichte einer physikalischen Größe
ζ	m	konkrete z-Koordinate
η	-	energetischer Wirkungsgrad
ϑ	°C	Temperatur
Θ	-	dimensionslose Temperatur
λ	W/mK	Wärmeleitfähigkeit
ν	m ² /s	kinematische Viskosität
ξ	-	dimensionslose Wegkoordinate

ξ	J/kg	spezifische Exergie
Ξ	J	Exergie
ρ	kg/m ³	Dichte
τ	-	dimensionslose Zeit
ϕ	-	Exponent
ϕ	J/kg, J/kgK	spezifische allgemeine Zustandsgröße
Φ	J, J/K	allgemeine Zustandsgröße
$\dot{\Phi}$	W, W/K	Strom der allgemeine Zustandsgröße
φ	rad	Winkelkoordinate der Zylinderkoordinaten
ψ	-	exergetischer Wirkungsgrad

Indices

Index	Beschreibung
amb	Umgebung
ax	axial
A...E	Zählindices
bottom	Boden
ch	beladen
crit	kritisch
disch	entladen
down	hinunter
eddy	Wirbel
eff	effektiv
F	Fluid
flow	Massenstrom
i	Koordinate im Indexraum
i	allgemeiner Zählindex
in	Eintritt
in	innen
ini	anfangs
int	intern
j	zeitliche Diskretisierung
konv	konvektiv
KV	Kontrollvolumen
loss	Verlust

m	vollständig durchmischt
m	mittel
max	maximal
min	minimal
node	Knoten
out	Austritt
out	außen
p	isobar
pc	Phasenwechsel
rad	radial
ref	Referenz
S	Sättigungszustand
St	Speicher
top	Deckel
up	hinauf
v	isochor
W	Wand
0	Beginn des Zeitschritts
1	Ende des Zeitschritts
1...6	Zählindices
1/2	Halbwert
+	KV-Grenze darüber
-	KV-Grenze darunter
'	volumetrisch

Mathematische Operatoren

Bezeichnung	Beschreibung
Δ	Differenz
∇	Nabla Operator
O	Größenordnung
Σ	Summe

Akronyme

Akronym	Beschreibung
CFD	Computational Fluid Dynamics
FD	Finite Differenzen
FE	Finite Elemente
FV	Finite Volumen
IAPWS	International Association for the Properties of Water and Steam
KV	Kontrollvolumen
PCM	Phase Change Material
TDMA	Tridiagonalmatrix Algorithmus,
URBEM-DK	Urbanes Energie- und Mobilitätssystem - Doktoratskolleg

Inhaltsverzeichnis

Kurzfassung	ii
Abstract	iii
Nomenklatur	iv
Inhaltsverzeichnis	ix
1 Einleitung	1
2 Grundlagen	3
2.1 Klassifikation thermischer Speicher	3
2.2 Anforderungen	8
2.3 Wasser als Speichermedium	10
3 Thermodynamische Prozesse im Speicher	19
3.1 Die Bilanzgleichungen	19
3.2 Transportprozesse im ruhenden Speicher	21
3.3 Transportprozesse im strömenden Fluid	29
3.4 Analytische Lösung	38
4 Numerische Modelle	43
4.1 Diskretisierungsmethoden	43
4.2 Die Modellgleichung	45
5 Energie- und Exergieanalyse	56
6 Programmbeschreibung und Simulation	63
6.1 Beschreibung ausgewählter Funktionen	64
6.2 Demonstratives Beispiel	66
7 Zusammenfassung	75

<i>INHALTSVERZEICHNIS</i>	x
Abbildungsverzeichnis	77
Tabellenverzeichnis	79
Literaturverzeichnis	80
A Programmcode	83

Kapitel 1

Einleitung

Speicher für thermische Energie, umgangssprachlich auch als Wärmespeicher bezeichnet, ermöglichen es, das thermische Energieangebot im Zeitverlauf und in der Leistung, dem Bedarf anzupassen. Erst durch thermische Energiespeicher lassen sich viele Energiequellen, wie Sonnenenergie oder industrielle Abwärme, effizient nutzen. Sie helfen Anlagen effektiver zu betreiben und tragen damit zu einer Einsparung von Primärenergie bei. Auch im Bereich der Versorgungssicherheit und beim Transport thermischer Energie spielen sie eine wichtige Rolle.

Die Sonneneinstrahlung ist ein typisches Beispiel für stark wechselndes Angebot an Primärenergie. Daher stellt ein thermischer Speicher einen zentralen Baustein, zur solarthermisch gestützten Warmwasserbereitung und Heizung, dar. Zu Tageszeiten starker Sonneneinstrahlung, wird der Speicher, mit überschüssiger Energie, beladen, in Zeiten schwacher Einstrahlung, entladen. Saisonale Speicherung ist hier auch ein großes Thema. Etwa zwei Drittel der Solarstrahlung treten in den Monaten Mai bis Oktober auf, in denen nur knapp 25 % der jährlichen Heizenergie benötigt werden. [Fis05]

Ein wichtiges Beispiel ist der Einsatz von großen Speichern in Fernwärmenetzen, die aus Kraftwerken gespeist werden. Die kombinierte Erzeugung von elektrischer und thermischer Energie, ist eines der primärenergetisch günstigsten Verfahren rationeller Energienutzung überhaupt, da hier ein großer Teil, der bei der Stromerzeugung in kalorischen Kraftwerken anfallende Abwärme, genutzt werden kann. Der Strombedarf weist morgens und abends, für ein bis zwei Stunden, Spitzen auf, für deren Deckung teure Spitzenlastkraftwerke eingesetzt werden müssen. Wenn in diesen Zeiten die Produktion thermischer Energie eingestellt wird, lässt sich die Stromerzeugung der Kraftwerke steigern, und man kann auf den Bau solcher Kraftwerke verzichten. Die thermische Versorgung wird während dieser Zeiten aus großen Speichern gedeckt. Der Speicher wird als Kurzzeitspeicher, zum Ausgleich von Lastspitzen im Stromnetz, also Schwankungen der Nachfrage, verwendet.

Ein weiterer wichtiger Anwendungsfall, in Kombination mit einem Fernwärmenetz, ist die

saisonale Speicherung. Beispielsweise werden Müllverbrennungsanlagen über das ganze Jahr mit nahezu konstanter Last betrieben. Durch Saisonalspeicher ist es möglich, überschüssige Energie aus den Sommermonaten, in den nachfragestarken Wintermonaten bereitzustellen.

Der Anspruch an künftige Energieverteilungssysteme ist jedoch bedeutend höher, da durch die Forderung höherer Energieeffizienz des Gesamtsystems, thermische Energie auch dezentral, beispielsweise solarthermische Energie, geothermische Energie, oder auch nutzbare Abwärme von thermischen Prozessen in Gewerbe und Industrie, in Verteilnetze eingespeist werden soll. Im Zuge des, von der *Wiener Stadtwerke Holding AG* und der *Technischen Universität Wien*, eingerichteten Doktoratskollegs, mit dem Titel „*Urbanes Energie- und Mobilitätssystem (URBEM-DK)*“, sollen unter anderem Tools entstehen, die das dynamische Verhalten von thermischen Netzen mit angekoppelten Speichern simulieren können, um durch Modellrechnung, verschiedene Ausbauszenarien, hinsichtlich Energieeffizienz, Kosten, environmental footprint (etc.), bewerten zu können.

Das primäre Ziel dieser Arbeit ist nun, rechnergestützte Speichermodelle, die auf einfache Weise in das Modell des Fernwärmenetzes integrierbar sind, bereitzustellen.

Bei den zu modellierenden Speichern handelt es sich um thermische Schichtspeicher, die die Temperaturabhängigkeit der Massendichte des Speichermediums Wasser, ausnutzen. Es sind zwei Bauformen zu betrachten. Speicher, mit direkter Beladung am Speicherdeckel und direkter Entladung am Speicherboden, die im Betrieb von Energiequellen konstanter Temperatur gespeist werden. Für Energiequellen nicht konstanter Temperatur, ist ein Schichtladesystem nachzubilden, bei dem die Einspeicherhöhe, von der Beladetemperatur und der momentanen Temperaturverteilung im Speicher abhängt.

Zusätzlich soll ein Tool zur Verfügung stehen, das ein Gegenüberstellen und Beurteilen von Speichern, bzw. Speicherzyklen, hinsichtlich ihrer Effizienz, erlaubt.

Kapitel 2

Grundlagen

Dieses Kapitel führt in die physikalischen Grundlagen der thermischen Energiespeicherung ein. Die daraus resultierenden Technologien, werden hinsichtlich ihrer Vor- und Nachteile, den wichtigsten Anforderungen gegenübergestellt und bewertet. Aufgrund der technischen Bedeutung, der weiten Verbreitung und der Relevanz in diese Arbeit, wird im letzten Abschnitt der Schwerpunkt auf sensible Wasserspeicher gelegt.

2.1 Klassifikation thermischer Speicher

Primär wird aufgrund der thermophysikalischen Prinzipien zwischen fühlbaren oder sensiblen, latenten und thermochemischen Speichern unterschieden. Abb. 2.1 zeigt einen Überblick der vorhandenen Technologien, die in weiterer Folge auch näher erläutert werden.

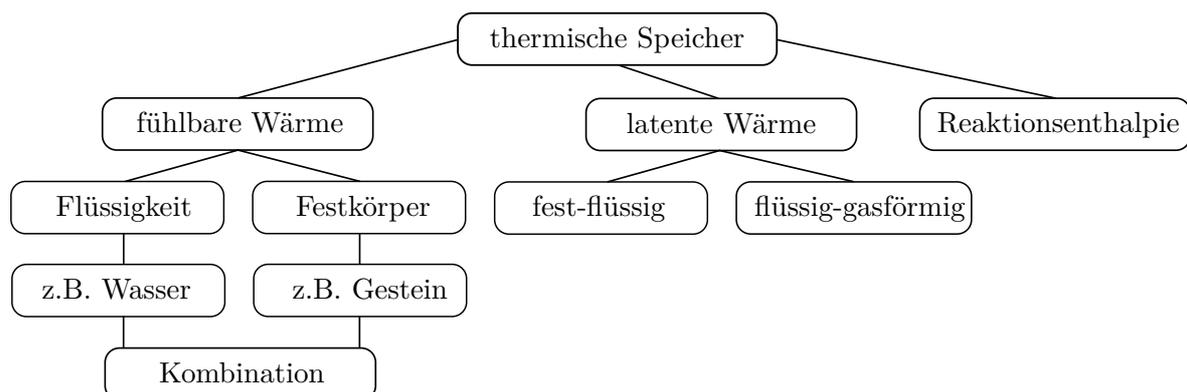


Abbildung 2.1: Einteilung thermischer Speicher

Weiters können thermische Speicher hinsichtlich verschiedenster Kriterien unterschieden werden [Hau13]:

- nach dem Temperaturniveau in Nieder-, Mittel- und Hochtemperaturspeicher. Aber auch „Kältespeicher“, sollte die Temperatur der Umgebung höher als das zu speichernde Temperaturniveau liegen.
- nach der Speicherdauer in Kurzzeit- (Stunden oder Tage), Langzeit- oder Saisonalspeicher (Wochen bis zu einem Jahr).
- nach der Position im Energiesystem in zentrale, dezentrale, stationäre und mobile Speicher.
- nach Speichermedien, Einsatzbereich, Speicherkapazität, Energiedichte (etc.).

Sensible Speicherung thermischer Energie

Als sensible Wärme, wird die Energieaufnahme oder -abgabe bezeichnet, die mit einer fühlbaren Änderung der Temperatur verbunden ist. Bei den betrachteten Speichervorgängen, geht man davon aus, dass diese bei konstantem Druck stattfinden, und es sich somit um eine isobare Zustandsänderung handelt. Deshalb führt eine Wärmezufuhr Q_{12} zu einer Änderung der Enthalpie ($H_2 - H_1$) in gleichem Maße und einem Temperaturanstieg nach folgender Gleichung:

$$Q_{12} = H_2 - H_1 = m \int_{T_1}^{T_2} c_p dT \quad (2.1)$$

Die Temperaturdifferenz ist neben der zugeführten Wärmemenge Q_{12} , auch von der Stoffmasse m und der spezifischen isobaren Wärmekapazität c_p abhängig.

Geht man von einer isochoren Zustandsänderung, einer Zustandsänderung bei konstantem Volumen aus, dann führt eine Wärmezufuhr Q_{12} zu einer Änderung der inneren Energie ($U_2 - U_1$). Unter Verwendung der spezifischen isochoren Wärmekapazität c_v , ergibt sich folgender Zusammenhang:

$$Q_{12} = U_2 - U_1 = m \int_{T_1}^{T_2} c_v dT \quad (2.2)$$

Bei inkompressible Medien gilt $c_p = c_v = c$, folglich nehmen Enthalpie und innere Energie den selben Wert an $H = U$.

Für einige gängige Stoffe ist die spezifische Wärmekapazität in Tab. 2.1 zusammengestellt. Da für die Speicherung thermischer Energie oft nicht die Masse, sondern das Volumen entscheidender ist, sind zusätzlich die Stoffdichte ρ und die volumetrische Wärmekapazität c'_p aufgeführt.

Sand, Kies und Gestein werden vorwiegend in Erdreich-Speichern genutzt, Beton bei thermischer Speicherung in Gebäuden. In Speichern zur Brauchwassererwärmung und in Heizungsanlagen kommt überwiegend Wasser zum Einsatz. Wärmeträgeröl wird in Solaranlagen mit

Parabollinnenkollektoren als Energietransport- und auch als Speichermedium eingesetzt. In Solarturmkraftwerken werden Salzschnmelzen und auch Natrium verwendet. [Fis05]

Tabelle 2.1: Spezifische und volumetrische Wärmekapazität von Speichermaterialien bei 20°C [Fis05]

Medium	Temperaturbereich °C	Spezifische Wärmekapazität c_p kJ/kgK	Volumetrische Wärmekapazität c'_p kJ/m ³ K	Dichte ρ kg/m ³
Wasser	0-100	4,19	4175	998
Kies, Sand	0-800	0,71	1.278-1.420	1.800-2.000
Granit	0-800	0,75	2.062	2.750
Beton	0-500	0,88	1.672-2.074	1.900-2.300
Ziegelstein	0-1.000	0,84	1.176-1.596	1.400-1.900
Eisen	0-800	0,47	3.655	7.860
Wärmeträgeröl	0-400	1,6-1,8	1.360-1.620	850-900
Kies-Wasser-Schüttung (37 Vol.-% Wasser)	0-100	1,32	2.904	2.200
Salzschnmelze (53KNO ₃ + 40NaNO ₂ + 7NaNO ₃)	150-450	1,3	1.970-1.725	2.561-2.243
Natrium	100-800	1,3	925-750	1.203-975

Latente Speicherung thermischer Energie

Als latente Wärme, bezeichnet man die bei einem Phasenübergang aufgenommene oder abgegebene Energiemenge. Dabei kommt es zu keiner merklichen Temperaturänderung.

Die in einem Latentspeichermaterial zwischen den Temperaturen T_1 und T_2 gespeicherte thermische Energie, setzt sich aus dem sensiblen Anteil unterhalb der Umwandlungstemperatur T_{pc} , der Umwandlungsenthalpie, und dem sensiblen Anteil oberhalb der Umwandlungstemperatur zusammen und wird für isobare Zustandsänderungen wie folgt berechnet.

$$Q_{12} = H_2 - H_1 = m \left(\int_{T_1}^{T_{pc}} c_{p1} dT + \Delta h_{pc} + \int_{T_{pc}}^{T_2} c_{p2} dT \right) \quad (2.3)$$

Bei Latentspeichermaterialien, die üblicherweise kurz PCM (Phase Change Material) bezeichnet werden, wird der Temperaturbereich T_1 bis T_2 klein gewählt, da das Hauptaugenmerk auf der latenten Energiespeicherung liegt, und eine möglichst konstante Ausspeichertemperatur meist wünschenswert ist. Folglich kann, wie in Abb. 2.2 ersichtlich, im Vergleich zu sensiblen Speichern deutlich mehr Energie bei gleicher Temperaturdifferenz gespeichert werden.

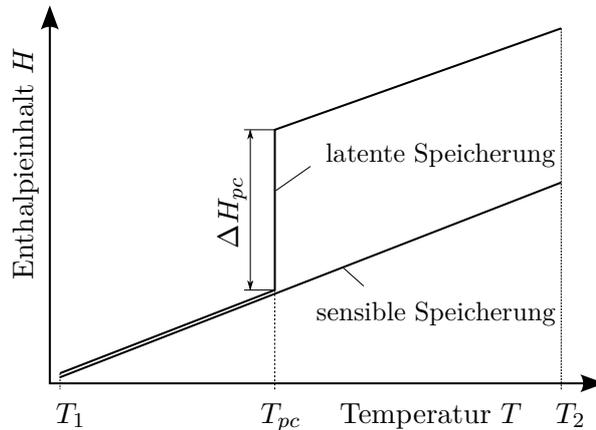


Abbildung 2.2: Abhängigkeit des Energieinhaltes eines thermischen Speichers von seiner Temperatur, für sensible und latente Speicherung

Tabelle 2.2: Umwandlungstemperatur und Umwandlungsenthalpien einiger Latentspeichermaterialien [Fis05]

Medium	Umwandlung	Umwandlungs- temperatur ϑ_{pc} °C	Umwandlungs- enthalpie Δh_{pc} kJ/kg	Spezifische Wärmekapa- zität c_{p1}/c_{p2} kJ/kgK
Wasser	fest/flüssig	0	335	2,1/4,19
	flüssig/gasförmig	100	2.540	4,19/1,86
Paraffine:				
Eicosan	fest/flüssig	36,6	243	1,94/2,08
Rohparaffin	fest/flüssig	34,3	142	
Fettsäuren:				
Laurinsäure	fest/flüssig	44	183	1,8/2,16
Myristinsäure	fest/flüssig	54	187	
Stearinsäure	fest/flüssig			
- rein		69,7	221	1,83/2,3
- technisch		64,8	203	
Salzhydrate:				
Na ₂ SO ₄ 10H ₂ O	fest/flüssig	32	241	
Na ₂ S ₂ O ₃ 5H ₂ O	fest/flüssig	48	201,2	
Ba(OH) ₂ 8H ₂ O	fest/flüssig	78	266,7	
Salzgemische:				
48NaCl/52MgCl ₂	fest/flüssig	450	432	0,9/1,0
67NaF/33MgF ₂	fest/flüssig	832	618	1,42/1,38

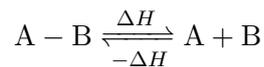
Wie man Tab. 2.2 entnehmen kann, konzentrieren sich die Anwendungen von Latentspeichermaterialien vor allem auf den Phasenübergang fest-flüssig. Hauptgrund dafür ist die enorme Volumenzunahme beim Phasenwechsel flüssig/gasförmig.

Obwohl die Energiedichte latenter Speicher um ein vielfaches höher ist als die von sensiblen, haben sie es noch nicht geschafft, sich als Standardtechnologie durchzusetzen. Hauptaufgaben an Forschung und Entwicklung sind zur Zeit: [Fis05]

- PCM weisen meist eine schlechte Wärmeleitfähigkeit auf, so dass die Wärmetransportwege im Speicher, im Hinblick auf eine nennenswerte Leistungsentnahme optimiert werden müssen.
- PCM ändern ihr Volumen beim Schmelzen und Erstarren. Durch die Bildung von Hohlräumen in der Phase mit dem geringeren Volumen, wird der Energietransport behindert.
- Einige Materialien (z.B. Salzhydrate) sind korrosiv gegen Metalle, so dass teure Speicherbehälter aus Edelstahl erforderlich sind. Andere Materialien sind nicht langzeitstabil. Ihr Verhalten kann durch Entmischung und durch chemische Veränderungen beeinträchtigt werden.

Thermochemische Speicherprozesse

Bei der chemischen Speicherung thermischer Energie, nutzt man die Reaktionsenthalpie ΔH aus. Dazu werden Reaktionen gesucht, die im technischen Maßstab reversibel sind. Die Vorgänge der Speicherbeladung (v.l.n.r) und -entladung (v.r.n.l) lassen sich mit der folgenden chemische Reaktionsgleichung schematisch darstellen.



Tab. 2.3 enthält beispielhaft einige Stoffpaare und die dazugehörigen Reaktionsenthalpien, die für thermochemische Energiespeicherung in Frage kommen.

Man kann erkennen, dass die Reaktionsenthalpien deutlich größer als die Umwandlungsenthalpien latenter Speichermaterialien sind. Dennoch betragen sie nur einen Bruchteil der Reaktionsenthalpien von den üblichen Verbrennungsvorgänge mit Sauerstoff. Fossile Brennstoffe sind genau genommen, nichts anderes, als im Laufe von Jahrmillionen entstandene chemische Energiespeicher. Die Reaktionstemperaturen liegen meist sehr hoch. Das bedeutet, dass zur Speicherung von Sonnenenergie konzentrierende Systeme notwendig sind. Der große Vorteil gegenüber sensiblen und latenten Speichern liegt allerdings darin, dass die Reaktionspartner bei Umgebungstemperatur beliebig lange gelagert werden können, ohne dass Energieverluste zu verzeichnen sind. [Fis05]

Tabelle 2.3: Reaktionspaare für die chemische Speicherung thermischer Energie [Fis05]

Reaktion	Reaktions- temperatur °C	Reaktions- enthalpie kJ/kg
Reversible Wasserabspaltung		
$\text{CaCl}_2 \cdot 2\text{H}_2\text{O} \rightleftharpoons \text{CaCl}_2 \cdot \text{H}_2\text{O} + \text{Dampf}$	258	555
$\text{Ca}(\text{OH})_2 \rightleftharpoons \text{CaO} + \text{Dampf}$	479	1.473
Reversible Zersetzung von Metallhydriden		
$\text{MgH}_2 \rightleftharpoons \text{Mg} + \text{H}_2$	280-500	2.885
Reversible Zersetzung von Salzen		
$\text{NH}_4\text{SO}_4 \rightleftharpoons \text{NH}_3 + \text{H}_2\text{O} + \text{SO}_3$	467	2.930
Adsorption von Wasser in Silicagel		
$\text{Silicagel} + \text{Dampf} \rightleftharpoons \text{Silicagel feucht}$	40-100	991
Reversible Zersetzung von Metallkarbonaten		
$\text{CaCO}_3 \rightleftharpoons \text{CaO} + \text{CO}_2$	837	1.780

Anzumerken sei, dass der Begriff der thermochemischen Energiespeicherung etwas unscharf ist, weil darunter oft eine Reihe von Speichersystemen verstanden werden, die nicht nur auf chemischen Reaktionen basieren. Dazu gehören: [Hau13]

- Adsorption: physikalische Bindung - z.B. durch Van-der-Waals-Kräfte - eines Stoffes an der Oberfläche des Adsorbens.
- Absorption: Einlagerung bzw. Lösung eines Stoffes in das freie Volumen der absorbierenden Phase.

Abschließend lässt sich sagen, dass die drei verschiedenen Speicherarten unterschiedlich weit entwickelt sind. Während sensible Speicher, im Wesentlichen kommerziell erhältliche Produkte sind, befinden sich latenten Speicher - wie oben schon erwähnt - noch in Entwicklung. Es sind zwar schon einzelne Produkte auf dem Markt, großer Bedarf herrscht aber in Sachen Materialforschung. Thermochemische Speicherkonzepte befinden sich noch in der Grundlagenforschung. Es sind kaum Produkte auf dem Markt. [Hau13]

2.2 Anforderungen

Im vorhergehenden Abschnitt wurden schon einige Anforderungen an thermische Speicher diskutiert. Dabei war der Fokus auf Energiedichte und Betriebstemperatur gerichtet, wodurch schon Anwendungseinschränkungen getätigt werden mussten. Weiters sind eine Vielzahl anderer Kriterien ebenso wichtig, um eine geeignete Speichertechnologie zu wählen.

Wie oben schon erwähnt, tritt bei sensiblen und latenten Speichern immer eine Temperaturdifferenz ($\vartheta_{St} - \vartheta_{amb}$) zur Umgebung auf, aus welcher ein Verlustwärmestrom \dot{Q}_{loss} resultiert.

$$\dot{Q}_{loss} = \int_{A_{KV}} k (\vartheta_{St} - \vartheta_{amb}) dA \quad (2.4)$$

Betrachtet man Glg. 2.4, so wird es bei zunehmender Temperaturdifferenz scheinbar immer wichtiger, den lokalen Wärmedurchgangskoeffizienten k und die Speicheroberfläche A möglichst klein zu halten. A_{KV} stellt die Oberfläche der betrachteten Bilanzgrenze dar, die in späteren Betrachtungen sinnvollerweise mit der Oberfläche der Speicherinnenwand übereinstimmt.

Der erste Punkt fordert, vor allem gute Dämmeigenschaften, d.h. eine geringe Wärmeleitfähigkeit λ_i und eine große Wandstärke δ_i , der maßgeblichen Isolationsschicht der Speicherhülle. Lokale Schwachstellen sind hier immer Anschlüsse für Rohre, Temperaturfühler (etc.).

$$\frac{1}{k dA} = \frac{1}{\alpha_{in} dA_{in}} + \sum_{i=1}^n \frac{\delta_i}{\lambda_i dA_{m,i}} + \frac{1}{\alpha_{out} dA_{out}} \quad (2.5)$$

Das wird klar, wenn man die allgemeine Bestimmungsglg. 2.5, des lokalen Wärmedurchgangskoeffizienten k , betrachtet. Dabei ist, α_{in} der Wärmeübergangskoeffizient zwischen Speichermedium und der inneren Schicht der Speicherhülle, dA_{in} das dabei betrachtete Flächenelement, α_{out} der Wärmeübergangskoeffizient zwischen Umgebung und der äußeren Schicht der Speicherhülle, dA_{out} das dabei betrachtete Flächenelement, n die Anzahl der Schichten der Speicherhülle und $dA_{m,i}$ das für die Wärmeleitung, in der Schicht i , maßgebliche Flächenelement.

Der zweite Punkt ist nur bedingt richtig. Entscheidend sind nämlich, die Verluste, bezogen auf das Speichervolumen V . Mit zunehmenden Abmessungen wächst zwar seine Oberfläche, das Volumen nimmt aber stärker zu, und die Verluste pro Volumen fallen. Das Verhältnis A/V wird ganz klar von der Speichergeometrie bestimmt und nimmt für die Kugel den minimalen Wert an. Auch aus Gründen der Festigkeit, ist die Kugelschale die ideale Form. Dennoch werden die meisten Speicher als stehende Zylinder ausgeführt. Grund dafür sind die einfachere Fertigung und Aufstellmöglichkeit. Für Heizungssysteme ist es also sinnvoll, große Speicher - die von mehreren Parteien genutzt werden - mit geeignetem Verhältnis der Abmessungen zu bauen. Aus dem selben Grund, werden Saisonspeicher oft als große Erdbeckenspeicher ausgeführt. Dafür spricht auch das Fallen der volumenbezogenen Baukosten mit steigendem Speichervolumen.

Ein entscheidender Punkt hinsichtlich Zusatzapparaturen, und damit den Investitionskosten, Betriebskosten (etc.), ist die Art des Energieaustausches. Flüssigkeiten können prinzipiell als Transport- sowie als Speichermedium für thermische Energie eingesetzt werden, wodurch ein direktes Ent- und/oder Beladen möglich ist. Das klassische Anwendungsbeispiel ist hier

wieder Wasser. Die Entnahme von Festkörpern aus dem Speicherkörper, ist im Vergleich zu Flüssigkeiten mit einem hohem Aufwand verbunden. Aber auch bei Flüssigkeitsspeichern ist oft ein direktes Be- bzw. Entladen nicht möglich. Das kann verschiedenste Gründe haben, beispielsweise wenn das Medium der thermischen Energiequelle bzw. -senke aus prozesstechnischen Gründen strikt vorgegeben ist. Man benötigt dann mehrere Stoffkreisläufe, mit all den notwendigen Apparaturen (Wärmetauscher, Umwälzpumpen), um die Energien ohne große Zusatzverluste hin und her transportieren zu können.

Der Energietransport im Speicher selbst, ist durch die Beweglichkeit der Flüssigkeiten meist auch ausgezeichnet. In Feststoffspeichern hingegen, erfolgt der Energietransport ausschließlich durch Wärmeleitung und damit recht langsam. Daher müssen Feststoffe im Speicher so vorliegen, dass die Wege für die Wärmeleitung möglichst kurz sind, und große Kontaktflächen für den Wärmetausch vorliegen.

Diese Sachverhalte haben entscheidenden Einfluss auf die Be- und Entladeleistung und stellen daher große Anforderungen an die Konstruktion.

Ergänzende Anforderungen:

- Verfügbarkeit und Preis des Speichermediums selbst
- Umweltverträglichkeit
- Sicherheit (toxisch, feuergefährlich, explosiv)
- Verträglichkeit mit Speicherhülle und Anbauten (Korrosivität)
- Lebensdauer
- Recyclebarkeit
- eine einfache Entsorgung

Für die Wahl des Speichermediums wird letztendlich die Summe aller Eigenschaften entscheidend sein, wobei einige allein schon genügen, um ein Material (z.B. der Temperaturbereich) auszuschneiden. [Din11], [Hau13], [Fis05]

2.3 Wasser als Speichermedium

Bisher wurde ein grober Überblick vorhandener Technologien thermischer Speicher geschaffen. Wie man dem entnehmen kann, handelt es sich um ein sehr weites Spektrum. Daraus ergeben sich eine Vielzahl von Anwendungsmöglichkeiten.

Von nun an soll der Fokus auf sensible Wasserspeicher gelegt werden. Diese sind in den verschiedensten Ausführungen sehr weit verbreitet. Gründe dafür wurden schon genannt.

Es folgt eine Zusammenfassung, bzw. Ergänzung, der zur thermischen Energiespeicherung relevanten Vor- und Nachteile von flüssigem Wasser: [Fis05]

- Es ist überall verfügbar, billig und leicht zu transportieren.
- Es ist nicht giftig und nicht aggressiv.
- Es liegen große Erfahrungen im Umgang mit Wasser vor, die Technologie wird von vielen Handwerkern beherrscht.
- Wasser besitzt eine hohe spezifische Wärmekapazität (vgl. Tab. 2.1) und eine geringe Wärmeleitfähigkeit.
- Wasser, ist in der Lage, thermische Energie konvektiv zu transportieren und erlaubt dadurch hohe Be- und Entladeleistungen (das Speichermedium selbst wird entladen).
- Wasser ist nicht nur Energieträger- und Energiespeichermedium, im Falle der Brauchwassererwärmung ist es gleichzeitig das Produkt, das benötigt wird.

Wasser hat neben seiner hohen spezifischen Wärmekapazität zwei weitere, sehr vorteilhafte Eigenschaften, die es erlauben, kaltes und warmes Wasser, in einem Speicher getrennt aufzubewahren.

- Seine geringe Wärmeleitfähigkeit erschwert einen Energietausch durch Wärmeleitung zwischen heißem und kaltem Wasser.
- Seine mit der Temperatur $> 4\text{ °C}$ abnehmende Dichte - 1 L Wasser von 90 °C Wasser wiegt etwa 3,5 % weniger als Wasser von 20 °C - erlaubt es, Wasser unterschiedlicher Temperatur so zu schichten, dass sich heißes und kaltes Wasser nicht vermischen.

An dieser Stelle muss eine Ergänzung zum vorhergehenden Abschnitt, hinsichtlich der Speichergeometrie, getätigt werden. Um den Wärmestrom von heißen zu kalten Schichten möglichst klein zu halten, sollte deren Kontaktfläche nicht zu groß ausgeführt werden. Dies spricht für hohe schlanke Speicher.

- Gegenüber latenten und thermochemischen Speichern sind die Energieverluste höher und die Energiedichte geringer.
- Der verwendbare Temperaturbereich bei Umgebungsdruck ist relativ klein. Wasser tritt in seiner flüssigen Form nur zwischen 0 °C und 100 °C auf. Der Exergieanteil ist demzufolge recht gering.

Betrachtet man die Gruppe der sensiblen Speicher alleine, so ist nur der letzte Punkt ein wesentlicher Nachteil. Der mögliche Temperaturbereich kann aber relativ einfach etwas ausgedehnt werden. Wird der Betriebsdruck erhöht, so steigt die Siedetemperatur an. Das wird

- wie schon der Name verrät - auch in den beiden Hochdruckspeichern der Fernwärme Wien ausgenutzt. Am Speicherboden herrschen 10 bar und an der Speicherdecke 6 bar [Wie]. D.h., an der Speicherdecke, wo sich auch die heißeste Schicht befindet, liegt die Siedetemperatur bei ca. 160 °C.

Konfigurationen

Grundsätzlich kann zwischen Einspeicher- und Zweispeichersystemen, die in Abb. 2.3 schematisch dargestellt sind, unterschieden werden. Das Ziel beider Systeme ist das gleiche, nämlich kaltes und warmes Fluid zu separieren und Mischungsvorgänge möglichst zu vermeiden. Das wird beim Betrachten der im System vorhandenen Exergie, also des technisch nutzbaren Anteils der Enthalpie, klar. Die Exergie hat bei Umgebungsbedingungen definitionsgemäß ihren Nullpunkt und zeigt progressives Wachstum mit steigender Temperatur.

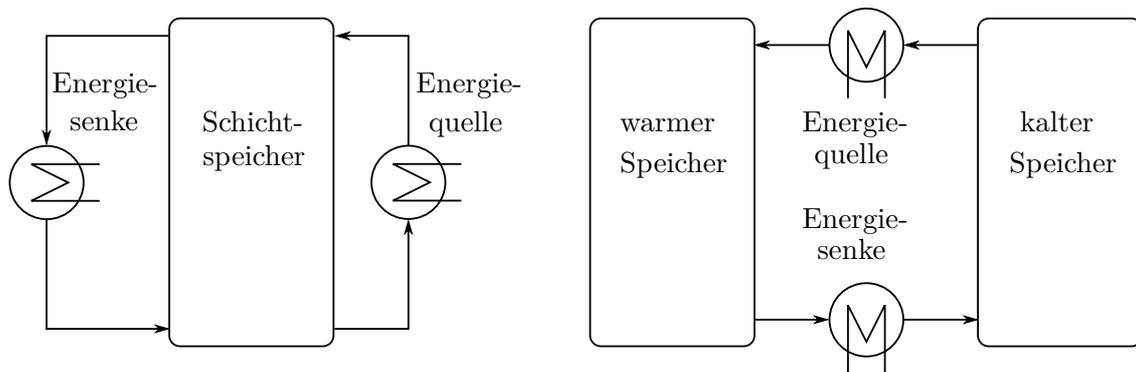


Abbildung 2.3: Einspeicher-, Zweispeichersystem

Das Zweispeichersystem weist das beste exergetische Verhalten auf, da es kaltes und warmes Wasser perfekt trennt. Beim Einspeichersystem mit ausgebildeter Temperaturschichtung, haben eine Vielzahl von Faktoren Einfluss auf den Exergieerhalt. Wie oben schon erwähnt, sind Dichteunterschiede und eine geringe Wärmeleitfähigkeit des Fluids wichtig. Aber auch das Speicherdesign, Dämmung, Gestaltung der Ein- u. Auslässe und die Betriebsart selbst, sind entscheidend. Mischt sich das Fluid völlig durch, so ändert sich der Energieinhalt nicht, sein Exergieanteil erreicht aber ein Minimum. [Pow13]

Die Ausbildung von Temperaturschichtung und die Einflüsse darauf, seien hier nur kurz erwähnt. Diese Effekte sind von großer Bedeutung und werden daher später in einem eigenen Abschnitt detaillierter behandelt.

Ein Nachteil des Zweispeichersystems ist, dass man das doppelte Speichervolumen benötigt, was wiederum zu größeren Investitionskosten führt. Außerdem sind solche Systeme schwerer zu kontrollieren, es muss viel Speichermedium umgewälzt werden und das größere Verhältnis A/V fördert die Energieverluste. [Din11]

Möglichkeiten der Be- und Entladung

Im Bereich der Kurzzeitspeicherung thermischer Energie über mehrere Tage, stellen Pufferspeicher sowie Warmwasserspeicher für den häuslichen Gebrauch, die wesentlichen beiden Anwendungsformen dar. Dabei kommen unterschiedliche Energietransfersysteme zum Einsatz.

Die konstruktiv einfachste Art der Systemintegration von thermischen Speichern, ist die direkte Be- und Entladung, ohne weiteren externen oder speicherinternen Wärmetauscher. Das Speichermedium wird hierbei über eine Energiequelle erwärmt und über eine Verbindungsleitung dem Speicher zugeführt. Beim Beladen, wird warmes Wasser von oben in den Speicher eingespeist, und dafür die entsprechende Menge kalten Wassers, aus dem unteren Speicherbereich entnommen. Diese Art der Energiezufuhr setzt man immer dann ein, wenn eine Trennung von Speicherwasser und Energieträger nicht notwendig ist.

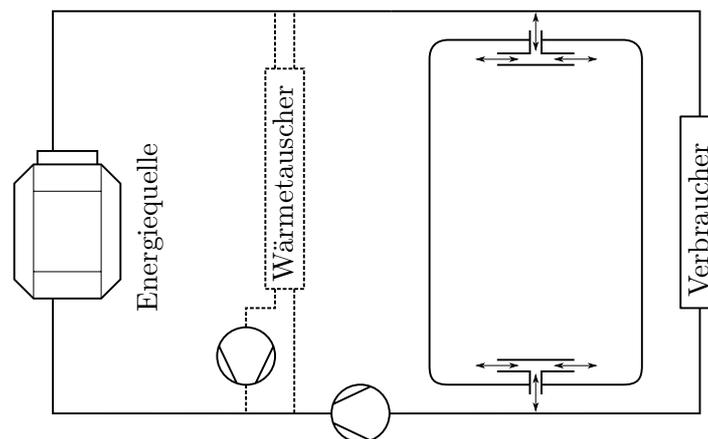


Abbildung 2.4: Warmwasserspeicher als Pufferspeicher mit direkter Be- und Entladung [Fis05]

Das warme Wasser schichtet sich aufgrund der durch die höhere Temperatur geringeren Dichte im oberen Bereich ein, und bewegt sich bei fortschreitender Beladung als Front, wie in Abb. 2.5 ersichtlich, langsam nach unten. Die Entladung wird mit umgekehrter Strömungsrichtung von unten nach oben durchgeführt.

Damit die Ausbildung der Schichtung im Speicher nicht beeinträchtigt wird, sind neben der Begrenzung der Fließgeschwindigkeit auch konstruktive Maßnahmen notwendig, um den Impuls des einströmenden Fluids abzubauen. [Fis05] Außerdem sollten die Be- und Entladeöffnungen möglichst nahe bei Speicherboden und -deckel angeordnet sein, um sogenannte Totwasserzonen klein zu halten. Diese reduzieren das effektive Speichervolumen. [Din11]

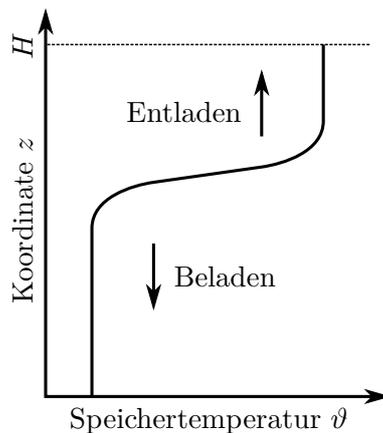


Abbildung 2.5: Bewegung der Temperaturfront [Fis05]

Bei Verwendung unterschiedlicher Energieträger zur Versorgung des Speichers ist eine konstruktive Trennung auf der Be- und/oder Entladeseite vorzusehen. Anwendungsbeispiele sind Speicher in thermischen Solaranlagen, in denen eine Wärmeübertragung, von dem in der Kollektoranlage erwärmten Wasser/Frostschutzgemisch, an das Speichermedium erfolgt. Eine indirekte Beladung kann über speicherintegrierte oder externe Wärmetauscher erfolgen.

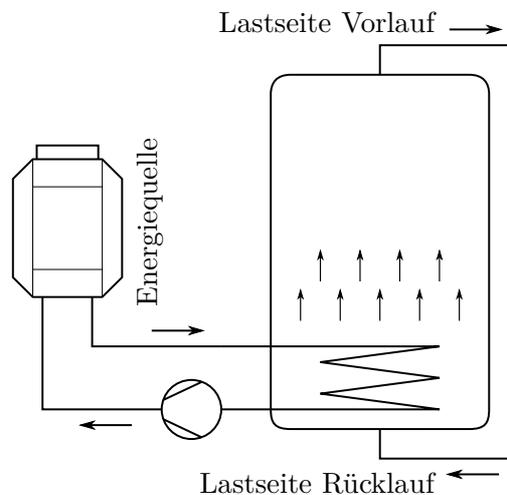


Abbildung 2.6: Warmwasserspeicher mit indirekter Energiezufuhr über Wärmetauscher [Fis05]

Abb. 2.6 zeigt einen Warmwasserspeicher mit indirekter Energiezufuhr über Wärmetauscher. Bei der Beheizung erwärmt sich das Wasser außen am Wärmetauscher und strömt aufgrund der Dichteverringerng nach oben. Die sich ausbildende Strömung sorgt für einheitliches Aufheizen des Bereichs oberhalb des Wärmetauschers. Unterhalb des Wärmetauschers erfolgt eine Erwärmung äußerst langsam, so dass dieser Raum für die Speicherung verloren ist. Bei der indirekten Entladung wird der Wärmetauscher im oberen Bereich des Speichers angeordnet. Das umliegende Wasser kühlt ab und strömt aufgrund der Dichtezunahme nach unten. [Fis05]

Die oben gezeigte konventionelle Art der direkten Beladung fordert konstante Beladetemperatur und sollte möglichst nahe bei der Temperatur der heißesten Schicht liegen. Bei Beheizung des Speichers über eine Solaranlage, deren Kollektoren mit konstantem Massenstrom durchflossen werden, verändert sich das Temperaturniveau im Speicherladekreis mit der Eintrittstemperatur des Kollektorfluids und wechselnder solarer Einstrahlung. Bei konventionellen Bauarten kommt es zu einer Aufschichtung von Wasser geringeren, auf Wasser höheren Temperaturniveaus. Das bezeichnet man als inverse Schichtung. Das Absinken des kälteren, schwereren Wassers führt zu einer Störung bzw. einem Abbau der Schichtung im Speicher. Es sind also spezielle Konstruktionen notwendig, um, bei wechselnder Speisetemperatur, das einströmende Wasser an die Stelle im Speicher zu leiten, die die selbe Temperatur hat. Ein Lösungsansatz ist der in Abb. 2.7 dargestellte „schwebende Schlauch“, der ein dem Wasser entsprechendes spezifisches Gewicht aufweist.

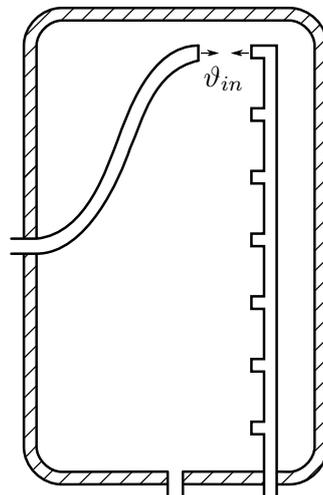


Abbildung 2.7: Warmwasserspeicher mit idealer Einspeisung

Mit schwankendem Temperatur- und Dichteniveau des zugeführten Wassers zum einen, und dem Speichermedium zum anderen, bewegt sich der Schlauch entlang der Speicherhöhe im Segment entsprechender Temperatur. Eine dieses Funktionsprinzip nutzende Einrichtung ist die Schichtladelanze, die ebenfalls in Abb. 2.7 zu sehen ist. Der bewegliche Schlauch wird hierbei durch eine fest stehende Beladeeinrichtung ersetzt. An den Stützen, in den unterschiedlichen Höhen, befinden sich leicht bewegliche Rückschlagklappen. In Abhängigkeit der Dichtedifferenz, von im Beladerohr aufsteigendem und im Speicher befindlichem Wasser, werden die Klappen geöffnet und durchströmt. [Fis05]

In Kapitel 3.3 wird dieses Funktionsprinzip nochmals genauer erläutert.

Pufferspeicher in Fernwärmenetzen

Die Integration von thermischen Speichern in Nah- und Fernwärmenetzen trägt erheblich zur Verbesserung der Brennstoffausnutzung, und mit der Kompensation von Lastspitzen, auch zur Verbesserung des Anlagenbetriebs bei.

In Abb. 2.8 ist ein Beladevorgang des im November 2013 fertig gestellten Hochdruckspeicher in Wien Simmering ersichtlich. Die beiden Speicher sind über eine Pumpstation mit Vor- und Rücklauf des Fernwärmenetzes verbunden: [Wie]

- 2 Speicherbehälter: Durchmesser 14 m, Höhe 44,5 m, Volumen 5.600 m³ pro Behälter, Energieinhalt 830 MWh, Solldruck am Speicherboden 10,3 bar
- Ausgleichsbehälter: Durchmesser 8 m, Höhe 27 m, Volumen 1.250 m³
- Vorlauf Fernwärmenetz: 95 – 153 °C, 13 – 18 bar
- Rücklauf Fernwärmenetz: 58 – 70 °C, 15 – 20 bar

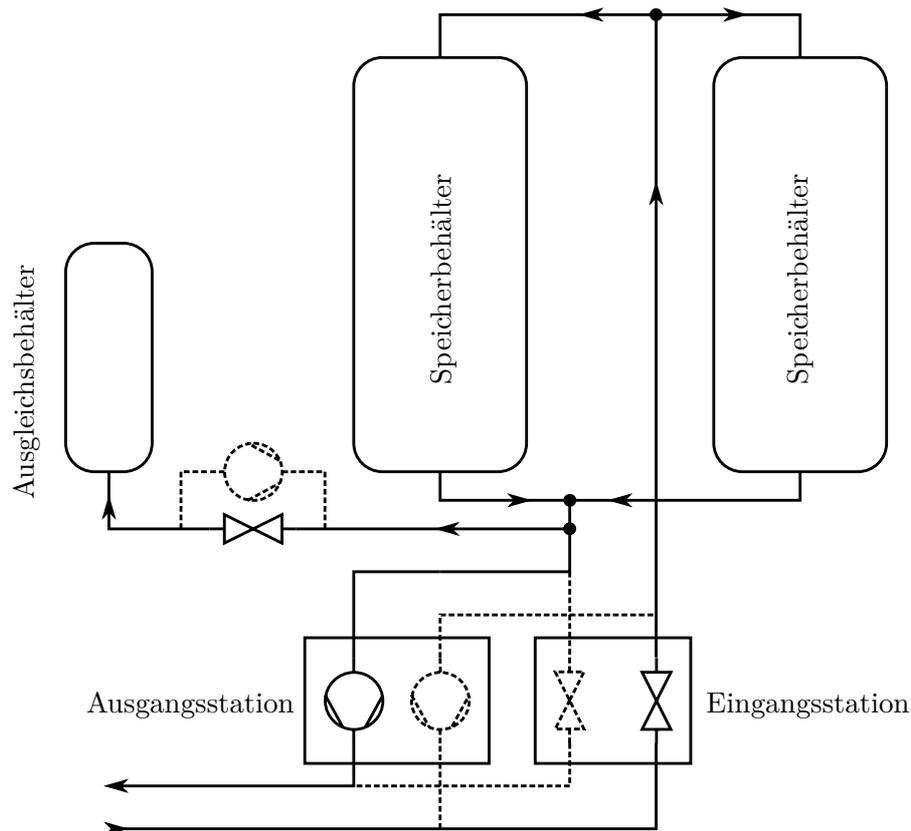


Abbildung 2.8: Integration der Speicheranlage Simmering in das Fernwärmenetz

Speicherkonzepte zur Langzeitspeicherung

Bei solarem Heizen mit hohem solarem Deckungsanteil spielt die saisonale Speicherung thermischer Energie eine entscheidende Rolle, da hier, die in den Sommermonaten bei hoher Einstrahlung generierte Energie, in den Winter, mit seinem hohen Heizenergiebedarf verschoben werden muss. Aufgrund der wenigen Speicherzyklen pro Jahr, muss auf die Energieverluste besonderer Wert gelegt werden. Daher bedarf es, wie oben schon erwähnt, sehr großer Speichervolumina um große Energiemengen effizient und wirtschaftlich zu speichern. Außerdem zeigt sich mit zunehmender Größe ein deutlicher Rückgang der spezifischen Baukosten.

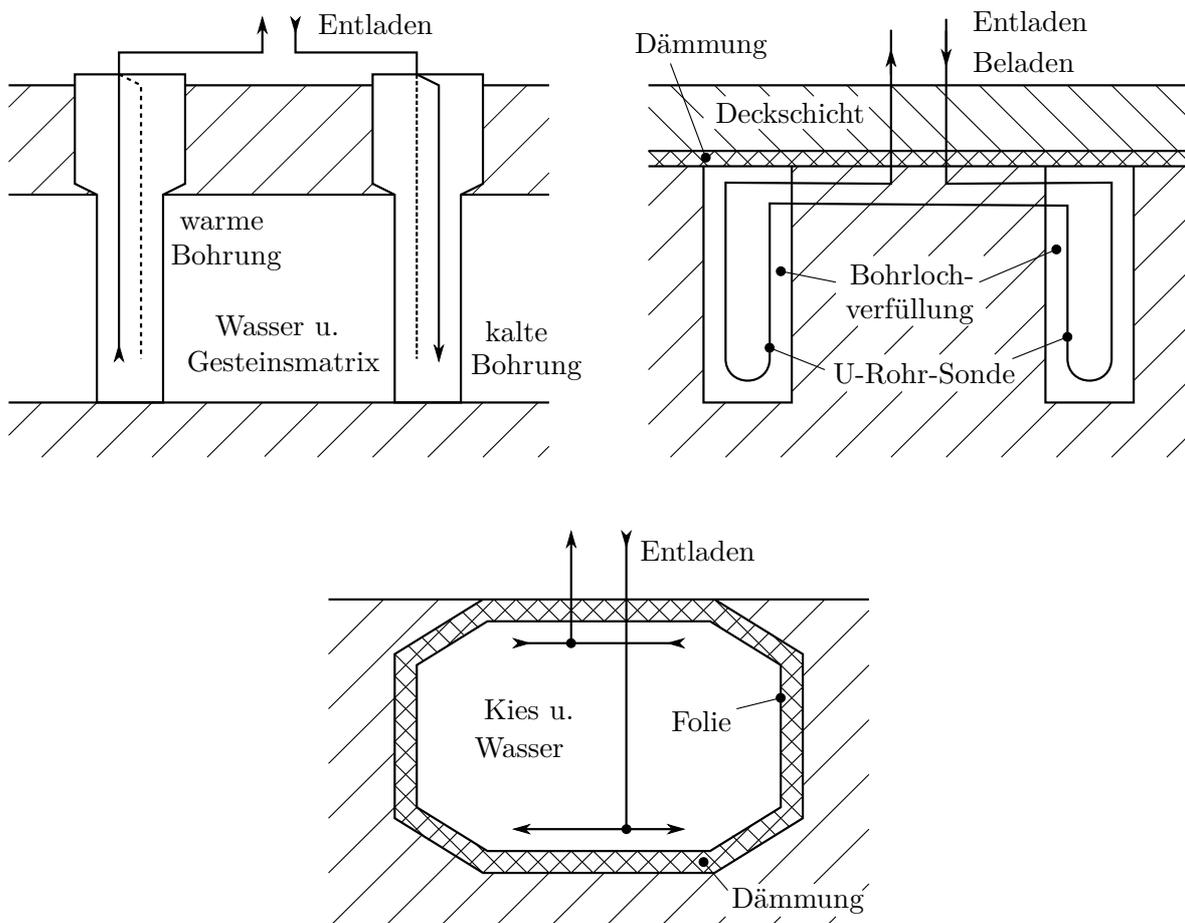


Abbildung 2.9: Aquifer-, Erdsonden- und Erdbeckenspeicher

Die drei gängigsten Untergrundspeicherkonzepte sind hier kurz beschrieben und in Abb. 2.9 schematisch dargestellt: [Hau13]

- Beim Aquiferspeicher wird Wasser einschließlich der Gesteinsmatrix, als Speichermedium genutzt. Über Brunnen wird das Wasser gefördert, aufgeheizt oder abgekühlt und wieder in denselben Aquifer zurück geleitet. Dabei spielen für die Machbarkeit die

hydrogeologischen und hydrochemischen Bedingungen am Standort eine entscheidende Rolle.

- Erdsondenspeicher nutzen den gewachsenen Untergrund als Speichermedium, das Be- und Entladen erfolgt über Wärmetauscher im Untergrund. Auch hier hängt die Machbarkeit an den geologischen Standortgegebenheiten.
- Eine nur geringe Abhängigkeit von den Standortbedingungen haben Erdbeckenspeicher oder Behälterspeicher. Es wird eine entsprechend große Grube ausgehoben und mit geeigneter Dämmung und einer wasserdampfdichten Folie ausgekleidet. Anschließend wird das Becken mit Kies aufgefüllt und mit Abdichtfolie, sowie einer dicken Lage Dämmschicht abgedeckt. Die Kiesschüttung wird mit Wasser gefüllt, das zusammen mit dem Kies das Speichermedium darstellt und gleichzeitig als Energieträgermedium dient.

Kapitel 3

Thermodynamische Prozesse im Speicher

Dieses Kapitel untersucht den Massen- und Energietransport in thermischen Schichtspeichern. Ausgehend von der allgemeinen räumlichen Problemstellung, wird zuerst der ruhende, und dann der durchströmte Speicher betrachtet. Dabei sollen vertretbare Modellierungsansätze gefunden werden, welche eine Vereinfachung des räumlichen Gleichungssystems zulassen. Ziel ist es, den Rechenaufwand zu reduzieren, ohne die physikalischen Effekte zu verzerren, bzw. zu verfälschen. Abschließend werden für die gefundene Modellgleichung, analytische Lösungen, zweier recht einfacher Anfangswert-Randwertprobleme präsentiert.

3.1 Die Bilanzgleichungen

Strömungsmechanische Probleme können mittels Bilanzierung extensiver Zustandsgrößen, über geeignete Kontrollvolumina, der Finite-Volumen-Methode, beschrieben werden. Ausgehend von der allgemeinen Transportgleichung, werden die Formulierungen speziell für Masse, Impuls und Energie vorgestellt.

Die allgemeine Transportgleichung

Die allgemeine Form der Transportgleichung ergibt sich aus dem *Reynoldsschen Transport-Theorem*. Es führt die zeitliche Änderung der Dichte ϵ einer physikalischen Größe in einem Volumen, das sich mit der Strömung mitbewegt (substantielles Volumen), auf Änderungsraten in einem laborfesten Volumen zurück. In differentieller Form erhält man so die Erhaltungsgleichungen im Laborsystem, in der Form:

$$\frac{\partial \epsilon}{\partial t} + \nabla \cdot (\epsilon \mathbf{u}) = q \quad (3.1)$$

Hierbei kann ϵ die Dichte der Masse, des Impulses, der Energie (etc.) sein. Die zu ϵ gehörige

Stromdichte ist $\epsilon \mathbf{u}$. Z.B. ist $\rho \mathbf{u}$ die Massenstromdichte. Ihr Betrag gibt an, wieviel Masse pro Zeit und Fläche, durch eine Fläche senkrecht zu \mathbf{u} tritt. Alle Größen hängen i.a. vom Ort \mathbf{x} und der Zeit t ab. Die Dichte ϵ an einem festen Punkt \mathbf{x} , kann sich ändern, indem die Größe ϵ durch die Strömung \mathbf{u} an den Punkt \mathbf{x} heran- oder von ihm wegtransportiert wird. Diese Änderung der Strömung wird durch die negative Divergenz der Stromdichte $-\nabla \cdot (\epsilon \mathbf{u})$ beschrieben. Alle anderen Prozesse, die eine Änderung von ϵ bewirken, sind in der Quelledichte q zusammengefasst. Zu diesen Prozessen zählen insbesondere der diffusive Transport, der noch zu spezifizieren wäre. [Kuh10]

Kontinuitätsgleichung

Setzt man nun für ϵ die Massendichte ρ , unter Berücksichtigung der Massenerhaltung $q = 0$, in Glg. 3.1 ein, so erhält man die Kontinuitätsgleichung kompressibler Strömungen:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (3.2)$$

Für inkompressible Fluide und kleine Strömungsgeschwindigkeiten ist $\rho = \text{const.}$ bzw. $\frac{\partial \rho}{\partial t} = 0$, somit vereinfacht sich die Kontinuitätsgleichung zu:

$$\nabla \cdot (\rho \mathbf{u}) = 0 \quad (3.3)$$

Impulsbilanz für Newtonsche Fluide

In ähnlicher Weise ergibt sich die Bilanzgleichung der Impulsänderung, wenn man für ϵ die Impulsdichte $\rho \mathbf{u}$ einsetzt.

Hierbei ist jedoch zu beachten, dass die Quelledichte des Impulses nicht verschwindet. Fluidbewegungen werden auch durch Druckvariationen erzeugt. Der negative Druckgradient $-\nabla p$ stellt dabei eine Kraft pro Volumen dar, und muss als Quellterm für die Impulsdichte berücksichtigt werden.

Bei realen Fluiden müssen viskose Effekte ebenfalls im Quellterm berücksichtigt werden. *Newtonsche Fluide*, z.B. Wasser, weisen ein ideal-elastisches, also lineares, unelastisches Fließverhalten auf. Dabei ist die Schergeschwindigkeit proportional der Scherspannung. Zur genaueren Betrachtung, der Einführung eines viskosen Spannungstensors, der diesem Materialgesetz genügt, sei an dieser Stelle auf [Oer12] verwiesen.

Mit der Kontinuitätsgleichung inkompressibler Fluide (Glg. 3.3), erhält man nun aus Glg. 3.1 die vereinfachte *Navier-Stokes-Gleichung*:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f} \quad (3.4)$$

Hier ist ν die kinematische Viskosität. Außerdem wurde noch eine Volumenkraft pro Masse \mathbf{f} eingeführt. \mathbf{f} kann z.B. die Erdbeschleunigung \mathbf{g} , also dem daraus hervorgerufenen hydro-

statischen Druck, beinhalten. Aber auch Auftriebseffekte. Zur Berechnung der thermischen Konvektion bei geringen Strömungsgeschwindigkeiten, kann das Fluid, wie oben schon erwähnt, bezüglich der Strömung als inkompressibel aufgefasst werden, nicht aber bezüglich der thermischen Expansion. Man berücksichtigt die Variation der Dichte dann nur durch die *Oberbeck-Boussinesq-Approximation* $\mathbf{f} = -\beta \Delta\vartheta \mathbf{g}$. In dieser Näherung ist β der thermische Ausdehnungskoeffizient und $\Delta\vartheta$ die Abweichung von der mittleren Temperatur. [Kuh10]

Energieerhaltung

Wenn für ϵ die Dichte der Gesamtenergie eingesetzt wird, kommt man auf die häufig verwendete Näherung, in der ϑ die abhängige Variable ist:

$$\rho c_p \left(\frac{\partial \vartheta}{\partial t} + \mathbf{u} \cdot \nabla \vartheta \right) = \nabla \cdot (\lambda \nabla \vartheta) + q \quad (3.5)$$

Hierin, wurde nur der wichtigste Quellterm der thermischen Energie, die thermische Diffusion, konkret eingesetzt. Der Diffusionsterm ergibt sich aus dem *Fourierschen Gesetz* für die Wärmestromdichte. q fasst eventuelle Verluste, bzw. Quellen durch Wärmeleitung über die Systemgrenzen, und Erwärmung aufgrund innerer Reibung, zusammen. [Kuh10]

Glg. 3.4 und Glg. 3.5 sind nichtlineare partielle Differentialgleichungen zweiter Ordnung, und besitzen im Allgemeinen, nur für sehr einfache Systeme, analytische Lösungen. Für komplexere Anwendungsfälle, z.B. instationäre Randbedingungen, gewinnen numerische Lösungsmethoden an Bedeutung. Diese sind üblicherweise mit erheblich größerem Rechenaufwand verbunden. Um diesen zu mindern, sollen die Vorgänge in thermischen Schichtspeichern genauer untersucht, und weitere vertretbare Vereinfachungen der Gleichungen, gefunden werden.

3.2 Transportprozesse im ruhenden Speicher

Um gewisse Vorgänge im Inneren eines zylindrischen Speichers besser zu verstehen, betrachtet man einen Wasserbehälter mit nicht notwendiger homogener Temperaturverteilung im ruhenden Zustand, also ohne Be- oder- Entladung. Es lassen sich folgende Temperaturtransportprozesse beobachten:

- Das Temperaturprofil baut sich aufgrund der Wärmeleitfähigkeit des Wassers ab.
- Ein Wärmestrom durch die Behälterwand in vertikaler Richtung, erwärmt kältere, tiefer liegende Schichten und induziert damit einen Energietransport durch freie Konvektion.
- Die Wärmeverluste an die Umgebung führen zum Abbau des vertikalen Temperaturprofils.

Daran kann man schon erkennen, dass auch in den Betriebsperioden ohne Be- oder Entladung, das Vorhandensein von Strömungen im Inneren des Speichers nicht auszuschließen ist. In den folgenden Abschnitten sollen diese Prozesse genauer untersucht und Möglichkeiten zur Quantifizierung gesucht werden.

Ausbildung des vertikalen Temperaturprofils im adiabaten Speicher

Wird ein thermischer Speicher mit nichtleitender Wandung, im ruhenden Zustand betrachtet, so kann angenommen werden, dass der Druck p nur von der Höhe z abhängt (die z -Achse habe stets vertikale Orientierung):

$$\nabla p = \rho \mathbf{g} \quad (3.6)$$

$$p = \text{const} - \rho g z \quad (3.7)$$

Falls an irgend einer Stelle in der Potentialebene, ein davon verschiedener Druck herrscht, entsteht daraus eine Bewegung, bis sich die Flüssigkeit wieder im mechanischen Gleichgewicht befindet. Andererseits kann dann die Dichte nur von der Höhe abhängen. Wenn also $p = p(z)$ und $\rho = \rho(z)$, dann kann für die Temperatur, die durch Druck und Dichte eindeutig festgelegt ist, auch nur $\vartheta = \vartheta(z)$ gelten. Falls nun die Temperatur im Allgemeinen eine Funktion aller Raumkoordinaten ist, so kann sich die Flüssigkeit nicht im mechanischen Gleichgewicht befinden. Mit anderen Worten: Verharrt ein Fluid in einem Behälter im ruhenden Zustand, so ist die Temperaturverteilung in allen horizontalen Ebenen konstant.

Dieser Sachverhalt ist auch anschaulich klar, wenn z.B. die Masse in einem Volumenelement in einer Flüssigkeitsschicht, eine höhere Temperatur als die aller Nachbarvolumenelemente hat. Es ergibt sich eine Auftriebskraft, aufgrund der geringeren Dichte in diesem Volumenelement. Ist die Auftriebskraft genügend groß gegen die Zähigkeit im Fluid, so strebt der Masseninhalt nach oben, bis die Kräfte ausgeglichen sind. Man spricht von freier Konvektion. [Joh02]

Es stellt sich nun die Frage, wie groß der Temperaturunterschied werden muss damit freie Konvektion einsetzt. Als entscheidende Größe wird die Rayleigh-Zahl Ra , das Produkt aus Prandtl- Pr und Grashof-Zahl Gr herangezogen.

$$Ra = Pr Gr = \frac{g\beta}{\nu a} \Delta\vartheta L^3 \quad (3.8)$$

Dabei ist a die Temperaturleitfähigkeit des Fluid, $\Delta\vartheta$ der gesuchte Temperaturunterschied und L die charakteristische Länge.

Wenn die Rayleigh-Zahl unterhalb eines kritischen Wertes Ra_{crit} für das Fluid bleibt, ist die Energieübertragung primär durch Wärmeleitung gegeben. Wenn sie den kritischen Wert übersteigt, ist die Energieübertragung primär durch Konvektion gegeben.

Ein für diesen Fall vergleichbares Experiment wird in [Inc07] vorgestellt. Beim Betrachten eines Hohlraums mit rechteckigen Querschnitt, setzte freier Konvektion bei $Ra_{crit} = 1.708$ ein. Für die charakteristische Länge, wird im Experiment die kleinere Abmessung des Rechteckquerschnitts verwendet, hier soll es aber der Durchmesser D des Behälters sein.

$$Ra = \frac{g \beta}{\nu a} \Delta\vartheta D^3 > Ra_{crit} = 1.708 \quad (3.9)$$

Setzt man die Stoffwerte für Wasser bei 70 °C ein, so erhält man nach der Umformung:

$$\Delta\vartheta > \frac{1,9936 \cdot 10^{-8}}{D^3} \text{ m}^3 \text{ K} \quad (3.10)$$

Das bedeutet, dass bereits für sehr kleinen Speicherdurchmessern, sich keine stabilen Inversionslagen mit merkbarem Temperaturunterschied bilden können. In [Joh02] wird als Ursache dafür, die sehr geringe kinematische Viskosität von Wasser, gegenüber anderer Flüssigkeiten genannt.

Beim Speichermodell in [New95] werden benachbarte Schichten ab einem Temperaturunterschied von $\Delta\vartheta = 10^{-3}$ K gemischt.

Abbau der Schichtung durch Wärmeleitung des Fluids im adiabaten Speicher

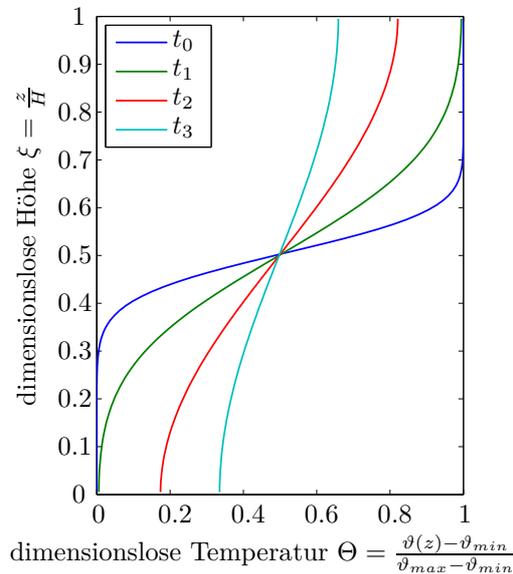


Abbildung 3.1: Abbau der Schichtung durch Wärmeleitung des Fluids im adiabaten Speicher

Es wurde eben gezeigt, dass der ruhende, adiabte Zylinder als eindimensionales Problem betrachtet werden kann. Somit reduziert sich die allgemeine Energiebilanz in Glg. 3.5 mit $a = \frac{\lambda}{c_p \rho}$ zur *Fourierschen Wärmeleitungsgleichung* für $\vartheta = \vartheta(z, t)$:

$$\frac{\partial \vartheta}{\partial t} = a \frac{\partial^2 \vartheta}{\partial z^2} \quad (3.11)$$

Abb. 3.1 veranschaulicht qualitativ die Lösung der Wärmeleitungsgleichung ($t_0 < t_1 < t_2 < t_3$). Ein vorhandener Temperaturgradient $\frac{\partial \vartheta}{\partial z}$ wird gemäß der obigen Glg. allmählich abgebaut, bis er schließlich für $t \rightarrow \infty$ gegen 0 geht.

Abbau der Schichtung durch Wärmeverluste an die Umgebung

Ausgangspunkt ist wieder die allgemeine Energiebilanz mit Glg. 3.5, im eindimensionalen Fall. Unter den Annahmen dass, der Behälter sich in Ruhe befindet, sein Deckel und Boden adiabat sind, die Behälterwände nur in radiale Richtung Wärme leiten, das beinhaltete Wasser mit $\lambda = 0$ keine Wärme leitet, erhält man folgende Glg.:

$$\rho c_p \frac{\partial \vartheta}{\partial t} = -\frac{1}{V} (kA)_{rad} (\vartheta - \vartheta_{amb}) \quad (3.12)$$

Dabei ist (kA) die Wärmeverlustrate in radiale Richtung.

In Abb. 3.2 ist für $t_0 < t_1 < t_2 < t_3$ der Effekt von diesen radialen Wärmeverlusten auf das Temperaturprofil ersichtlich. Die heißeren Regionen, werden aufgrund der größeren Temperaturdifferenz zur Umgebung stärker abgekühlt, als die kälteren.

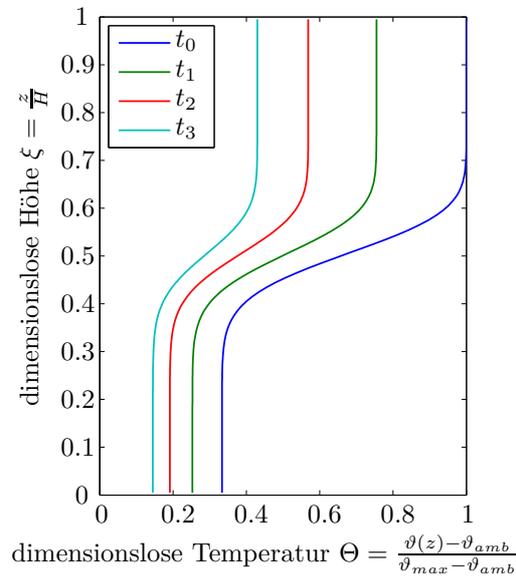


Abbildung 3.2: Abbau der Schichtung durch Wärmeverluste an die Umgebung

Zur experimentellen Bestimmung speicherspezifischer Wärmeverlustraten $(kA)_{St}$ sind in [Deu12] geeignete Prüfverfahren zu finden.

Konkrete Werte verschiedener Speicher können z.B. [Dru97] oder [Dru98] entnommen werden. Diese liegen für relativ gut isolierte Kleinspeicher in der Größenordnung von wenigen W/K . Allerdings ist damit die Vergleichbarkeit von Speichern, hinsichtlich Güte der Dämmung, auf Speicher mit gleich großer Oberfläche eingeschränkt. Bezieht man die Wärmeverlustrate auf die Gesamtoberfläche des Speichers, so ergeben sich Werte zwischen 0,1 und 1,0 W/m^2K . [Joh02]

Abbau der Schichtung durch Wärmeleitung im Behältermantel

In realen Speichern, kann die Annahme von innen, ideal gedämmten Behälterwänden, nicht aufrecht erhalten werden. Es ist denkbar, dass durch eine, im Vergleich zur Flüssigkeit, sehr gute Wärmeleitung, und ein daraus entstehender vertikaler Wärmestrom in der Behälterwand, die weiter unten liegenden Wasserschichten erwärmt. Lt. Glg. 3.10 würden dann sofort diese Inversionslagen, durch Einsetzen von freier Konvektion, abgebaut werden. Die Durchmischung des Speichers wäre die Folge. Diese Art von Wirbel ist im unteren kalten Speicherbereich zu erwarten.

Andererseits kühlt sich im oberen heißen Bereich ein Teil des Wassers, aufgrund der Wärmeverluste an die Umgebung, ab und sinkt im Randbereich nach unten. An der vertikalen Grenze zum kalten Speicherwasser mischt sich ein Teil dieses abströmenden Wassers, erwärmt dort die kalte Schicht und strömt im Zentralbereich zurück. Dieser Wirbel ist mit entgegengesetzter Strömungsrichtung zu erwarten. [Joh02]

Außerdem müssten Schichten nahe am Speicherdeckel, durch die lokal viel größere Kontaktfläche mit der Behälterwand stärker abkühlen, als Schichten weiter darunter. Schichtinversion bzw. verstärkte Mischvorgänge wären die Folge.

In der Literatur finden sich experimentelle Untersuchungen, über den Einfluss von Wandmaterialien, Wandstärke und Dämmung auf die Temperaturschichtung. Dabei werden vertikale Temperaturprofile erzeugt und der Speicher im Stillstand beobachtet. In [Joh02] sind einige dieser Experimente erklärt, deren Kernaussagen hier kurz zusammengefasst sind:

- Horizontale Temperaturunterschiede gleichen sich innerhalb von Minuten aus.
- Die Verluste im Bereich des Deckels erzeugen offenbar freie Konvektionsströmungen an der horizontalen Oberfläche, sodass Temperaturgradienten im oberen Speicherbereich abgebaut werden. Insbesondere bleibt der homogen initialisierte Speicher homogen.
- Axiale Wärmeleitung durch das Wasser und durch die Behälterwandung heben zunächst das Temperaturniveau im unteren Speicherteil an, erst später, wenn die axiale Wärmeleitung klein wird gegen die Verluste, sinkt die Temperatur unten wieder.
- Je dicker die Wandstärke bei gleicher Dämmstärke, desto stärker ist der Abbau der Schichtung durch axiale Transportprozesse geprägt. Die innenliegende Dämmung zeigt eine bessere Erhaltung der Schichtung.

- Geschwindigkeitsmessungen im unterem kalten Speicherbereich zeigten, dass Fluidelemente in Randnähe mit maximaler Geschwindigkeit steigen. Diese Strömungsschicht nimmt nur etwa 5 % des Radius vom Behälter ein. Am Rand selbst, gilt die Haftbedingung. Im Zentrum des Speichers fließt das Fluid relativ homogen und langsam wieder nach unten. Die Abwärtsbewegung erstreckt sich fast über die gesamte Speicherfläche. Radiale Geschwindigkeiten treten äußerst langsam auf. Insgesamt ergibt sich ein Kreiswirbel (vgl. Abb. 3.3).

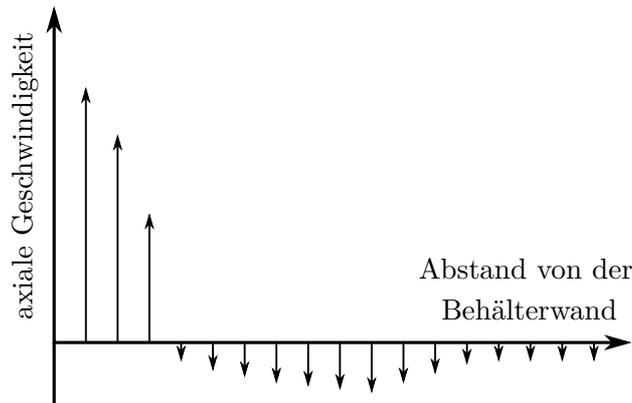


Abbildung 3.3: Qualitatives Geschwindigkeitsprofil im kalten Speicherquerschnitt [New95]

All diese Ergebnisse legen es nahe, den durch die freie Konvektion verursachten Transport thermischer Energie, in ein eindimensionales Wärmeleitungsphänomen zu reduzieren. In [Joh02] wird eine effektive Wärmeleitfähigkeit λ_{eff} zur Charakterisierung der axialen Transportprozesse vorgestellt. Dabei wird auch ein geeignetes Verfahren zur Quantifizierung dieses Parameters angegeben. Bildet man die Energiebilanz über das Kontrollvolumen in Abb. 3.4, so erhält man folgende Gleichung:

$$\begin{aligned}
 H_2 - H_1 &= A_{ax} \rho c_p \int_0^\zeta (\vartheta(z, t_2) - \vartheta(z, t_1)) dz = \int_{t_1}^{t_2} (\dot{Q}_{eff} - \dot{Q}_{rad}) dt \\
 &= A_{ax} \lambda_{eff} \int_{t_1}^{t_2} \left. \frac{\partial \vartheta(t)}{\partial z} \right|_{z=\zeta} dt - k_{rad} D \pi \zeta (\vartheta_m - \vartheta_{amb})(t_2 - t_1)
 \end{aligned} \tag{3.13}$$

Dabei ist ϑ_m die über Zeit ($t = t_1 \dots t_2$) und Ort ($z = 0 \dots \zeta$) gemittelte Speichertemperatur. Die thermophysikalischen Eigenschaften von Fluid und Behälterwand werden als konstant angenommen.

Sind experimentelle Daten eines bestehenden Speichers vorhanden, kann die effektive Wärmeleitfähigkeit damit relativ einfach bestimmt werden. Ein geeignetes Prüfverfahren wird in [Deu12] vorgestellt.

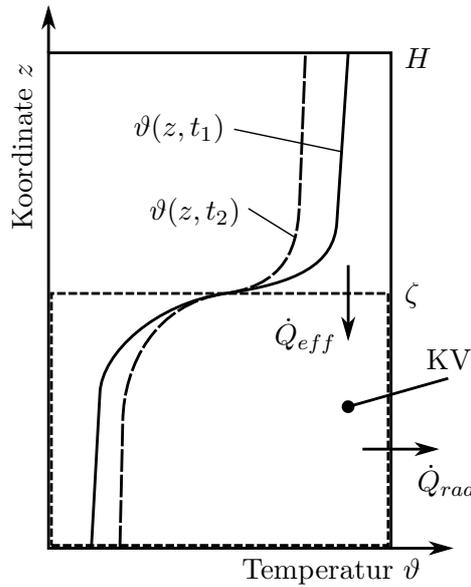


Abbildung 3.4: Ermittlung der effektiven Wärmeleitfähigkeit aus experimentellen Daten

Sollten keine experimentellen Daten oder Erfahrungswerte für λ_{eff} vorhanden sein, kann die in [Haf99] oder [New95] vorgestellte Schätzung verwendet werden: Dabei wird vorausgesetzt, dass Wand- und Fluidtemperatur für gleiches z gleich groß sind.

$$\dot{Q} = \dot{Q}_F + \dot{Q}_W = A_{ax} \frac{\lambda_F}{\Delta z} (\vartheta_2 - \vartheta_1) + A_W \frac{\lambda_W}{\Delta z} (\vartheta_2 - \vartheta_1) \quad (3.14)$$

$$\lambda_{eff} = \lambda_F + \lambda_W \frac{A_W}{A_{ax}} \quad (3.15)$$

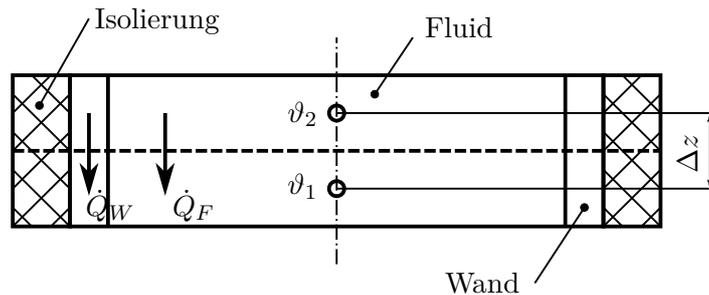


Abbildung 3.5: Näherung der effektiven Wärmeleitfähigkeit

λ_{eff} steigt mit zunehmender Behälterwandstärke, oder größerer Wärmeleitfähigkeit der Behälterwand und mit zunehmender Dämmstärke. Das bedeutet jedoch nicht, dass ein schlecht gedämmter Speicher mit entsprechend geringerem λ_{eff} starke Temperaturgradienten länger aufrecht erhalten kann, als ein gut gedämmter Speicher. Das heißt lediglich, dass bei gut gedämmten Speichern der Diffusionsprozess eine größere Rolle spielt.

Generell kann man sagen, dass für Großspeicher ohne Einbauten, die effektive Wärmeleitfähigkeit in etwa bei der des Wassers liegt. Grund dafür ist das größer werdende Verhältnis von Speicherhöhe zu Wandstärke. Bei kleinen Speichern variiert der Wert in der Größenordnung von zwei- bis fünffacher Wärmeleitfähigkeit des Wassers. [Joh02]

Konkrete Werte für λ_{eff} von verschiedenen Speichertypen können z.B. [Dru97] oder [Dru98] entnommen werden.

Freie Konvektion am Speicherdeckel

Die eben eingeführte effektive Wärmeleitfähigkeit λ_{eff} erfasst zwar die Effekte der freien Konvektion aufgrund von Wärmeleitung im Behältermantel, nicht aber, bzw. zu langsam, die Effekte durch die größeren Wärmeverluste am Speicherdeckel. Dieser Modellansatz würde Schichtinversionen in den höchsten Speicherregionen zulassen, was aber den experimentellen Beobachtungen widerspricht.

Eine naheliegende Lösung wäre, die effektive Wärmeleitfähigkeit von der Höhe $\lambda_{eff} = \lambda_{eff}(z)$ abhängig zu machen, und in diesen Regionen um einige Zehnerpotenzen zu erhöhen. Dadurch wird verstärkt thermische Energie aus den unteren Schichten zugeführt um die Inversionen abzubauen.

Eine weitere recht anschauliche Möglichkeit, wäre ein Eingriff durch kalorische Mischen der betroffenen Region, mit den darunter liegenden Schichten ($c_p = \text{const.}, \rho = \text{const.}$):

$$\rho c_p \int_{\zeta}^H \vartheta(z) dz = \rho c_p \vartheta_m (H - \zeta) \tag{3.16}$$

Diese Glg. ist nur iterativ zu lösen, da die z-Koordinate ζ , ab der gemischt werden soll, und die Temperatur ϑ_m , die sich nach dem Mischvorgang einstellt, unbekannt sind. Eine geeignete Vorgehensweise zum Einbinden dieses Mischvorgangs in das Modell, wird später gezeigt.

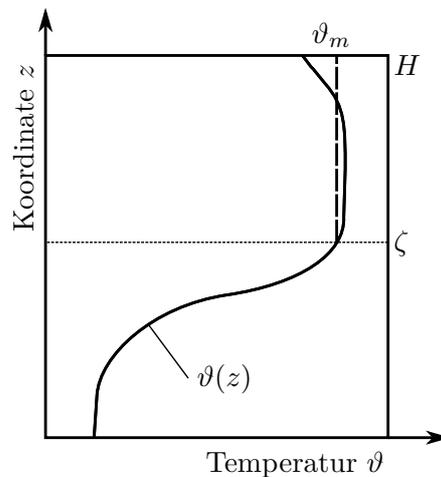


Abbildung 3.6: Kalorisches Mischen

Streng genommen könnten am Speicherboden ebenfalls Schichtinversionen auftreten. Wenn die Umgebungstemperatur ϑ_{amb} oberhalb der kalten Wassertemperatur am Speicherboden liegt, stellt sich ein Wärmestrom von außen nach innen ein, der die unterste Schicht am stärksten erwärmt.

3.3 Transportprozesse im strömenden Fluid

In diesem Abschnitt soll erörtert werden, wie sehr sich Be- und Entladevorgänge auf den Erhalt der Temperaturschichtung auswirken. Darüber hinaus soll geklärt werden, ob und unter welchen Voraussetzungen eine wie oben durchgeführte eindimensionale Betrachtung des Problems legitim ist.

Dimensionslose Kennzahlen zur Charakterisierung der Schichtung und Beladeeffizienz

Unter der Annahme, das Problem sei zylindersymmetrisch, reduziert sich das Gleichungssystem aus Kapitel 3.1 um eine Dimension. Änderungen der Geschwindigkeit und Temperatur des Fluids, werden nur in radialer und axialer Richtung zugelassen:

$$\frac{\partial}{\partial \varphi} = 0 \quad (3.17)$$

Um Strömungs- und Wärmetransportcharakteristiken im adiabaten zylindrischen Speichern beim Beladevorgang zu erkennen, wird in [Hah98] eine numerische Untersuchung mit Hilfe eines zweidimensionalen Modells präsentiert.

Es konnte gezeigt werden, dass Strömungs- und Temperaturfelder vor allem stark von der modifizierten Richardson-Zahl¹ abhängen:

$$Ri = \frac{Gr}{Re^2} = \frac{g H \beta (\vartheta_{ch} - \vartheta_{ini})}{w_{in}^2} \quad (3.18)$$

Dabei ist ϑ_{ch} die Temperatur des an der Tankoberseite mit der Geschwindigkeit w_{in} vertikal einströmenden Wassers und ϑ_{ini} die Anfangstemperatur im gesamten Speicher.

Für sehr kleine Ri Werte überwiegt der Impuls des einströmenden Fluids, gegenüber den Auftriebskräften aufgrund des Dichteunterschieds. Der Großteil des einströmenden Fluids wird direkt am Speicherboden wieder entladen, da sich entlang der Zylinderachse ein vertikaler Strahl ausbildet, dessen Temperatur über die gesamte Höhe annähernd bei ϑ_{ch} liegt. In den äußeren Regionen des Zylinders entsteht dadurch ein großer Wirbel.

¹Die dimensionslosen Kennzahlen Ri und Pe sind in ihrer Definition modifiziert. In diesem Unterabschnitt wird für die charakteristische Länge L die Speicherhöhe H und für die Geschwindigkeit jedoch die Einströmgeschwindigkeit w_{in} eingesetzt.

Bei großen Richardson-Zahlen, wird der Impuls gegenüber den Auftriebseffekten klein. In diesem Fall kann heißes Wasser nicht bis zu den kalten Regionen am Boden durchbrechen. Es sammelt sich im oberen Bereich des Tanks und drückt allmählich das sich darunter befindliche Fluid kolbenflussartig nach unten. Die Temperaturverteilung liegt in Form von horizontalen Isothermen vor. Eine kleine Zone mit großem Temperaturgradienten zeugt von gutem Schichtungsverhalten.

Zur Beurteilung des Aufrechterhaltens der thermischen Schichtung während eines Beladevorgangs wird die Beladeeffizienz η_{ch} herangezogen:

$$\eta_{ch} = \frac{\int_0^{t_{ch}} \dot{m}_{in} c_p [\vartheta_{ch} - \vartheta_{disch}(t)] dt}{m c_p (\vartheta_{ch} - \vartheta_{ini})} \quad (3.19)$$

Sie ist definitionsgemäß der Quotient aus der eingespeicherten Netto-Energie und der maximal speicherbaren Energie, beim einmaligen vollständigen Austausch des Speicherinhalts. Mit \dot{m}_{in} als Belademassenstrom, ϑ_{disch} als Temperatur des Wasser das unten wieder austritt, m als die Masse des gesamten Speicherinhalts und ϑ_{ini} als konstant verteilte Temperatur zu Beginn des Beladevorgangs.

Die numerische Untersuchung mit unterschiedlichen Speicherabmessungen und Betriebszuständen brachte folgende Korrelation für die Beladeeffizienz:

$$\eta_{ch} = 1 - 0,206 Ri^{-0,57} Pe^{-0,49} Fo^{-0,74} (H/D)^{-1,1} \quad (3.20)$$

mit

$$Pe = \frac{w_{in} H}{a} \quad \text{und} \quad Fo = \frac{a t}{H^2} \quad (3.21)$$

Demzufolge wird die Beladeeffizienz, und damit der Erhalt der Schichtung durch die modifizierte Richardson- Ri , die modifizierte Péclet- Pe , die Fourier-Zahl Fo und dem Verhältnis aus Speicherhöhe zu -durchmesser beeinflusst. Eine Zusammenfassung der Kernaussagen aus [Hah98] folgt:

- Mit zunehmender Temperaturdifferenz ($\vartheta_{ch} - \vartheta_{ini}$), wird der Auftriebseffekt, gegenüber dem Impuls des einströmenden Strahls, immer stärker und die Beladeeffizienz steigt. Ab einem Temperaturunterschied von 20 K, nimmt sie einen annähernd konstanten Wert an. Nur mehr 2 bis 3 % gehen aufgrund von Wärmeleitung zwischen Schichten unterschiedlicher Temperatur verloren.
- Für eine gegebene Temperaturdifferenz fällt η_{ch} mit zunehmender Einströmgeschwindigkeit, da sich die Fluide im Bereich der Zuströmung intensiv mischen. Ab einer Temperaturdifferenz von 20 K können die Effekte der Einströmgeschwindigkeit allerdings vernachlässigt werden.

- Der Einfluss des Verhältnisses H/D ist vergleichsweise klein.
- Nimmt die Richardson-Zahl einen größeren Wert als 0,25 an, so bewegt sich η_{ch} zwischen 97 und 98 %. 2 bis 3 % gehen aufgrund der schwachen Wärmeleitung zwischen Schichten unterschiedlicher Temperatur verloren.
- Der Einfluss der Péclet-Zahl Pe kann für $Ri > 0,25$ vernachlässigt werden. Das Strömungsfeld gleicht einer idealen eindimensionalen Kolbenströmung.
- Für $Ri < 0,1$ ist das Strömungsfeld stets zweidimensional.
- Die Fourier-Zahl Fo repräsentiert die dimensionslose Zeit beim Beladeprozess. Mit steigender Fo steht dem Fluid mehr Zeit zum Abbau von Temperaturgradienten durch Wärmeleitung zur Verfügung und die Beladeeffizienz fällt.

Ergänzende Bemerkungen aus [Joh02]:

- Experimente mit verschiedenen horizontalen Einlässen haben nachgewiesen, dass ab $Ri > 0,5$ der Mischungsvorgang im Speicher innerhalb einer Schicht verläuft, die weniger als 5 % der Speichhöhe ausmacht.
- Theoretische Untersuchungen eines rechtwinkligen, ideal isolierten Speichers haben gezeigt, dass ab $Ri > 1$ der horizontal, (unrealistisch) auf der gesamten Behälterbreite zweidimensional stattfindende Einströmprozess zu einer kolbenflussartigen vertikalen Durchströmung des Speichers führt.
- Ferner gilt, dass kleine Richardson-Zahlen, wenn sie allein durch kleine Werte $\Delta\vartheta$ erreicht werden, nicht so kritisch sind, weil im Fall einer stärkeren Durchmischung, keine wesentlichen Auswirkungen auf die Temperatur in der Schicht darunter entstehen.

Die ideale Kolbenströmung

Durch die obigen Erkenntnisse kann man mit der Voraussetzung $Ri > 0,25$ davon ausgehen, dass sich ein idealer Kolbenfluss im Speicher ausbildet. Dann reduziert sich das in Kapitel 3.1 angegebene, allgemeine, dreidimensionale Gleichungssystem, zu einem nur noch von z abhängigen, eindimensionalen System. Die Zeitabhängigkeit ist nach wie vor gegeben.

Die Kontinuitätsgleichung inkompressibler Fluide (Glg. 3.3) ergibt sich zu:

$$\frac{\partial w}{\partial z} = 0 \quad (3.22)$$

Das bedeutet, dass für einen festen Zeitpunkt t die Geschwindigkeit w für alle Orte z konstant ist:

$$w(z) = \text{const.} \quad (3.23)$$

Aus der Navier-Stokes-Gleichung (Glg. 3.4) folgt

$$\frac{\partial w}{\partial t} = -\frac{1}{\rho} \frac{\partial p}{\partial z} \quad (3.24)$$

unter den Annahmen: Druckunterschiede aufgrund der Gravitationskraft sind gegenüber den Drücken, welche durch die Pumpe verursacht werden, zu vernachlässigen. Viskose Effekte spielen ebenfalls eine untergeordnete Rolle. Auftriebseffekte werden in der Energiebilanz (Glg. 3.5) durch Einführen einer effektiven Wärmeleitfähigkeit λ_{eff} berücksichtigt.

$$\rho c_p \left(\frac{\partial \vartheta}{\partial t} + w \frac{\partial \vartheta}{\partial z} \right) = \lambda_{eff} \frac{\partial^2 \vartheta}{\partial z^2} + q \quad (3.25)$$

Dabei ist w als Randbedingung explizit vorgegeben und damit keine unbekannt Funktion mehr. Der Quellterm q beinhaltet nach wie vor die Wärmeverluste über die Systemgrenzen.

Mischungsvorgänge beim Einströmen

Auch wenn für größere Richardson-Zahlen eine ideale Kolbenströmung im Speicher vorhergesagt wird, kommt es bei Einströmvorgängen am Boden oder Deckel des Zylinders zu turbulenten Verwirbelungen. Diese sind am Ort der Einströmung am stärksten ausgeprägt und klingen mit zunehmender Entfernung davon sehr schnell ab.

Um diese Mischungsvorgänge zu erfassen, führt Oppel in [Opp86] einen vom Ort abhängigen Verwirbelungsfaktor $\epsilon = \epsilon(z)$ ein und erweitert den Diffusionsterm seiner Modellgleichung damit:

$$\epsilon a \frac{\partial^2 \vartheta}{\partial z^2} = \frac{a + a_{eddy}}{a} a \frac{\partial^2 \vartheta}{\partial z^2} = \frac{\lambda + \lambda_{eddy}}{\rho c_p} \frac{\partial^2 \vartheta}{\partial z^2} \quad (3.26)$$

Es werden also die räumlichen Temperaturtransportprozesse aufgrund der Verwirbelungen wieder in einen dadurch verstärkten eindimensionalen Wärmeleitungsprozess übergeführt. Sind keine Turbulenzen vorhanden $\epsilon = 1$ bzw. $\lambda_{eddy} = 0$ geht Glg. 3.26 wieder in ihre ursprünglich Form über.

Nun stellen sich folgende Fragen: Wie stark sind die Mischungsvorgänge am Einlass? Wie schnell und auf welche Weise klingen sie ab?

Durch Vergleich von Simulationsergebnissen unter Verwendung unterschiedlicher Abklingverhalten (linear, exponentiell, hyperbolisch) mit experimentellen Daten, fand man für die hyperbolische Funktion mit den Hilfskoeffizienten A und B die beste Übereinstimmung:

$$\epsilon = \frac{A}{i} + B \quad (3.27)$$

Einströmung unten - Entladevorgang:

$$A = \frac{\epsilon_{in} - 1}{1 - \frac{1}{n}} \quad \text{und} \quad B = \epsilon_{in} - A \quad (3.28)$$

Einströmung oben - Beladevorgang:

$$A = \frac{1 - \epsilon_{in}}{1 - \frac{1}{n}} \quad \text{und} \quad B = 1 - A \quad (3.29)$$

Dabei ist ϵ_{in} der maximal auftretende Verwirbelungsfaktor am Einlass, n die Knotenanzahl des numerischen Modells und i die diskretisierte Koordinate der Knoten.² In Abb. 3.7 kann man sehen, dass der Verlauf des Verwirbelungsfaktors $\epsilon(z)$ von der gewählten Knotenanzahl n abhängt. In dem Fall ist n eine Stellschraube zur Validierung des Rechenmodells.

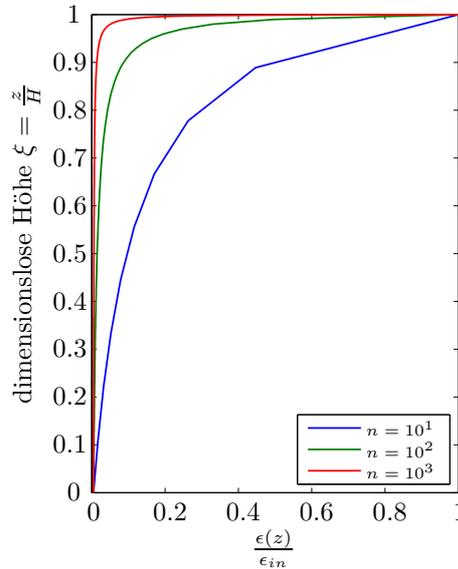


Abbildung 3.7: Verwirbelungsfaktor $\epsilon = \epsilon(z, n)$

Zur Quantifizierung des Verwirbelungsfaktors an der Einströmstelle wurden experimentelle Untersuchungen durchgeführt. Folgende Korrelation als Funktion von Re/Ri konnte gefunden werden: [Opp86]

$$\epsilon_{in} = a \left(\frac{Re}{Ri} \right)^b \quad (3.30)$$

Dabei charakterisieren der Koeffizient a und der Exponent b die Einlassgeometrie. Re repräsentiert das Ausmaß der verursachten Turbulenzen und Ri die Fähigkeit zum Abbau dieser.

²Vorgriff: Zum Lösen partieller Differentialgleichungen wird oft das Finite-Volumen-Verfahren verwendet. Das gesamte Rechenggebiet wird in eine endliche Zahl n von Teilvolumina mit den diskretisierten Koordinaten $1 \dots i \dots n$ unterteilt.

In [Zur91] wurden experimentelle Untersuchungen unterschiedlicher Einlassgeometrien durchgeführt und verglichen. Abb. 3.8 zeigt die verschiedenen Anordnungen: (1) radiale Einströmung von der Seite, (2) radiale Einströmung von der Seite mit darunter liegender perforierter Platte, (3) axiale am Speicherdeckel aufprallende Einströmung.

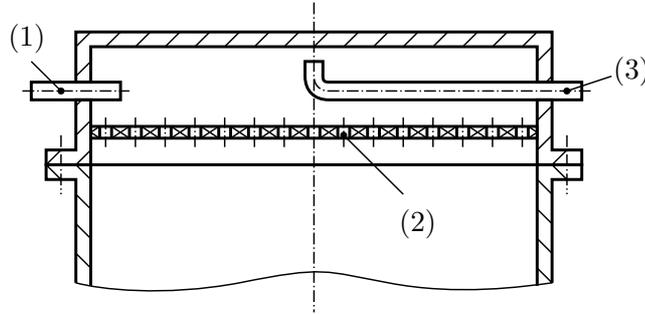


Abbildung 3.8: Einlassgeometrie [Zur91]

Für radiale Einströmung (1) konnte folgende Korrelation gefunden werden:

$$\epsilon_{in} = 0,344 \left(\frac{Re}{Ri} \right)^{0,894} \quad (3.31)$$

Für radiale Einströmung mit perforierter Platte (2):

$$\epsilon_{in} = 3,54 \left(\frac{Re}{Ri} \right)^{0,586} \quad (3.32)$$

Für die Aufpralleinströmung (3):

$$\epsilon_{in} = 4,75 \left(\frac{Re}{Ri} \right)^{0,522} \quad (3.33)$$

$$Re = \frac{w_{in} D_{in}}{\nu_m} \quad \text{und} \quad Ri = \frac{g H \beta_m \Delta\vartheta}{w_{in}^2} \quad (3.34)$$

Dabei ist w_{in} die Einströmgeschwindigkeit, D_{in} der Einlassdurchmesser, $\Delta\vartheta$ die Temperaturdifferenz zwischen einströmendem Wasser und der obersten bzw. untersten Schicht, ν_m der Mittelwert der kinematischen Viskositäten ν und β_m der Mittelwert der thermischen Ausdehnungskoeffizienten β , für die aufeinander treffenden Temperaturen.³

Abb. 3.9 stellt die Ergebnisse des Experiments gegenüber. Die turbulenten Verwirbelungen am Einlass steigen mit zunehmender Dynamik der Einströmung und abnehmenden Auftriebswirkungen. Das schlechteste Verhalten weist der radiale Einlass auf, das Verwenden der perfo-

³Hier sind Re und Ri wieder modifiziert. Bei Korrelationen ist immer auf die Definition der verwendeten dimensionslosen Kennzahlen zu achten. Beispielsweise verwendet Oppel in Glg. 3.30 eine andere Definition.

rierten Platte reduziert die Turbulenzen aber entscheidend. Die Aufprallströmung zeigt die beste Performances beim Abbau des eintretenden Impulses.

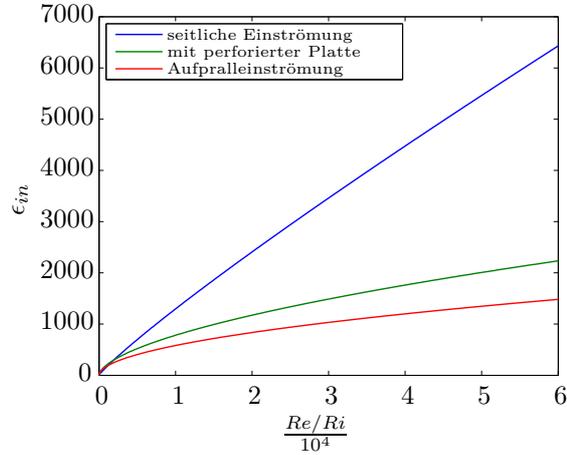


Abbildung 3.9: Verwirbelungsfaktor $\epsilon_{in} = \epsilon_{in}(Re/Ri, \text{Geometrie})$ [Zur91]

Eine weitere wichtige Beobachtung konnte in [Zur91] gemacht werden. Betrachtet man den Verlauf von ϵ_{in} über der Richardson-Zahl, bei festgehaltenem $\Delta\vartheta$ und variieren von w_{in} , alleine, so ist ab $Ri \geq 3,6$ kein signifikanter Unterschied aufgrund der Geometrie erkennbar.

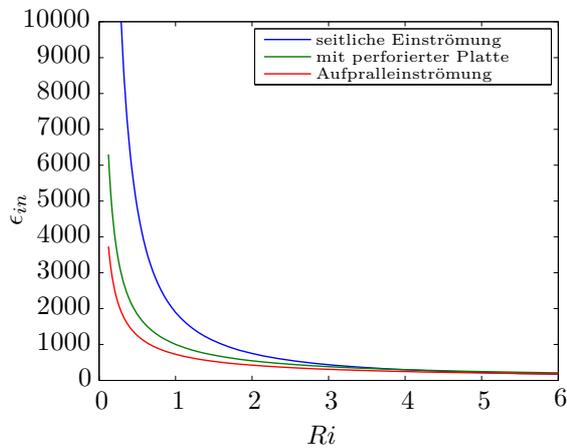


Abbildung 3.10: Verwirbelungsfaktor $\epsilon_{in} = \epsilon_{in}(Ri, \text{Geometrie})$ [Zur91]

Ergänzend sei hier angeführt, dass z.B. in [AN97] ein exponentieller Ansatz für $\epsilon = \epsilon(z)$ verwendet wird:

$$\epsilon = \epsilon_{in} e^{-\phi z} \quad (3.35)$$

Dabei ist ϕ der Abklingparameter. Das Optimum von a , b und ϕ ist jenes, welches die beste Korrelation zwischen zu erwartenden Ergebnissen und experimentellen Daten spezifischer Einlässe ergibt. Diese Modellparameter müssen von Fall zu Fall validiert werden. Es gibt keine

universellen, optimalen Werte für alle Geometrien.

Falsches Be- und Entladen

Darunter versteht man eine sehr unvorteilhafte Betriebsführung des Speichers. Das beim Beladevorgang einströmende Fluid am Speicherdeckel besitzt eine tiefere Temperatur, als die oberste heiße Schicht im Speicher. Falsches Entladen tritt auf, wenn das am Speicherboden eintretende Wasser wärmer, als die Schicht am Boden ist. Dann würden Inversionslagen auftreten, da bei der Modellierung des Speichers durch Glg. 3.36, das einströmende Fluid nur durch Wärmeleitung mit den Nachbarschichten thermische Energie tauschen kann. Es kommt zum Widerspruch mit den bisherigen Ausführungen zu langlebigen Inversionslagen.

Hier schafft wieder, das schon in Kapitel 3.2 erwähnte künstliche Beseitigen von Inversionslagen durch kalorisches Mischen, Abhilfe.

Die Schichtbeladung

Hat man allerdings keine Energiequelle mit konstanter Temperatur zur Verfügung, würden sich die oben beschriebenen unvorteilhaften Betriebsperioden mit konventionellen Speichern nicht vermeiden lassen. Bei solar-thermisch gestützten Heizsystemen ist das der Fall. Daher werden in solchen Fällen die Speicher mit einem Schichtladesystem ausgestattet, mit dem einströmendes Wasser, genau in die Schicht entsprechender Temperatur und damit entsprechender Dichte geleitet wird (vgl. Abb. 2.7).

Hierzu ist im Speicher ein Kunststoffrohr, eine sogenannte Schichtladelanze, größeren Durchmessers angeordnet, das eine geringe Strömungsgeschwindigkeit des aufsteigenden Wassers ermöglicht. Über die Höhe verteilt, sind an dem Rohr leicht bewegliche Rückschlagklappen aus Kunststoff angebracht, die sich nach außen öffnen lassen.

Ohne Durchfluss von Wasser in der Schichtladelanze, stellt sich im Rohr dieselbe Temperaturverteilung ein, wie im Speicher. Die Dichte im Rohr, entspricht damit jeweils derjenigen im Speicher und es wirken keine Kräfte auf die Klappen.

Bei Aufstieg wärmeren Wassers ergeben sich Temperaturunterschiede innerhalb und außerhalb des Rohrs. Das aufsteigende Wasser sei wärmer, als das in den unteren Schichten des Speichers. Dann bleiben die Klappen in diesen Bereichen aufgrund der außen größeren Dichte geschlossen. Nach weiterem Aufstieg erreicht das Wasser die Schicht, in der Temperaturgleichheit herrscht. Der dynamische Druck, des in der Schichtladelanze aufsteigenden Wassers reicht aus, um die Klappe zu öffnen. Das Wasser im Rohr tritt auf dem entsprechenden Temperaturniveau im Speicher aus. Das oberhalb der Klappe befindliche Wasser ist wärmer und hat daher eine geringere Dichte als das Wasser unterhalb. Bei dem größeren Querschnitt des Rohrs und der resultierenden geringeren Strömungsgeschwindigkeit, wird das aufsteigende Wasser

schnell abgebremst und kann daher nicht höher steigen. Dieses System funktioniert ohne Regelung, nur mit dem Naturumlaufprinzip, da die Klappen äußerst sensitiv gegenüber Druckunterschiede sind. [Fis05]

Für den Fall der Schichtbeladung wird ein eigenes mathematisches Modell erstellt. Die Ladelanze selbst und das darin befindliche Volumen wird dabei nicht abgebildet. Beim Durchströmen der Ladelanze findet somit auch kein Wärmeaustausch zwischen Wasser in der Lanze und den umgebenden Wasserschichten statt. Stattdessen wird lediglich die Einsortierung in eine gleich temperierte Schicht nachgebildet. Das dem Speicher zugeführte Volumen, verdrängt ein gleich großes Wasservolumen aus der untersten Schicht. Auf die gleiche Weise wird auch in [Bec06] die Schichtbeladung implementiert.

Ergänzungen

Nach Einbinden des Verwirbelungsfaktors ϵ ist man bei der fertigen Modellgleichung der Energiebilanz angekommen:

$$\rho c_p \left(\frac{\partial \vartheta}{\partial t} + w \frac{\partial \vartheta}{\partial z} \right) = \epsilon \lambda_{eff} \frac{\partial^2 \vartheta}{\partial z^2} + q \quad (3.36)$$

Diese soll nun für geeignete Anfangs- und Randbedingungen gelöst werden. Eine enorme Erleichterung ist, das sie aufgrund der demonstrierten Modellannahmen und Vereinfachungen nicht simultan mit der Impulsbilanz gelöst werden muss. w wird explizit als Randbedingung vorgegeben und ist lt. Kontinuitätsgleichung für alle z bei festgehaltener Zeit konstant.

All die im Verlauf dieses Kapitels verwendeten Vereinfachungen werden auch in der DIN EN 12977-3 (Thermische Solaranlagen und ihre Bauteile - Kundenspezifisch gefertigte Anlagen - Teil 3: Leistungsprüfung von Warmwasserspeichern für Solaranlagen) erlaubt. Die Anweisungen zur Modellbildung von Speichern werden wie folgt zusammengefasst: [Deu12]

- Für jede Komponente des Speichers kann angenommen werden, dass diese in horizontaler Richtung isotherm ist.
- Temperaturinversion innerhalb des Wasserraums, was bedeutet dass, $\frac{\partial \vartheta}{\partial z} < 0$ ist, können durch einen geeigneten Algorithmus am Ende jedes Zeitschritts beseitigt werden.
- die physikalischen Effekte der Wärmeleitung im Wasser und im Metall der Behälterwand, sowie die Konvektion im Wasser, können in eine effektive Wärmeleitfähigkeit zusammengefasst werden.

Bestehende Modelle von [Dru06], [Bec06], [Kle06], [Haf99], [New95] und [Kle90] verwenden auch solche eindimensionale Modellierungsansätze und sind teilweise validiert. Wie auch in [Deu12] handelt es sich hier jedoch immer um Kleinspeicher. Erst seit den Untersuchungen

von [Joh02] kann der Schluss gezogen werden, dass der eindimensionale Modellansatz von Kleinspeichern, auch zur Betrachtung von Großspeichern gerechtfertigt ist.

3.4 Analytische Lösung

Analytische Lösungen zu Glg. 3.36 sind nur für sehr einfache Anwendungsfälle zu finden. Der folgende Abschnitt soll solche Lösungen vorstellen und auch zeigen, wie schnell diese an ihre Grenzen stoßen, und somit die Notwendigkeit numerischer Lösungsverfahren verdeutlichen.

Der vollständig durchmischte Speicher

Das Modell habe die Gestalt eines perfekten nicht adiabaten Zylinders. Setzt man für den Quellterm in Glg. 3.36 die Wärmeverluste an die Umgebung mit der Wärmeverlustrate (kA) ein, so ergibt sich für die Speichertemperatur $\vartheta = \vartheta(z, t)$:

$$\rho c_p \left(\frac{\partial \vartheta}{\partial t} + w \frac{\partial \vartheta}{\partial z} \right) = \epsilon \lambda_{eff} \frac{\partial^2 \vartheta}{\partial z^2} - \frac{1}{V} (kA) (\vartheta - \vartheta_{amb}) \quad (3.37)$$

Der Speicher befinde sich im Stand-By Betrieb, d.h. $w = 0$ und seine Temperatur sei für alle z die gleiche, d.h. $\frac{\partial \vartheta}{\partial z} = 0$ und $\frac{\partial^2 \vartheta}{\partial z^2} = 0$. Die thermophysikalischen Eigenschaften von Fluid und Behälterwand seien konstant. Die Wärmeverlustrate ergibt sich aufgrund axialer Verluste durch Speicherboden und -deckel, und radialer Verluste durch den Speichermantel zu $(kA) = 2 A_{ax} k_{ax} + A_{rad} k_{rad}$. Aufgrund dieser Annahmen, wird $\vartheta(z, t)$ zu $\vartheta_m(t)$ und Glg. 3.37 vereinfacht sich zu folgender, gewöhnlichen, linearen Differentialgleichung, erster Ordnung:

$$\frac{\partial \vartheta_m}{\partial t} + \frac{2 A_{ax} k_{ax} + A_{rad} k_{rad}}{V \rho c_p} \vartheta_m = \frac{2 A_{ax} k_{ax} + A_{rad} k_{rad}}{V \rho c_p} \vartheta_{amb} \quad (3.38)$$

$$\vartheta_m(0) = \vartheta_{ini} \quad (3.39)$$

Mit der Anfangsbedingung aus Glg. 3.39 erhält man folgende Lösung:

$$\vartheta_m(t) = \vartheta_{amb} + (\vartheta_{ini} - \vartheta_{amb}) \exp \left(- \frac{2 A_{ax} k_{ax} + A_{rad} k_{rad}}{V \rho c_p} t \right) \quad (3.40)$$

Es handelt sich dabei um eine mit der Zeit t abklingende Exponentialfunktion, die für $t \rightarrow \infty$ nach der Umgebungstemperatur ϑ_{amb} strebt.

[Hau13] stellt eine recht anschauliche Größe, zur Beurteilung der Fähigkeit der Temperaturerhaltung vor. Nämlich die Halbwertszeit $t_{1/2}$, definiert als die Zeit, innerhalb der ein Speicher auf die Hälfte seiner ursprünglichen Temperaturdifferenz zur Umgebung abkühlt. Für Glg. 3.40 ergibt sie sich zu:

$$t_{1/2} = \frac{\ln(2)}{\frac{2 A_{ax} k_{ax} + A_{rad} k_{rad}}{V \rho c_p}} \quad (3.41)$$

Abb. 3.11 zeigt die Abkühlkurve des beschriebenen Modells für die Parameter: Speicherdurchmesser $D = 0,7$ m, Speicherhöhe $H = 1,2$ m, Umgebungstemperatur $\vartheta_{amb} = 20$ °C, Anfangstemperatur $\vartheta_{ini} = 90$ °C und Wärmedurchgangskoeffizienten $k_{rad} = k_{ax} = 0,12$ W/m²K. Die Halbwertszeit $t_{1/2}$ ergibt sich dabei zu 37,3 Tagen.

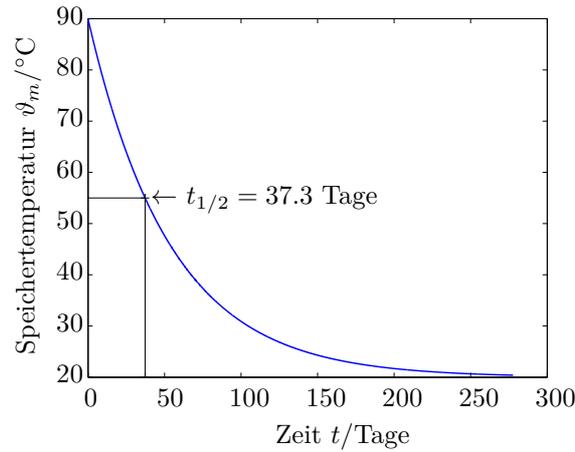


Abbildung 3.11: Abkühlkurve des vollständig durchmischten Speichers

Der ideal geschichtete Speicher

Zur Untersuchung des Beladevorgangs eines Schichtspeichers soll wieder ein idealer Zylinder als Modell dienen. Abb. 3.12 zeigt einen Speicher, der anfangs mit kaltem Wasser ϑ_{amb} gefüllt ist. Ab dem Zeitpunkt $t = 0$ s strömt heißes Fluid mit der Temperatur ϑ_{in} in den Behälter und der Beladevorgang beginnt.

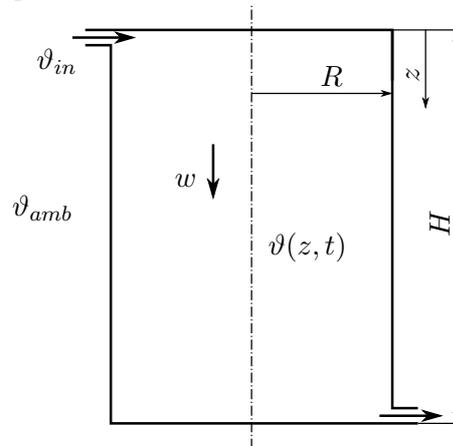


Abbildung 3.12: Modellierung des idealen Beladevorgangs in einem Schichtspeicher [Yoo96]

Zur Vereinfachung des mathematischen Modells wurden folgende Annahmen getroffen:

- Im gesamten Speicher sei eine Kolbenströmung mit der Geschwindigkeit w ausgeprägt.
- Das Einströmen an der Behälteroberseite geht ohne Mischen mit den umliegenden Regionen einher ($\epsilon(z) = 1$).
- Die thermophysikalischen Eigenschaften von Fluid und Behälterwand seien konstant.
- Die Behälterwände sind in axiale Richtung nicht leitend ($\lambda_{eff} = \lambda$).
- Massenstrom \dot{m} und Einströmtemperatur ϑ_{in} werden konstant gehalten.

Es werden insgesamt drei Lösungen vorgestellt, die sich hinsichtlich der Isolierung, also der Möglichkeit des Wärmetransports über die Behälterwände, unterscheiden. Betrachtet man Glg. 3.36 und setzt für den Quellterm die radialen Wärmeverluste über den Behältermantel ein, so erhält man für $\vartheta = \vartheta(z, t)$ folgende Gleichung:

$$\rho c_p \left(\frac{\partial \vartheta}{\partial t} + w \frac{\partial \vartheta}{\partial z} \right) = \lambda \frac{\partial^2 \vartheta}{\partial z^2} - \underbrace{\frac{2}{R} k_{rad} (\vartheta - \vartheta_{amb})}_{\text{radiale Verluste}} \quad (3.42)$$

Anfangsbedingung und Randbedingungen:

$$\vartheta(z, 0) = \vartheta_{amb} \quad (3.43)$$

$$\vartheta(0, t) = \vartheta_{in} \quad (3.44)$$

$$\vartheta(z, t) = \vartheta_{amb} \quad \text{für } z \rightarrow \infty \quad (3.45)$$

Als Anfangsbedingung sei thermisches Gleichgewicht des gesamten Speichers, mit seiner Umgebung gefordert. Die erste Randbedingung besagt, dass die Temperatur des Speicherdeckels, für alle Zeiten, gleich der Einströmtemperatur ϑ_{in} ist. Durch die zweite Randbedingung wird ein in z-Richtung unendlich ausgedehnter Speicher angenommen, welcher allerdings im Modell nur bis $z = H$ interpretiert wird. In dem Modell wird somit ein systematischer Wärmestrom durch die Systemgrenze - den Speicherboden - zugelassen. Diese Betrachtung hat gewisse Vorteile bei der Lösung der obigen Differentialgleichung. [Yoo96]

Durch Einführen und Einsetzen der dimensionslosen Größen

$$\xi = \frac{z}{H}, \quad \tau = Fo = \frac{t a}{H^2}, \quad Pe = \frac{w H}{a} \quad \text{und} \quad \Theta = \frac{\vartheta - \vartheta_{amb}}{\vartheta_{in} - \vartheta_{amb}} \quad (3.46)$$

in die Gleichungen 3.42 bis 3.45, wird $\vartheta(z, t)$ zu $\Theta(\xi, \tau)$ und es resultieren die nachstehenden Gleichungen:

$$\frac{\partial \Theta}{\partial \tau} + Pe \frac{\partial \Theta}{\partial \xi} = \frac{\partial^2 \Theta}{\partial \xi^2} - \underbrace{\frac{2H^2}{\lambda R} k_{rad}}_{\text{radiale Verluste}} \Theta \quad (3.47)$$

$$\Theta(\xi, 0) = 0 \quad (3.48)$$

$$\Theta(0, \tau) = 1 \quad (3.49)$$

$$\Theta(\xi, \tau) = 0 \quad \text{für } \xi \rightarrow \infty \quad (3.50)$$

[Yoo96] gibt die Lösung des beschriebenen Problems für adiabaten Behältermantel, also ohne Verlustterm in Glg. 3.47, an. Folgende Gleichung wurde durch Transformation in den Laplace-Raum, Lösen im Laplace-Raum und Rücktransformation gefunden:

$$\Theta(\xi, \tau) = \frac{1}{2} \left[\operatorname{erfc} \left(\frac{\xi - Pe\tau}{\sqrt{4\tau}} \right) + e^{Pe\xi} \operatorname{erfc} \left(\frac{\xi + Pe\tau}{\sqrt{4\tau}} \right) \right] \quad (3.51)$$

mit

$$\operatorname{erfc}(x) = 1 - \operatorname{erf}(x) = 1 - \frac{2}{\sqrt{\pi}} \int_0^x e^{-x^2} dx \quad (3.52)$$

[Joh02] zeigt den Lösungsweg des Problems, ohne Vernachlässigung der radialen Speicherverluste und erhält das Ergebnis:

$$\begin{aligned} \Theta(\xi, \tau) = & \frac{1}{2} \exp \left[\left(\frac{Pe}{2} - \frac{\sqrt{Pe^2 - \frac{8H^2}{\lambda R} k_{rad}}}{2} \right) \xi \right] \operatorname{erfc} \left(\frac{\xi - \tau \sqrt{Pe^2 - \frac{8H^2}{\lambda R} k_{rad}}}{\sqrt{4\tau}} \right) \\ & + \frac{1}{2} \exp \left[\left(\frac{Pe}{2} + \frac{\sqrt{Pe^2 - \frac{8H^2}{\lambda R} k_{rad}}}{2} \right) \xi \right] \operatorname{erfc} \left(\frac{\xi + \tau \sqrt{Pe^2 - \frac{8H^2}{\lambda R} k_{rad}}}{\sqrt{4\tau}} \right) \end{aligned} \quad (3.53)$$

Dabei kann man leicht erkennen, dass sich diese Lösung mit $k_{rad} = 0$ wieder auf Glg. 3.51 zurückführen lässt.

Für den dritten Fall werden die Speicherverluste wieder vernachlässigt und die Randbedingung am Speicherboden wird abgeändert:

$$\left. \frac{\partial \vartheta(z, t)}{\partial z} \right|_{z=H} = 0 \quad \text{bzw.} \quad \left. \frac{\partial \Theta(\xi, \tau)}{\partial \xi} \right|_{\xi=1} = 0 \quad (3.54)$$

Diese Neumannsche Randbedingung ist etwas anschaulicher und stellt einen ideal isolierten Speicherboden dar. Somit ist die gesamte Systemgrenze adiabater Natur, lediglich durch den Stoffstrom kann Energie über diese transportiert werden. In [Yoo96] wird der Lösungsweg

und schlussendlich das Ergebnis gezeigt:

$$\begin{aligned} \Theta(\xi, \tau) = & \frac{1}{2} \left[\operatorname{erfc} \left(\frac{\xi - Pe\tau}{\sqrt{4\tau}} \right) + e^{Pe\xi} \operatorname{erfc} \left(\frac{\xi + Pe\tau}{\sqrt{4\tau}} \right) \right] \\ & + e^{Pe} \left[\operatorname{erfc} \left(\frac{2 - \xi + Pe\tau}{\sqrt{4\tau}} \right) - Pe\sqrt{\tau} \operatorname{ierfc} \left(\frac{2 - \xi + Pe\tau}{\sqrt{4\tau}} \right) \right] \end{aligned} \quad (3.55)$$

mit

$$\operatorname{ierfc}(x) = \frac{e^{-x^2}}{\sqrt{\pi}} - x \operatorname{erfc}(x) \quad (3.56)$$

Direkt zu sehen ist, dass der erste Term von Glg. 3.55, ident mit Glg. 3.51 ist. Physikalisch kann der zweite Term als eine Korrektur, aufgrund der Endlichkeit des Lösungsbereiches interpretiert werden. [Yoo96]

Für die abgeänderte Randbedingung, unter Berücksichtigung von radialen Wärmeverlusten, konnte eine Lösung im Laplace-Raum gefunden werden, jedoch ist die Rücktransformation nicht gelungen. [Joh02]

Kapitel 4

Numerische Modelle

Dieses Kapitel befasst sich einleitend mit den gängigsten Diskretisierungsmethoden, welche kurz vorgestellt und hinsichtlich ihrer Eigenschaften verglichen werden.

Im weiteren Verlauf, wird die im vorhergehenden Kapitel gefundene Modellgleichung, mit Hilfe des Finite-Volumen-Verfahrens, diskretisiert. Das Ziel ist ein Modell zur Simulation von thermischen Schichtspeichern, mit direkter Beladung am Speicherboden und direkter Entladung am Speicherdeckel. Am Ende des Kapitels wird ein weiteres Modell, das eine direkte Schichtbeladung mittels Schichtladelanze zulässt, vorgestellt.

4.1 Diskretisierungsmethoden

Nach Auswahl des mathematischen Modells muss man eine geeignete Diskretisierungsmethode wählen, d.h. eine Methode zur Approximation der Differentialgleichungen, durch ein System algebraischer Gleichungen, für die Variablen an einem Satz von diskreten Stellen, in Raum und Zeit. Es gibt viele Herangehensweisen, von denen die wichtigsten sind: Finite-Differenzen-, Finite-Volumen- und Finite-Elemente-Methoden.

Andere Methoden, wie die Spektralmethoden und Rand-Elemente-Methoden, finden zwar in der CFD Anwendung, jedoch ist ihr Einsatz auf besondere Problemgruppen beschränkt.

Jede Methode ergibt die gleiche Lösung, wenn das Gitter sehr fein ist. Doch sind bestimmte Methoden für einige Problemgruppen mehr geeignet, als andere Methoden. Der Vorzug ist oft vom Geschmack des Entwicklers abhängig. [Fer08]

Finite-Differenzen-Methode

Ausgangspunkt ist die Erhaltungsgleichung in Differentialform. Das Lösungsgebiet wird von einem Gitter überdeckt. In jedem Gitterpunkt wird die Differentialgleichung genähert, indem die partiellen Ableitungen durch Approximationen, basierend auf Knotenwerten der Variable, ersetzt werden. Das Ergebnis ist eine algebraische Gleichung pro Gitterpunkt, in der die Va-

riablenwerte im selben und in einer bestimmten Anzahl von Nachbarpunkten als Unbekannte vorkommen.

Die Taylor-Reihenentwicklung oder ein Polynomansatz wird benutzt, um Approximationen der ersten und zweiten Ableitungen der Variablen, bezüglich der Koordinaten zu erhalten. Wenn nötig, können diese Methoden ebenfalls verwendet werden, um die Variablenwerte an Stellen zu berechnen, die keine Gitterpunkte sind (Interpolation).

Bei strukturierten Gittern ist die FD-Methode sehr einfach und effektiv. Insbesondere ist es einfach, Methoden höherer Ordnung, auf regelmäßigen Gittern zu erhalten. Der Nachteil der FD-Methoden besteht darin, dass sie, ohne besondere Maßnahmen, nicht konservativ⁴ sind. Außerdem ist die Beschränkung auf einfache Geometrien ein bedeutender Nachteil. [Fer08]

Finite-Volumen-Methode

Die FV-Methode verwendet die Integralform der Erhaltungsgleichung als Ausgangspunkt. Das Lösungsgebiet wird in eine finite Anzahl, nicht überlappender Kontrollvolumina unterteilt, und die Erhaltungsgleichung wird auf jedes Kontrollvolumen angewendet. Im Schwerpunkt eines jeden KV liegt der Rechenknoten, in dem die Variablenwerte berechnet werden. Interpolation wird verwendet, um die Variablenwerte auf der KV-Oberfläche mittels der Knotenwerte (KV-Zentren) auszudrücken. Oberflächen- und Volumenintegrale werden unter Verwendung passender Quadraturformeln approximiert. Als Ergebnis erhält man für jedes KV eine algebraische Gleichung, in der die Variablenwerte, aus dem eigenen Rechenknoten sowie aus einigen Nachbar-KVs vorkommen.

Die FV-Methode kann bei jedem Gittertyp angewendet werden, weshalb sie auch für komplexe Geometrien geeignet ist. Das Gitter bestimmt nur die Oberflächen der Kontrollvolumina und muss nicht auf ein Koordinatensystem bezogen werden. Die Methode ist per Definition konservativ, so lange die Oberflächenintegrale, die die konvektiven und diffusiven Flüsse darstellen, einheitlich für beide KVs, die eine gemeinsame Fläche haben, berechnet werden. Der Nachteil der FV-Methoden im Vergleich zu FD-Methoden besteht darin, dass Verfahren höherer Ordnung schwerer zu entwickeln sind. Dies ist auf den Umstand zurückzuführen, dass die FV-Methode drei Approximationsstufen verlangt: Interpolation, Differenziation und Integration. [Fer08]

Finite-Elemente-Methode

Die FE-Methode ist auf vielfältige Weise der FV-Methode ähnlich. Das Lösungsgebiet wird in einen Satz diskreter Volumen oder finiter Elemente, die generell unstrukturiert sind, aufgeteilt. Die unterscheidende Eigenschaft der FE-Methoden ist, dass die Gleichungen mit einer Ge-

⁴Da die zu lösenden Gleichungen Erhaltungsgesetze darstellen, sollte das numerische Verfahren ebenfalls, sowohl lokal als auch global - diese Gesetze erfüllen. Ist das der Fall, so nennt man das Verfahren konservativ. Dies ist eine wichtige Eigenschaft der Lösungsmethode, da sie den Lösungsfehler beschränkt. [Fer08]

wichtungsfunktion multipliziert werden, bevor sie über das gesamte Lösungsgebiet integriert werden. In den einfachsten FE-Methoden wird die Lösung mit einer linearen Formfunktion innerhalb eines jeden Elementes so approximiert, dass Kontinuität der Lösung, entlang der Elementeränder, garantiert ist. Solch eine Funktion kann aus Variablenwerten in den Ecken der Elemente konstruiert werden. Die Gewichtungsfunktion hat normalerweise die gleiche Form.

Diese Approximation wird dann in das gewichtete Integral der Erhaltungsgleichung substituiert, und die zu lösenden Gleichungen werden hergeleitet, indem man verlangt, dass die Ableitung des Integrals, in Bezug auf jeden Knotenwert, gleich null ist. Dies entspricht der Auswahl, der besten Lösung innerhalb eines Satzes von erlaubten Funktionen (die mit dem minimalen Residuum). Das Ergebnis ist ein Satz nichtlinearer algebraischer Gleichungen.

Ein wichtiger Vorteil der FE-Methode ist die Möglichkeit, beliebige Geometrien behandeln zu können. FE-Methoden sind relativ leicht mathematisch zu analysieren, und haben nachweislich optimale Eigenschaften für bestimmte Gleichungstypen. Der prinzipielle Nachteil, aller Methoden, die unstrukturierte Gitter verwenden, gemeinsam ist, besteht darin, dass die Matrizen der linearisierten Gleichungen nicht so gut strukturiert sind wie die für regelmäßige Gitter, was es schwieriger macht, eine effiziente Lösungsmethode zu finden.

Eine Hybridmethode, die auf Kontrollvolumina basierende Finite-Elemente-Methode genannt wird, sollte ebenfalls erwähnt werden. Bei ihr werden Formfunktionen zur Beschreibung des Verlaufs der Variablen über ein Element verwendet. Kontrollvolumen werden um jeden Gitterknoten durch Verbindung der Mittelpunkte der Elemente, der Seitenflächen und der Kanten gebildet. Die Erhaltungsgleichungen in Integralform werden auf diese KVs auf die gleiche Weise, wie bei der klassischen Finite-Volumen-Methode, angewendet. Die Flüsse durch die KV-Ränder und die Quellterme werden elementeweise berechnet. [Fer08]

4.2 Die Modellgleichung

Mit Hilfe der Finite-Volumen-Methode soll nun Schritt für Schritt die gefundene Differentialgleichung diskretisiert und approximiert werden.

Räumliche Diskretisierung

Das zu betrachtende Rechengebiet, der gesamte Speicherinhalt ohne Wände, wird in eine endliche Anzahl n äquidistanter ($\Delta z = \text{const.}$) scheibenförmiger Teilvolumina unterteilt. Abb. 4.1 zeigt den allgemeinen Fall der inneren Kontrollvolumina i , sowie die beiden Randvolumina 1 und n . Im Zentrum dieser Volumina befindet sich ein Knoten mit der diskretisierten Temperatur ϑ_i .

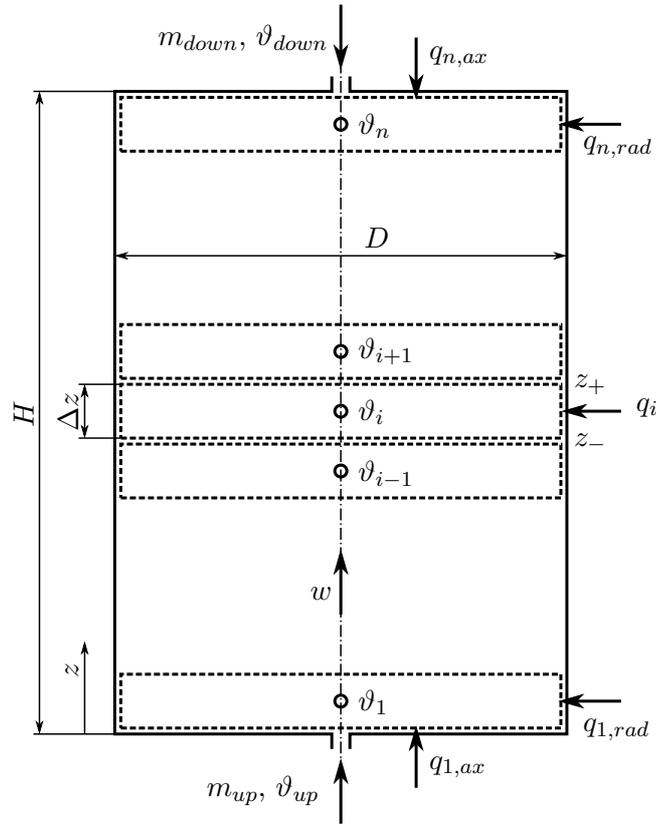


Abbildung 4.1: Zerlegung des Rechengebiets

Ausgangspunkt ist Glg. 3.36, die im allgemeinen Fall, über das Kontrollvolumen i integriert wird:

$$\int_{z_-}^{z_+} \rho c_p \left(\frac{\partial \vartheta}{\partial t} + w \frac{\partial \vartheta}{\partial z} \right) A_{ax} dz = \int_{z_-}^{z_+} \left(\epsilon \lambda_{eff} \frac{\partial^2 \vartheta}{\partial z^2} + q \right) A_{ax} dz \quad (4.1)$$

Hierbei handelt es sich um eine Energieerhaltungsgleichung, in der über c_p der Zusammenhang, zwischen der Temperatur ϑ und der Enthalpie h , hergestellt wird. Für eine präzisere Formulierung muss allerdings die integrale isobare spezifische Wärmekapazität \bar{c}_p eingeführt werden:

$$h(\vartheta_1) = \bar{c}_p(\vartheta_1) \vartheta_1 = \frac{1}{\vartheta_1} \left[h(\vartheta_{ref}) + \int_{\vartheta_{ref}}^{\vartheta_1} c_p(\vartheta) d\vartheta \right] \vartheta_1 \quad (4.2)$$

Die Referenztemperatur ϑ_{ref} des Wassers ist definitionsgemäß die Temperatur am Tripelpunkt mit $0,01 \text{ }^\circ\text{C}$.

Die Strömungsgeschwindigkeit w ist durch die Randbedingung eindeutig festgelegt, und für den betrachteten Zeitschritt über das gesamte Rechengebiet konstant. Mit den positiv definierten Massenströmen \dot{m}_{up} und \dot{m}_{down} , und der Voraussetzung, dass einer der beiden gleich Null gesetzt sein muss, ergibt sich folgende Gleichung:

$$w(z) = \frac{\dot{m}_{up} - \dot{m}_{down}}{A_{ax} \rho} = \text{const.} \quad (4.3)$$

$$\int_{z_-}^{z_+} \rho \bar{c}_p \frac{\partial \vartheta}{\partial t} dz = \int_{z_-}^{z_+} \left(-\frac{\dot{m}_{up} - \dot{m}_{down}}{A_{ax}} \bar{c}_p \frac{\partial \vartheta}{\partial z} + \epsilon \lambda_{eff} \frac{\partial^2 \vartheta}{\partial z^2} + q \right) dz \quad (4.4)$$

Integration über z mit $\lambda_{eff} = \text{const.}$ und $q = \text{const.}$ liefert:

$$\begin{aligned} \left(\rho \bar{c}_p \frac{\partial \vartheta}{\partial t} \right)_i \Delta z &= -\frac{\dot{m}_{up} - \dot{m}_{down}}{A_{ax}} [(\bar{c}_p \vartheta)_{z_+} - (\bar{c}_p \vartheta)_{z_-}] \\ &+ \lambda_{eff} \left[\left(\epsilon \frac{\partial \vartheta}{\partial z} \right)_{z_+} - \left(\epsilon \frac{\partial \vartheta}{\partial z} \right)_{z_-} \right] + q_i \Delta z \end{aligned} \quad (4.5)$$

Für alle, in der Gleichung, auftretenden Wärmekapazitäten wird $\bar{c}_p = \bar{c}_p(\vartheta_i)$ eingesetzt, da die Ermittlung der temperaturabhängigen Stoffwerte sehr rechenintensiv ist. Die Wärmekapazitäten ändern sich mit der Temperatur nur sehr langsam, d.h. der dadurch hervorgerufene Fehler ist minimal.

Lt. [Pat80] können die Temperaturen an den KV-Grenzen durch lineare Interpolation genähert werden:

$$\vartheta_{z_-} = \frac{\vartheta_i + \vartheta_{i-1}}{2} \quad \text{und} \quad \vartheta_{z_+} = \frac{\vartheta_{i+1} + \vartheta_i}{2} \quad (4.6)$$

Dann ergibt sich nach Umformen von Glg 4.5 folgender Ausdruck:

$$\frac{\vartheta_{i+1} - \vartheta_{i-1}}{2 \Delta z} \quad (4.7)$$

Das entspricht einer Approximation der ersten Ableitung des konvektiven Terms in Glg. 4.4 durch zentrale Differenzen, mit einem Abbruchfehler von der Größenordnung $O(\Delta z^2)$. Es können allerdings Oszillationen auftreten, das Verfahren wird instabil⁵. [Kuh10]

Um diese Instabilitäten zu vermeiden, müssen die Gleichungen modifiziert werden: Betrachtet man Glg. 4.5, so sind ϑ_{z_-} und ϑ_{z_+} die Temperaturen mit der die Massenströme, die KV-Grenzen passieren. Man verzichtet auf die Interpolation und setzt, unter Berücksichtigung der Fließrichtung, direkt die Knotentemperaturen ein:

für $w > 0$ bzw. $m_{up} \neq 0$:

$$\vartheta_{z_+} - \vartheta_{z_-} = \vartheta_i - \vartheta_{i-1} \quad (4.8)$$

⁵Bei numerischen Berechnungen treten immer spontane Störungen auf, die in der Regel durch Rundungsfehler bedingt sind. Diese spontanen Störungen verhindern eine exakte Lösung der mit dem Diskretisierungsfehler behafteten algebraischen Gleichung. Wenn die spontanen Fehler im Laufe der Rechnung, z.B. bei der zeitlichen Integration, kumulativ signifikant anwachsen, dann ist das numerische Verfahren instabil. Falls der kumulative Fehler aber für alle Zeiten vernachlässigbar klein bleibt, ist das Verfahren stabil. [Kuh10]

für $w < 0$ bzw. $m_{down} \neq 0$:

$$\vartheta_{z_+} - \vartheta_{z_-} = \vartheta_{i+1} - \vartheta_i \quad (4.9)$$

Das entspricht einer Approximation der ersten Ableitung durch Rückwärts- bzw. Vorwärtsdifferenzen. Der Abbruchfehler nimmt gegenüber den zentralen Differenzen zu und liegt bei $O(\Delta z)$. Das wirkt sich negativ auf das Phänomen der numerische Diffusion⁶ aus, Oszillationen werden allerdings unterdrückt, und das Verfahren wird dadurch stabil. [Kuh10]

Mit diesen Erkenntnissen, den zentrale Differenzen

$$\left. \frac{\partial \vartheta}{\partial z} \right|_{z_-} = \frac{\vartheta_i - \vartheta_{i-1}}{\Delta z}, \quad \left. \frac{\partial \vartheta}{\partial z} \right|_{z_+} = \frac{\vartheta_{i+1} - \vartheta_i}{\Delta z} \quad (4.10)$$

dem Knotenvolumen $V_{node} = A_{ax} \Delta z$ und den radialen Verlusten $q_i = -\frac{A_{rad}}{V_{node}} k_{rad} (\vartheta_i - \vartheta_{amb})$, ergibt sich für $i = 2 \dots (n-1)$ folgende Gleichung:

$$\begin{aligned} \rho \bar{c}_p \left. \frac{\partial \vartheta}{\partial t} \right|_i &= -\frac{\dot{m}_{up} \bar{c}_p}{V_{node}} (\vartheta_i - \vartheta_{i-1}) + \frac{\dot{m}_{down} \bar{c}_p}{V_{node}} (\vartheta_{i+1} - \vartheta_i) \\ &+ \frac{\lambda_{eff}}{\Delta z^2} [\epsilon_{z_+} (\vartheta_{i+1} - \vartheta_i) - \epsilon_{z_-} (\vartheta_i - \vartheta_{i-1})] - \frac{A_{rad} k_{rad}}{V_{node}} (\vartheta_i - \vartheta_{amb}) \end{aligned} \quad (4.11)$$

Umsortieren liefert:

$$\begin{aligned} \rho \bar{c}_p \left. \frac{\partial \vartheta}{\partial t} \right|_i &= \left[-\frac{\dot{m}_{up} \bar{c}_p}{V_{node}} - \frac{\dot{m}_{down} \bar{c}_p}{V_{node}} - \frac{\lambda_{eff}}{\Delta z^2} (\epsilon_{z_+} + \epsilon_{z_-}) - \frac{A_{rad} k_{rad}}{V_{node}} \right] \vartheta_i \\ &+ \left[\frac{\dot{m}_{down} \bar{c}_p}{V_{node}} + \frac{\lambda_{eff}}{\Delta z^2} \epsilon_{z_+} \right] \vartheta_{i+1} \\ &+ \left[\frac{\dot{m}_{up} \bar{c}_p}{V_{node}} + \frac{\lambda_{eff}}{\Delta z^2} \epsilon_{z_-} \right] \vartheta_{i-1} \\ &+ \left[\frac{A_{rad} k_{rad}}{V_{node}} \vartheta_{amb} \right] \end{aligned} \quad (4.12)$$

Für $i = 1$, dem Randvolumen am Speicherboden, setzt man im Falle der Strömung nach oben $\vartheta_{i-1} = \vartheta_{up}$. Der diffusive Term, der den Energieaustausch durch Wärmeleitung mit dem darunter liegenden Knoten ermöglicht, fällt sinngemäß raus und ein axialer Verlustterm durch die Behälterwand wird hinzugefügt:

$$\begin{aligned} \rho \bar{c}_p \left. \frac{\partial \vartheta}{\partial t} \right|_1 &= -\frac{\dot{m}_{up} \bar{c}_p}{V_{node}} (\vartheta_1 - \vartheta_{up}) + \frac{\dot{m}_{down} \bar{c}_p}{V_{node}} (\vartheta_2 - \vartheta_1) \\ &+ \frac{\lambda_{eff}}{\Delta z^2} \epsilon_{z_+} (\vartheta_2 - \vartheta_1) - \frac{A_{rad} k_{rad} + A_{ax} k_{ax}}{V_{node}} (\vartheta_1 - \vartheta_{amb}) \end{aligned} \quad (4.13)$$

⁶Numerische Diffusion, auch künstliche oder falsche Diffusion genannt, tritt immer dann auf, wenn der führende Abbruchfehlerterm der Approximation diffusiven Charakter aufweist. Das äußert sich in der Lösung der Bilanzgleichung durch eine scheinbar stärker vorhandene Diffusion. [Fer08]

$$\begin{aligned}
\rho \bar{c}_p \frac{\partial \vartheta}{\partial t} \Big|_1 &= \left[-\frac{\dot{m}_{up} \bar{c}_p}{V_{node}} - \frac{\dot{m}_{down} \bar{c}_p}{V_{node}} - \frac{\lambda_{eff}}{\Delta z^2} \epsilon_{z+} - \frac{A_{rad} k_{rad} + A_{ax} k_{ax}}{V_{node}} \right] \vartheta_1 \\
&+ \left[\frac{\dot{m}_{down} \bar{c}_p}{V_{node}} + \frac{\lambda_{eff}}{\Delta z^2} \epsilon_{z+} \right] \vartheta_2 \\
&+ \left[\frac{\dot{m}_{up} \bar{c}_p}{V_{node}} \vartheta_{up} + \frac{A_{rad} k_{rad} + A_{ax} k_{ax}}{V_{node}} \vartheta_{amb} \right]
\end{aligned} \tag{4.14}$$

Analog für $i = n$ dem KV am Speicherdeckel:

$$\begin{aligned}
\rho \bar{c}_p \frac{\partial \vartheta}{\partial t} \Big|_n &= -\frac{\dot{m}_{up} \bar{c}_p}{V_{node}} (\vartheta_n - \vartheta_{n-1}) + \frac{\dot{m}_{down} \bar{c}_p}{V_{node}} (\vartheta_{down} - \vartheta_n) \\
&- \frac{\lambda_{eff}}{\Delta z^2} \epsilon_{z-} (\vartheta_n - \vartheta_{n-1}) - \frac{A_{rad} k_{rad} + A_{ax} k_{ax}}{V_{node}} (\vartheta_n - \vartheta_{amb})
\end{aligned} \tag{4.15}$$

$$\begin{aligned}
\rho \bar{c}_p \frac{\partial \vartheta}{\partial t} \Big|_n &= \left[-\frac{\dot{m}_{up} \bar{c}_p}{V_{node}} - \frac{\dot{m}_{down} \bar{c}_p}{V_{node}} - \frac{\lambda_{eff}}{\Delta z^2} \epsilon_{z-} - \frac{A_{rad} k_{rad} + A_{ax} k_{ax}}{V_{node}} \right] \vartheta_n \\
&+ \left[\frac{\dot{m}_{up} \bar{c}_p}{V_{node}} + \frac{\lambda_{eff}}{\Delta z^2} \epsilon_{z-} \right] \vartheta_{n-1} \\
&+ \left[\frac{\dot{m}_{down} \bar{c}_p}{V_{node}} \vartheta_{down} + \frac{A_{rad} k_{rad} + A_{ax} k_{ax}}{V_{node}} \vartheta_{amb} \right]
\end{aligned} \tag{4.16}$$

Fasst man die Koeffizienten zusammen, so wird deutlich, dass die oben angeführten Gleichungen von gleicher Gestalt sind:

$$a_1 \frac{\partial \vartheta}{\partial t} \Big|_i = a_2 \vartheta_i + a_3 \vartheta_{i+1} + a_4 \vartheta_{i-1} + a_5 \tag{4.17}$$

Zeitintegration

Die Zeitachse wird in Intervalle mit der Weite Δt zerlegt, die nicht notwendigerweise von der gleichen Dauer sind. Nun kann Glg. 4.17 über einen allgemeinen Zeitschritt Δt integriert werden:

$$\int_t^{t+\Delta t} a_1 \frac{\partial \vartheta}{\partial t} \Big|_i dt = \int_t^{t+\Delta t} (a_2 \vartheta_i + a_3 \vartheta_{i+1} + a_4 \vartheta_{i-1} + a_5) dt \tag{4.18}$$

An diesem Punkt braucht man eine Annahme, über den Temperaturverlauf, über den Zeitschritt. [Pat80] Abhilfe schafft der eingeführte Gewichtungsfaktor f , mit dem sich nach der Integration folgende Glg. ergibt:

$$(a_1 \vartheta_i)^1 - (a_1 \vartheta_i)^0 = \left[(a_2 \vartheta_i + a_3 \vartheta_{i+1} + a_4 \vartheta_{i-1} + a_5)^1 f + (a_2 \vartheta_i + a_3 \vartheta_{i+1} + a_4 \vartheta_{i-1} + a_5)^0 (1 - f) \right] \Delta t \quad (4.19)$$

Ein hochgestellter Index 0 zeigt an, dass es sich dabei um Werte zu Beginn des Zeitschritts handelt, ein hochgestellter Index 1 entspricht dem Ende des Zeitschritts. Der Gewichtungsfaktor f kann Werte zwischen 0 und 1 annehmen. Abb. 4.2 zeigt verschiedene mögliche Temperaturverläufe über den Zeitschritt.

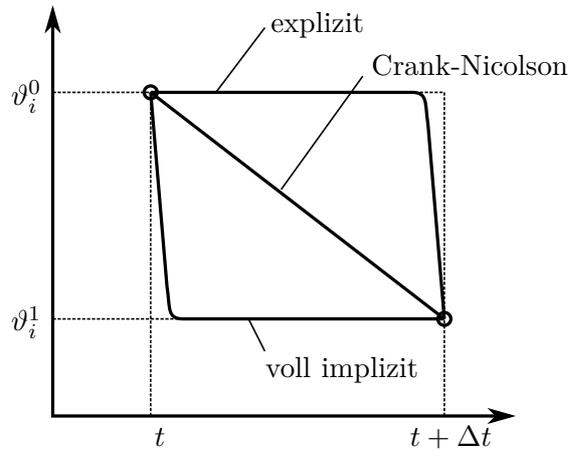


Abbildung 4.2: Temperaturverlauf während des Zeitschritts

Mit $f = 0$ erhält man das explizite-Euler-Verfahren. Die Knotentemperatur entspricht, während des gesamten Zeitschritts, der Starttemperatur und springt am Ende plötzlich auf den neuen Wert. Dieses Verfahren ist mit dem geringsten Rechenaufwand verbunden, da die folgenden Gleichungen für alle Knoten i direkt lösbar sind.

$$a_1 (\vartheta_i^1 - \vartheta_i^0) = (a_2 \vartheta_i^0 + a_3 \vartheta_{i+1}^0 + a_4 \vartheta_{i-1}^0 + a_5) \Delta t \quad (4.20)$$

$$\vartheta_i^1 = (a_2 \vartheta_i^0 + a_3 \vartheta_{i+1}^0 + a_4 \vartheta_{i-1}^0 + a_5) \frac{\Delta t}{a_1} + \vartheta_i^0 \quad (4.21)$$

Setzt man $f = 1$, so resultiert das voll-implizite-Schema. Die Knotentemperatur springt bereits zu Beginn des Zeitschritts auf den neuen Wert und behält diesen bis zum Endzeitpunkt bei. Umformen liefert Glg. 4.23, die für $i = 1 \dots n$ einem tridiagonalen Gleichungssystem entspricht, das mit beträchtlichem Mehraufwand an Rechenzeit verbunden ist. Es gibt verschiedene Lösungsmethoden, wie z.B. den Tridiagonalmatrix-Algorithmus (TDMA) oder auch Thomas-Algorithmus genannt.

$$a_1 (\vartheta_i^1 - \vartheta_i^0) = \left(a_2 \vartheta_i^1 + a_3 \vartheta_{i+1}^1 + a_4 \vartheta_{i-1}^1 + a_5 \right) \Delta t \quad (4.22)$$

$$\left(\frac{a_1}{\Delta t} - a_2 \right) \vartheta_i^1 = a_2 \vartheta_i^1 + a_3 \vartheta_{i+1}^1 + a_4 \vartheta_{i-1}^1 + a_5 + \frac{a_1}{\Delta t} \vartheta_i^0 \quad (4.23)$$

Mit $f = 0,5$ erhält man das sogenannte Crank-Nicolson-Verfahren, das einer Mischung aus explizitem und implizitem Verfahren entspricht. Die Knotentemperatur steigt linear über den Zeitschritt vom alten, zum neuen Wert. Man erhält wieder ein tridiagonales Gleichungssystem, bei dem der Rechenaufwand geringfügig größer, als der beim implizitem Verfahren ist.

$$a_1 (\vartheta_i^1 - \vartheta_i^0) = \left[\frac{a_2}{2} (\vartheta_i^1 + \vartheta_i^0) + \frac{a_3}{2} (\vartheta_{i+1}^1 + \vartheta_{i+1}^0) + \frac{a_4}{2} (\vartheta_{i-1}^1 + \vartheta_{i-1}^0) + a_5 \right] \Delta t \quad (4.24)$$

$$\left(\frac{a_1}{\Delta t} - \frac{a_2}{2} \right) \vartheta_i^1 = \frac{a_3}{2} \vartheta_{i+1}^1 + \frac{a_4}{2} \vartheta_{i-1}^1 + a_5 + \left(\frac{a_1}{\Delta t} + \frac{a_2}{2} \right) \vartheta_i^0 + \frac{a_3}{2} \vartheta_{i+1}^0 + \frac{a_4}{2} \vartheta_{i-1}^0 \quad (4.25)$$

Bei allen drei Verfahren wurde $a_1 = a_1^0 \dots a_5 = a_5^0$ angenommen. Eigentlich könnten diese Koeffizienten ebenfalls im Laufe des Zeitschritts variieren, da sie z.B. die Dichte ρ beinhalten, die wiederum von der Temperatur abhängt. Ohne dieser Vereinfachung würde das Gleichungssystem nur iterativ lösbar sein, da man zur Ermittlung der Dichte, am Ende des Zeitschritts, die gesuchte Temperatur selbst benötigt und der Rechenaufwand würde entscheidend steigen. Unter der Voraussetzung, dass Δt ausreichend klein gewählt wird, sodass sich die Temperaturen und damit auch die Stoffwerte nur geringfügig ändern, ist diese Vereinfachung vertretbar.

Welches dieser drei Verfahren das geeignetere ist, kann nicht generell gesagt werden. Jedes zeichnet sich für verschiedene Anforderungen aus, die Wahl ist also sehr stark von der Problemstellung selbst abhängig.

Abbau inverser Schichtung

Wie oben schon erwähnt, könnten nach jedem Zeitschritt inverse Schichtungen durch falsches Be- bzw. Entladen und durch die Wärmeverluste am Speicherdeckel auftauchen. Eine geeignete Methode, bei der die betroffenen Regionen durch die nur iterativ zu lösenden Glg. 3.16 kalorisch gemischt werden, wurde vorgestellt.

Die folgende Prozedur entspricht prinzipiell dem Lösen dieser Gleichung: Die Knotentemperaturen werden immer paarweise, von unten nach oben, miteinander verglichen. Sollte der höher gelegene Knoten kälter sein, so werden die beiden Zellen gemischt. Das wird so lange gemacht, bis keine Inversionslagen mehr vorhanden sind.

Für das äquidistante Gitter ergibt sich die Mischungsvorschrift:

$$\rho(\vartheta_i) \bar{c}_p(\vartheta_i) \vartheta_i + \rho(\vartheta_{i+1}) \bar{c}_p(\vartheta_{i+1}) \vartheta_{i+1} = 2 \rho(\vartheta_m) \bar{c}_p(\vartheta_m) \vartheta_m \quad (4.26)$$

Diese Glg. ist auch nur iterativ zu lösen, da die Stoffwerte auf der rechten Seite von der gesuchten Mischtemperatur ϑ_m selbst abhängig sind. Die Temperaturabhängigkeit der Stoffwerte ist allerdings sehr gering und kann in dem Fall vernachlässigt werden:

$$\vartheta_i + \vartheta_{i+1} = 2 \vartheta_m \quad (4.27)$$

Eine Vergleichsrechnung hat gezeigt, dass die dadurch verursachte Abweichung des Ergebnisses im 10^{-4} K Bereich liegen. Der Rechenaufwand reduziert sich dadurch dramatisch, da das Ermitteln der Stoffwerte und iterative Lösungsverfahren recht rechenintensiv sind.

Diese Prozedur muss nach jedem Zeitschritt durchgeführt werden, und nimmt mit steigender Knotenanzahl n immer mehr Rechenzeit in Anspruch. Um diese etwas zu schmälern werden Temperaturgradienten von -10^{-4} K/Knoten, ohne zu mischen, zugelassen. Das entspricht einem nicht fühlbaren Temperaturunterschied, und wird in [New95] auf ähnliche Weise gehandhabt.

Numerische Diffusion und Stabilität

In diesem Abschnitt soll gezeigt werden, welche der drei vorgestellten Zeitintegrationsmethoden, die geeignetere, für die Ansprüche an das Modell ist. Die Stabilität des Verfahrens, die Genauigkeit und der Rechenaufwand werden die entscheidenden Faktoren darstellen.

Im Fall von rein konvektivem Transport kann für das explizite Verfahren, mit Hilfe der Courant-Zahl, eine klare Stabilitätsgrenze angegeben werden:

$$C = \frac{w \Delta t}{\Delta z} = 1 \quad (4.28)$$

Sie gibt an, um wie viele Zellen sich eine betrachtete Größe, pro Zeitschritt, maximal fortbewegt. Tritt in der Transportgleichung auch ein diffusiver Term auf, dann wird die Stabilitätsgrenze unter $C = 1$ liegen. Der kritische Zeitschritt für das rein konvektive Problem ergibt sich dann zu:

$$\Delta t_{crit} = \frac{\Delta z}{w} \quad (4.29)$$

Abb. 4.3 vergleicht die drei Integrationsmethoden, hinsichtlich ihrer Auswirkung, auf die numerische Diffusion. Um diese sichtbar zu machen, wird ein rein konvektiver Beladevorgang, d.h. $\lambda_{eff} = 0$ ohne Verluste $k_{rad} = k_{ax} = 0$, bis zum Austausch des halben Speichervolumen betrachtet. Die Knotenanzahl wird für alle drei Verfahren mit $n = 10^3$ gewählt und die Zeitschrittweite auf $\Delta t = \Delta t_{crit}$ gesetzt.

Das explizite Verfahren an der Stabilitätsgrenze zeichnet sich hinsichtlich numerischer Diffusion aus. Diese Tatsache wird für das Modell in [Opp86] ausgenutzt.

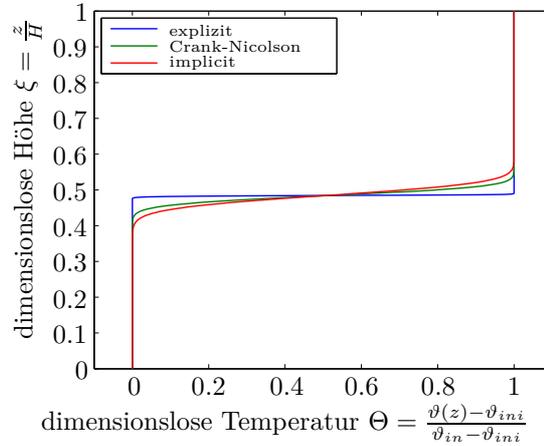


Abbildung 4.3: Einfluss der Integrationsmethode auf die numerische Diffusion

Die Abhängigkeit der numerischen Diffusion von der Knotenanzahl, soll durch Abb. 4.4 verdeutlicht werden. Dabei handelt es sich um den gleichen Beladevorgang wie oben. Mit steigender Knotenanzahl klingt der Diskretisierungsfehler, und damit auch die daraus resultierende numerische Diffusion, ab. Das führt aber auch zu einer Zunahme der Gleichungen des algebraischen Systems und der Rechenaufwand steigt. Einmal mehr zeigt sich, dass die Knotenanzahl eine große Rolle bei der Validierung des Modells spielt.

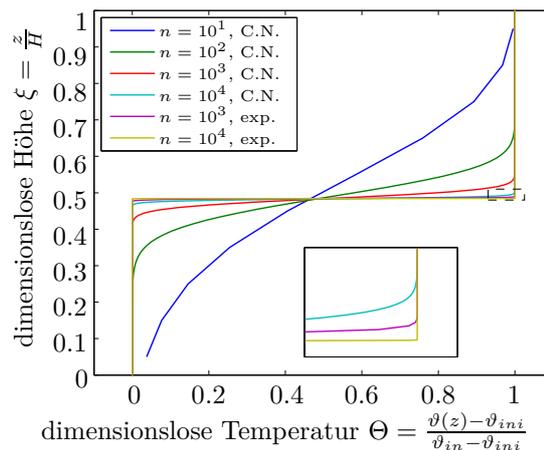


Abbildung 4.4: Einfluss der Knotenanzahl n auf die numerische Diffusion

Bisher wurde immer der rein konvektive Transport mit $\Delta t = \Delta t_{crit}$ untersucht. Es hat sich gezeigt, dass sich für diesen Fall, das explizite Verfahren auszeichnet.

In der Praxis sollen aber Speicherzyklen über die Dauer von mehreren Monaten, mit vertretbarem Rechenaufwand, simuliert werden können. Das erfordert ein Verfahren, das für große Zeitschrittweiten Δt hinreichend genau rechnet und vor allem stabil bleibt.

$$\Delta t = f_{crit} \Delta t_{crit} \quad (4.30)$$

Es wird nun ein allgemeiner Beladevorgang mit den realistischen Speicherparametern $k_{rad} = 1 \text{ W/m}^2\text{K}$ und $\lambda_{eff} = 1,6 \text{ W/mK}$ betrachtet. Die Knotenanzahl soll dabei $n = 100$ betragen. Für diese Untersuchung wird auf das kalorische Mischen nach jedem Zeitschritt verzichtet, da sollten Instabilitäten auftauchen, deren Erscheinungen ansonsten künstlich beseitigt werden. Es muss also darauf geachtet werden, dass keine Inversionslagen entstehen können ($k_{ax} = 0$, $\vartheta_{in} > \vartheta_{ini}$).

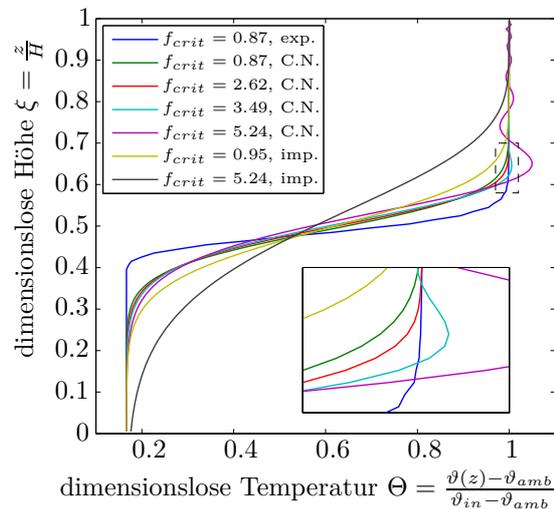


Abbildung 4.5: Gegenüberstellung der Integrationsmethoden

Die Ergebnisse sind in Abb. 4.5 für die drei Integrationsmethoden und verschiedene Zeitschrittweiten Δt zusammengefasst. Das explizite Verfahren befindet sich mit $f_{crit} = 0,87$ an der Stabilitätsgrenze, weist aber das beste Verhalten hinsichtlich numerischer Diffusion auf. Mit zunehmender Zeitschrittweite nimmt für das voll-implizite, Verfahren die numerische Diffusion beträchtlich zu, es scheint aber unbedingt stabil zu sein.

Beim Crank-Nicolson Verfahren hält sich die numerische Diffusion in Grenzen. Es sind nur sehr kleine Veränderungen mit zunehmender Zeitschrittweite zu erkennen. Jedoch neigt das Verfahren ab $f_{crit} = 3,49$ zu leichtem Überschwingen.

All diese Beobachtungen decken sich mit den Ausführungen in [Pat80] und [Fer08]. Das Crank-Nicolson Verfahren scheint also ein guter Kompromiss hinsichtlich Genauigkeit, Stabilität und möglicher Zeitschrittweite zu sein.

Zeitschritthandling

Bisher wurden immer nur Beladevorgänge betrachtet, bei denen die Zeitschrittweite sehr eingeschränkt ist. Im Stand-By Betrieb, der für Langzeitspeicherung klarerweise den Großteil der Simulationszeit ausmacht, sind keine Massenströme vorhanden und viel größere Zeitschrittweiten möglich. Es wäre daher sinnvoll, zwei unterschiedliche Zeitschrittweiten bei der Simulation zu verwenden:

Der Benutzer gibt den prinzipiellen Zeitschritt Δt zur Diskretisierung der Zeitachse vor. Dieser wird für die Berechnungen im Stand-By Betrieb in jedem Fall verwendet. Sollten aber Massenströme vorhanden sein, so wird mit einem ebenfalls vom Benutzer vorgegebenem Faktor f_{crit} und dem vom Massenstrom abhängigen Δt_{crit} , ein maximal, zulässiger Zeitschritt Δt_{max} berechnet.

$$\Delta t_{max} = f_{crit} \Delta t_{crit} \quad (4.31)$$

Ist $\Delta t_{max} > \Delta t$ sein, so wird ebenfalls mit Δt gerechnet. Sollte das aber nicht der Fall sein, dann wird das vom Benutzer vorgegebene Intervall Δt in mehrere Teilintervalle, die kleiner als Δt_{max} sind, zerlegt, das Gleichungssystem, für jedes dieser Teilintervalle mit gleichen Randbedingungen, gelöst, und nur das Ergebnis zum Ende des ursprünglichen Intervalls zurückgegeben.

Die Schichtbeladung

Ein weiteres Modell, das den Beladevorgang einer Schichtladelanze wieder spiegelt, soll implementiert werden. Dazu muss das eben vorgestellte Modell lediglich ein wenig modifiziert, bzw. erweitert werden:

- Einströmvorgänge und damit verbundene Verwirbelungen werden nicht berücksichtigt. Die Parameter ϵ_{z+} und ϵ_{z-} zur Beschreibung dieser, werden in den Glg. 4.11 bis 4.16 gleich 1 gesetzt. Zur Quantifizierung konnten keine konkreten Verfahren, bzw. Experimente gefunden werden.
- Beim Beladevorgang wird vor jedem Zeitschritt die Schicht mit dem Index $i = i_{in}$ gesucht, deren Temperatur, am ehesten der Beladetemperatur ϑ_{down} entspricht. Der Belademassenstrom tritt in diese Schicht ein und fließt nach unten. Die darüber liegenden Schichten werden nicht durchströmt. Glg. 4.11 bis 4.16 wurden in geeigneter Weise angepasst.

Kapitel 5

Energie- und Exergieanalyse

Nach dem Lösen der Modellgleichung, aus dem vorhergehenden Kapitel, hat man zwar Kenntnis über die Temperaturverteilung über die Speicherhöhe in Abhängigkeit von der Zeit, jedoch kann man nur beschränkte Aussagen über die Qualität des Speichers bzw. seiner Betriebsweise machen.

Im folgenden Kapitel sollen nun Kennzahlen, Parameter, Wirkungsgrade (etc.) vorgestellt werden, die eine Beurteilung und im Idealfall auch eine Vergleichbarkeit von thermischen Schichtspeichern zulassen.

Thermodynamische Eigenschaften von Wasser

Zur energetischen Untersuchung von Speicherzyklen, aber auch zum Lösen der Bilanzgleichungen in den vorhergehenden Kapiteln, muss die Temperaturabhängigkeit einiger thermodynamischer Größen zur Verfügung stehen.

Die Gleichungen zur Ermittlung der Dichte ρ , der spezifischen Enthalpie h , der spezifischen Entropie s und der kinematischen Viskosität ν sind im „IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam“ enthalten. [Wag98]

Es wird angenommen, dass das Wasser immer im flüssigen Zustand vorhanden ist. Dann ist die Druckabhängigkeit der Eigenschaften, bei moderaten Drücken, vernachlässigbar klein.

Durch Einsetzen des Drucks im Normzustand $p = 1,013$ bar wurden die Druckterme aus den Gleichungen eliminiert. Die Rechenzeit verringert sich dadurch, mit kaum merkbarem Genauigkeitsverlust, entscheidend, da die Ermittlung der thermodynamischen Eigenschaften ziemlich rechenintensiv ist.

Mit steigenden Drücken wird aber auch der Einfluss derer, auf die Eigenschaften stärker, daher wurde der Gültigkeitsbereich der vereinfachten Gleichungen herabgesetzt:

$$p_S(\vartheta) \leq p \leq 16,5 \text{ bar} \quad (5.1)$$

$$0 \text{ }^\circ\text{C} \leq \vartheta \leq \vartheta_S(16,5 \text{ bar}) \approx 203 \text{ }^\circ\text{C} \quad (5.2)$$

Dieser Bereich sollte, für in der Praxis auftretende Betriebsbedingungen von thermischen Speichern, ausreichend groß gewählt worden sein.

Weiters wird die Temperaturabhängigkeit des thermischen Ausdehnungskoeffizienten, der folgendermaßen definiert ist, benötigt: [VDI10]

$$\beta = \frac{1}{v} \left(\frac{\partial v}{\partial T} \right)_p = -\frac{1}{\rho} \left(\frac{\partial \rho}{\partial T} \right)_p \quad (5.3)$$

Auf recht einfache Weise ergibt sich aus den oben erwähnten Eigenschaften die spezifische Exergie ξ und die integrale spezifische isobare Wärmekapazität \bar{c}_p :

$$\xi = h - h_{amb} - T_{amb} (s - s_{amb}) \quad (5.4)$$

$$h(\vartheta) = \bar{c}_p(\vartheta) \vartheta \quad (5.5)$$

Dabei ist T_{amb} die thermodynamische Temperatur der Umgebung, h_{amb} und s_{amb} die spezifische Enthalpie und die spezifische Entropie des Fluids bei Umgebungsbedingungen. Aus der Definition folgt, dass ξ bei Umgebungsbedingungen gleich 0 wird.

Globale Bilanzierung

Zur Untersuchung soll vorerst die Bilanzierung einer allgemeinen Zustandsgröße Φ über den gesamten Speicherinhalt dienen. Die Behälterinnenseite wird als Systemgrenze gewählt (vgl. Abb. 5.1).

$$\frac{d\Phi_{st}}{dt} = \dot{\Phi}_{st} = \underbrace{\dot{\Phi}_{ch} + \dot{\Phi}_{in,bottom} - \dot{\Phi}_{disch} - \dot{\Phi}_{out,bottom}}_{\dot{\Phi}_{flow}} + \dot{\Phi}_{loss} + \dot{\Phi}_{int} \quad (5.6)$$

$\dot{\Phi}_{st}$ stellt dabei die Änderungsrate der allgemeinen Zustandsgröße im Speicher selbst dar. Sie kann durch die Terme auf der rechten Seite der Gleichung beeinflusst werden: $\dot{\Phi}_{flow}$ fasst alle Einflüsse, die in Form von Stoffströmen die Systemgrenze passieren, zusammen. $\dot{\Phi}_{loss}$ lässt einen konduktiven Austausch mit der Umgebung zu. $\dot{\Phi}_{int}$ fasst die Änderungsrate aufgrund von Mischungen und Konduktion des Fluids selbst zusammen.

Integration vom Anfangszeitpunkt $t = t_{ini}$ bis zum betrachteten Zeitpunkt t liefert:

$$\Delta\Phi_{st} = \Phi_{st} - \Phi_{st,ini} = \underbrace{\Delta\Phi_{ch} + \Delta\Phi_{in,bottom} - \Delta\Phi_{disch} - \Delta\Phi_{out,bottom}}_{\Delta\Phi_{flow}} + \Delta\Phi_{loss} + \Delta\Phi_{int} \quad (5.7)$$

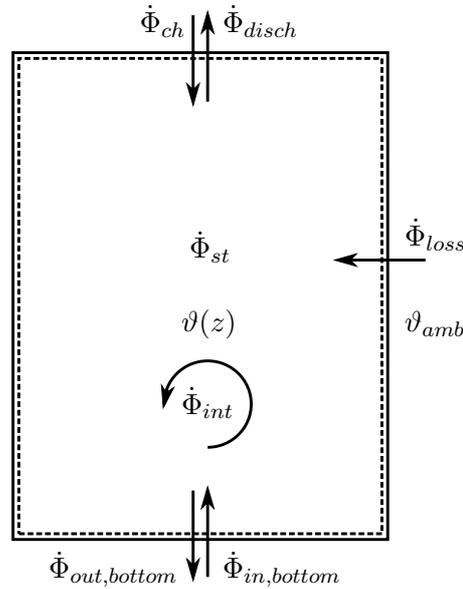


Abbildung 5.1: Globale Energiebilanz

Φ_{st} stellt die temperaturabhängige, allgemeine Zustandsgröße des Speicherinhalts im betrachteten Zeitpunkt dar:

$$\Phi_{st} = \int_{m_{KV}} \phi(\vartheta) dm, \quad \Phi_{st} = \sum_{i=1}^n \phi(\vartheta_i) m_i \quad (5.8)$$

Hierbei handelt es sich um die allgemeine analytische Berechnungsvorschrift und um die, für den diskretisierten Raum, anzuwendende Approximation. ϕ ist der spezifische Wert, also der auf die Masse m bezogene Wert von Φ .

Für den konvektiven Transport über die Systemgrenze ergibt sich:

$$\Delta\Phi_{konv} = \int_{t_{ini}}^t \dot{m} \phi(\vartheta) dt, \quad \Delta\Phi_{konv} = \sum_{j=1}^{nt} \dot{m}_j \phi(\vartheta_j) \Delta t_j \quad (5.9)$$

Hierbei handelt es sich wieder um die allgemeine analytische Berechnungsvorschrift und in dem Fall, um die für die diskretisierte Zeit anzuwendende Approximation. nt ist Anzahl der Zeitschritte und Δt_j die Dauer des betrachteten Zeitintervalls.

Zur Berechnung der restlichen Terme muss Φ präzisiert werden, da die Enthalpie $\Phi = H$, die Exergie $\Phi = \Xi$ und die Entropie $\Phi = S$ unterschiedlichen Berechnungsvorschriften unterliegen.

Die über die Systemgrenze ausgetauschte Energie ΔH_{loss} ⁷, Entropie ΔS_{loss} und Exergie $\Delta \Xi_{loss}$ ergeben sich zu:

$$\Delta H_{loss} = \int_{t_{ini}}^t \int_{A_{KV}} k (\vartheta - \vartheta_{amb}) dA dt, \quad \Delta H_{loss} = - \sum_{j=1}^{nt} \left[\sum_{i=1}^n (kA)_i (\vartheta_{i,j} - \vartheta_{amb}) \right] \Delta t_j \quad (5.10)$$

$$\Delta S_{loss} = \int_{t_{ini}}^t \int_{A_{KV}} k \frac{\vartheta - \vartheta_{amb}}{T_W} dA dt, \quad \Delta S_{loss} = - \sum_{j=1}^{nt} \left[\sum_{i=1}^n (kA)_i \frac{\vartheta_{i,j} - \vartheta_{amb}}{T_{i,j}} \right] \Delta t_j \quad (5.11)$$

$$\Delta \Xi_{loss} = \Delta H_{loss} - T_{amb} \Delta S_{loss} \quad (5.12)$$

Dabei wird angenommen, dass die Temperatur der Systemgrenze T_W gleich der Knotentemperatur $T_{i,j}$ ist.

Nun können die in Glg. 5.7 aufgrund interner Effekte auftauchenden Änderungen der Entropie ΔS_{int} und Exergie $\Delta \Xi_{int}$ ermittelt werden.

Mischungsvorgänge und Wärmeleitung im Fluid gehen isenthalp von statten, d.h. $\Delta \Phi_{int} = \Delta H_{int} = 0$. Aus dieser Tatsache lässt sich erkennen, dass solche internen Irreversibilitäten durch energetische Betrachtungen nicht erfasst werden können.

Beurteilung des thermischen Speichers

Wie in den vorhergehenden Kapiteln schon besprochen gibt es einige Faktoren, die die Schichtung im Speicher beeinflussen: Die Wärmeleitung des Fluids selbst, die Wärmeleitung in der Behälterwand und die daraus resultierende freie Konvektion, Mischungsvorgänge beim Einströmen und die Wärmeverluste durch die Speicherwand an die Umgebung. Im Modell wurden die ersten drei Effekte zu einer effektiven Wärmeleitfähigkeit ($\epsilon \lambda_{eff}$) vereint. Der Wärmedurchgangskoeffizient k beschreibt die Verluste thermischer Energie durch die Wand. Durch diese Parameter können unterschiedliche Speicher nur in beschränkter Weise miteinander verglichen werden. k lässt Aussagen über die Qualität der Isolierung zu. λ_{eff} fasst den vertikalen Transport thermischer Energie zusammen, und nimmt mit zunehmender Speicherdimension, kleinere Werte an.

Der entscheidende Punkt, der die Beurteilung von thermischen Speichern so schwierig macht, ist, dass die Effizienz enorm von der Betriebsweise und auch von der Betriebsdauer abhängt. Ein Speicherzyklus kann aus bis zu drei komplett unterschiedlichen Betriebsarten, nämlich

⁷Eigentlich müsste hier Q an Stelle von H geschrieben werden. Aus Gründen der Konsistenz wird aber darauf verzichtet.

Beladen, Stand-By und Entladen bestehen. Darüber hinaus können diese Betriebsarten mehrfach pro Zyklus auftauchen. Ein Vergleich, unterschiedlicher Schichtspeicher, ist also immer nur unter Angabe der Randbedingungen beim Test selbst möglich. [Deu12] beinhaltet z.B. standardisierte Prüfverfahren für Wasserspeicher von Solaranlagen.

In der Literatur finden sich viele unterschiedliche Namensgebungen und Definitionen von Kennzahlen zur Beurteilung von Speicherzyklen. Neben dem gebräuchlichen Begriff des Wirkungsgrades, spricht man oft von Nutzungsgrad, Rückgewinnungsfaktor (etc.).

Ziel ist es nun, geeignete Definitionen zu erkennen, die die Beurteilungen hinsichtlich der einzelnen Verluste, aber auch hinsichtlich der gesamten Verluste zulassen.

Dincer stellt in [Din11], einem sehr umfassenden Werk, einige mögliche Definitionen vor:

$$\eta_A = \frac{\Delta H_{disch} - \Delta H_{up,amb}}{\Delta H_{ch} - \Delta H_{down,amb}} \quad \psi_A = \frac{\Delta \Xi_{disch}}{\Delta \Xi_{ch}} \quad (5.13)$$

$$\eta_B = \frac{(\Delta H_{disch} - \Delta H_{up,amb}) + \Delta H_{st}}{\Delta H_{ch} - \Delta H_{down,amb}} \quad \psi_B = \frac{\Delta \Xi_{disch} + \Delta \Xi_{st}}{\Delta \Xi_{ch}} \quad (5.14)$$

$$\eta_C = \frac{\Delta H_{disch} - \Delta H_{up,amb}}{(\Delta H_{ch} - \Delta H_{down,amb}) + (H_{ini} - H_{amb})} \quad \psi_C = \frac{\Delta \Xi_{disch}}{\Delta \Xi_{ch} + \Xi_{ini}} \quad (5.15)$$

$$\eta_D = \frac{(\Delta H_{disch} - \Delta H_{up,amb}) + (H_{st} - H_{amb})}{(\Delta H_{ch} - \Delta H_{down,amb}) + (H_{ini} - H_{amb})} \quad \psi_D = \frac{\Delta \Xi_{disch} + \Xi_{st}}{\Delta \Xi_{ch} + \Xi_{ini}} \quad (5.16)$$

Es wird grundsätzlich zwischen energetischer, d.h. Vergleich von Enthalpien und exergetischer Betrachtung, d.h. Vergleich von Exergien, unterschieden. Wie oben schon erwähnt, können Mischungsvorgänge des Fluids selbst, durch energetische Betrachtungen nicht erfasst werden. $\Delta H_{up,amb}$ und $\Delta H_{down,amb}$ sind jene Energiemengen, die ein gleicher Stoffstrom mit Umgebungstemperatur über die Systemgrenze transportieren würde. H_{amb} ist der Energieinhalt des Speichers, wenn er sich im thermischen Gleichgewicht mit seiner Umgebung befindet.

Definitionsgemäß wird die Exergie bei Umgebungsbedingungen gleich Nullpunkt, daher gilt: $\Delta \Xi_{up,amb} = \Delta \Xi_{down,amb} = \Xi_{amb} = 0$.

Viele Prüfverfahren starten unter thermischem Gleichgewicht mit der Umgebung, d.h. $H_{ini} = H_{amb}$ bzw. $\Xi_{ini} = \Xi_{amb} = 0$. Die Wirkungsgrade A und C, sowie die Wirkungsgrade B und D, nehmen dann idente Werte an. Weiters kann man leicht sehen, dass die Definitionen A und C, nur zur Untersuchung von Entladevorgängen geeignet sind.

Aus der Definition D erhält man eine weitere, mögliche Betrachtung, wenn man die am Speicherboden eintretenden und austretenden Massenströme mitberücksichtigt:

$$\eta_E = \frac{(\Delta H_{disch} - \Delta H_{up,amb}) + (\Delta H_{out,bottom} - \Delta H_{down,amb}) + (H_{st} - H_{amb})}{(\Delta H_{ch} - \Delta H_{down,amb}) + (\Delta H_{in,bottom} - \Delta H_{up,amb}) + (H_{ini} - H_{amb})} \quad (5.17a)$$

$$\psi_E = \frac{\Delta \Xi_{disch} + \Delta \Xi_{out,bottom} + \Xi_{st}}{\Delta \Xi_{ch} + \Delta \Xi_{in,bottom} + \Xi_{ini}} \quad (5.17b)$$

Vergleicht man die Definitionen D und E, so wird deutlich, dass Definition D viel stärker von der Betriebsführung abhängt. Durch einen sehr ungünstigen Betrieb, z.B. Beladen eines bereits vollen Speichers, wächst der Nenner in D immer weiter an und der Wirkungsgrad fällt. Das ist bei Definition E nicht der Fall, da der Zähler ebenso durch den austretenden Massenstrom am Speicherboden wächst. Welche der beiden Definitionen nun die geeignetere ist, hängt also ganz stark von der Fragestellung ab. Die unterschiedlichen Definitionen werden in Abschnitt 6.2 anhand eines konkreten Anwendungsfalls veranschaulicht und genauer diskutiert.

Ein völlig anderer Zugang ist der Vergleich des zu untersuchenden Speichers mit einem idealen Referenzspeicher. Die mit Glg. 3.19 vorgestellte Beladeeffizienz liegt einem idealen Einspeichervorgang zu Grunde. Eine ähnliche Herangehensweise wird in [Hal10] gezeigt. Als Referenzspeicher wird der schlechtest mögliche Fall, der vollständig durchmischte Speicher verwendet. Dieser besitzt zu Beginn den gleichen Energieinhalt wie der eigentliche Speicher, nur herrscht im gesamten Speicher konstante Temperaturverteilung. Er wird unter den gleichen Randbedingungen betrieben, eintretendes Fluid mischt sich sofort.

Folgende Definition vergleicht die internen Irreversibilitäten, mit dem schlechtest möglichen Betrieb, gekapselt von den anderen Verlusten:

$$\psi_{int} = 1 - \frac{\Delta \Xi_{int}}{\Delta \Xi_{int,m}} = 1 - \frac{\Delta S_{int}}{\Delta S_{int,m}} \quad (5.18)$$

Der Exergieverlust ist bei Mischungsvorgängen direkt proportional zur Entropieproduktion:

$$T_{amb} \Delta S_{int} = -\Delta \Xi_{int} \quad (5.19)$$

Eine weitere Definition wird ebenfalls in [Hal10] vorgestellt. Dabei werden nicht nur die internen Verluste, sondern auch die Verluste durch Wärmeleitung über die Systemgrenzen, berücksichtigt:

$$\psi_{int,loss} = 1 - \frac{\Delta \Xi_{int} + \Delta \Xi_{loss}}{\Delta \Xi_{int,m} + \Delta \Xi_{loss,m}} \quad (5.20)$$

Die Aussagekraft hinsichtlich der Gesamtverluste ist aber anzuzweifeln. Aufgrund des allgemein tieferen Temperaturniveaus des vollständig durchmischten Speichers, ist die Tempera-

turdifferenz zur Umgebung, und damit die treibende Kraft der Exergieverluste, kleiner als bei dem zu untersuchenden Speicher. Es folgt, dass $|\Delta \Xi_{loss,m}| < |\Delta \Xi_{loss}|$ ist, der vermeintlich schlechte Referenzspeicher performt, hinsichtlich der Exergieverluste, besser.

Es wurden nun einige mögliche Definitionen und Herangehensweisen gezeigt, die an diesem Punkt noch etwas abstrakt erscheinen können. Im Abschnitt 6.2 wird ein konkreter Speicherzyklus ausgewertet und diskutiert. Die Stärken und Schwächen der einzelnen Definitionen sollten dann besser zur Geltung kommen.

Kapitel 6

Programmbeschreibung und Simulation

Ziel ist es nun, die fertigen mathematischen Modelle und alle Funktionen die zusätzlich rund um das Modell benötigt werden, mithilfe einer geeigneten Programmiersprache zu verarbeiten. Es handelt sich zum einen um das Speichermodell mit Direktbeladung am Speicherdeckel, und zum anderen um das etwas modifizierte Modell mit Schichtladelanze.

Die Speichermodelle sollen später, in das in der Einleitung erwähnte Modell des thermischen Netzes, leicht implementierbar sein. Sie werden als Pufferspeicher für Lastspitzen oder Saisonalspeicher verwendet. Das erfordert eine klar definierte, einfache Schnittstelle zum Datenaustausch zwischen den Modellen.

In weiterer Folge soll das Fernwärmenetz mit einer Vielzahl von dezentral angeordneten Pufferspeichern instationär simuliert werden können. Das bedeutet, dass das Modell parallel mit unterschiedlichen Konfigurationen und Randbedingungen aufrufbar und lauffähig sein muss. Außerdem sollen die Speichermodelle in andere Systeme leicht integrierbar sein. Beispielsweise als kleiner Pufferspeicher in häusliche Heizsysteme. Bei solarthermisch gestützten System spielt dann das Modell mit Schichtladelanze eine wichtige Rolle.

Darüber hinaus sind auch Simulationen von Testsequenzen am autonomen Speicher, also ohne Systemintegration, von Interesse.

Die Wahl der Programmiersprache fiel auf MATLAB, da es sich um ein sehr mächtiges Paket zur Lösung mathematischer Probleme und zur Visualisierung der Ergebnisse handelt. Diese Hochsprache ist primär für numerische Berechnungen mithilfe von Matrizen und Vektoren ausgelegt. Das Programmpaket liefert viele vordefinierte Funktionen zur numerischen Integration, Signalauswerten (etc.) mit. Externe Applikationen und einiger anderer Programmiersprachen entstammender Quellcode können über Schnittstellen leicht eingebunden werden. [Mat13a]

6.1 Beschreibung ausgewählter Funktionen

Wie oben erwähnt, sollen mehrere thermische Speicher gleichzeitig simuliert werden können. Man arbeitet daher am besten mit Funktionen. Diese werden in einem eigenen Arbeitsbereich abgearbeitet, es gibt daher keine unerwünschten Querverbindungen. Die Kommunikation nach Außen erfolgt durch Übergabeparameter.

Die wichtigste Funktion, die ebenfalls das Modell selbst darstellt, trägt den Namen *STEP_FILO.m*. *FILO* steht für *First In Last Out*. Es handelt sich um das Modell mit direkter Beladung am Speicherdeckel, der Speicherinhalt wird nach unten verdrängt und das kalte Wasser tritt ganz unten aus. Der Beladevorgang erfolgt im umgekehrten Sinn.

Die Funktion wird zu Beginn jedes Zeitschritts *STEP* aufgerufen und gibt den Zustand am Ende des Zeitschritts zurück. Dabei müssen alle notwendigen Parameter vom Benutzer selbst, oder vom System in das das Speichermodell integriert ist, übergeben werden. Klarerweise müssen die Kommunikationspartner die selben diskreten Zeitpunkte und Zeitschrittweiten verwenden.

Die In- und Output Variablen:

- *storage* beinhaltet die nötigen Speicherparameter wie Höhe, Durchmesser, Wärmedurchgänge und effektive Wärmeleitfähigkeit des Fluid.
- Mit *Geometry_node* werden die Anzahl der Knoten und alle möglichen Abmessungen dieser übergeben. Die Knotenabmessungen könnten eigentlich in der Funktion selbst berechnet werden, da es sich um ein äquidistantes Gitter handelt. Man nimmt allerdings etwas mehr Speicherbedarf in Anspruch, da die Priorität auf Seiten der Rechenzeit liegt.
- Die Struktur *Inlet* beschreibt die Einlaufgeometrie, mit Durchmesser und charakteristischen Werten der Fähigkeit zum Impulsabbau. Sollten solche Daten nicht zur Verfügung stehen, kann durch Null setzen der letzten Strukturvariable, das Miteinbeziehen der Zulaufgeometrie in die Berechnung, ausgekoppelt werden.
- Randbedingungen wie Massenströme und Temperaturen müssen ebenfalls übergeben werden. Abb. 6.1 soll verdeutlichen, zu welchem Zeitpunkt bzw. über welche Zeitdauer die einzelnen Variablen auftauchen. Einer der beiden Massenströme \dot{m}_{up} und \dot{m}_{down} muss immer Null gesetzt sein, ansonsten bricht das Programm ab und gibt eine Errormeldung aus.
- Die Zeitschrittweite Δt und der kritische Zeitschrittfaktor f_{crit} müssen ebenfalls übergeben werden (vgl. Kapitel 4.2).

- Die Rückgabewerte sind die Temperaturverteilung ϑ^1 am Ende des Zeitschritts und die Austrittstemperaturen am Speicherboden $\vartheta_{out,bottom}^1$ und -deckel $\vartheta_{out,top}^1$ während des Zeitschritts. Die Austrittsmassenströme werden nicht gesondert zurückgegeben, da diese mit den Eintrittsmassenströmen übereinstimmen.

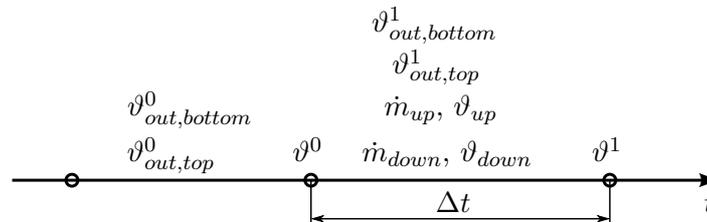


Abbildung 6.1: Zeitachse

Es ist zu beachten, dass alle Inputs und Outputs auf ihre SI-Einheiten bezogen sind. Für detailliertere Beschreibungen ist der Quellcode dieser und aller anderen Funktionen im Anhang A angefügt.

Das zweite, geringfügig modifizierte Modell trägt den Dateinamen *STEP_VAR_IN.m* (*Variable Inlet*). Die Einspeicherhöhe ist nun von der Beladetemperatur und der Temperaturverteilung im Speicher abhängig. Der einzige Unterschied im Aufruf des Modell zum eben beschriebenen ist lediglich, dass die Parameterstruktur *Inlet* wegfällt. Es werden also Verwirbelungen an der Zuströmöffnung nicht berücksichtigt.

Zusätzlich wurden 3 Funktionen (*TESTING_L.m*, *TESTING_NB.m* und *TESTING_V.m*), die verschiedene Testsequenzen am Speichermodell *STEP_FILO* abarbeiten, implementiert. Es handelt sich dabei um die in [Deu12] vorgestellten Prüfverfahren für Wasserspeicher in Solaranlagen.

Eine weitere Funktion *ANALYSE_CYCLE.m* zum Auswerten der Ergebnisse eines Speicherzyklus steht zur Verfügung. Sie liefert alle möglichen Enthalpie-, Exergie- und Entropieänderungen, sowie verschiedenste Wirkungsgrade (vgl. Kapitel 5).

Für genauere Beschreibungen sei hier wieder auf den Quellcode in Anhang A verwiesen. Zur Veranschaulichung wird in Abschnitt 6.2 anhand eines Simulationsbeispiels der Zyklus V genauer erläutert und ausgewertet.

Einbinden in Simulink

Als kleine Draufgabe wurden die Modelle in SIMULINK, einem Softwarepaket das zur Modellierung verschiedenster Systeme durch grafische Blöcke dient, eingebunden. Es ist ein Zusatzprodukt von Matlab und benötigt dieses zum Ausführen. Eine Vielzahl von vordefinierten

Blöcken werden in mitgelieferten Bibliotheken zur Verfügung gestellt. Darüber hinaus kann mit sogenannten S-Functions eigener Code in das Modell integriert werden. [Mat13b]

Dazu wurde eine eigene Bibliothek *Storage_Lib.slx* angelegt die zwei Modellblöcke enthält. Hinter diesen Blöcken verbergen sich diese S-Functions, die einem speziellen Abarbeitungsprozess folgen. Daher mussten die Modellfunktionen etwas abgeändert werden (*STEP_FILO_modfsl.m* und *STEP_VAR_IN_modfsl.m*). Die Speicherparameter werden durch sogenannte Blockparameter übergeben, die Randbedingungen und Simulationsergebnisse entsprechen den Blockanschlüssen.

Die Ergebnisse aus Simulink sind aber mit Vorsicht zu genießen. Es konnten kleine Abweichungen zu den Ergebnissen aus der Matlab Simulation erkannt werden. Bereits bei der Initialisierung waren Unterschiede vorhanden, deren Ursprung leider nicht gefunden werden konnte.

Abschließend lässt sich sagen, dass das Simulink Modell durch die grafische Oberfläche auf den ersten Blick verständlich und leicht interpretierbar ist. Es ähnelt in seiner Darstellung einem physischen Speicher mit Rohranschlüssen. Die Systemintegration ist daher schnell und einfach möglich.

Das Matlab Modell hingegen erscheint etwas abstrakter und erfordert eine gewisse Zeit zur Einarbeitung. Hat man sich aber einmal damit auseinandergesetzt, so spielt es seine Vorteile klar aus. Die Flexibilität und Durchsichtigkeit lassen Eingriffe und Änderungen vom außen leicht zu.

6.2 Demonstratives Beispiel

In diesem Abschnitt werden die Ergebnisse und deren Auswertungen einer konkreten Speichersimulation vorgeführt. Das Prüfverfahren V aus [Deu12], wird am Speicher mit in der Praxis gängigen Parametern und Randbedingungen abgearbeitet:

- Speicherparameter: Speicherhöhe $H = 1,3$ m, Speicherdurchmesser $D = 0,7$ m, radialer Wärmedurchgang $k_{rad} = 1$ W/m²K, axialer Wärmedurchgang $k_{ax} = 1$ W/m²K, effektive Wärmeleitfähigkeit $\lambda_{eff} = 1,6$ W/mK
- Geometrie der Zuströmung: Einströmdurchmesser $D_{in} = 0,05$ m, Aufpralleinströmung mit $a = 4,75$ und $b = 0,522$, $\epsilon(z)$ hat hyperbolischen Charakter
- Knotenanzahl $n = 100$
- kritischer Zeitschrittfaktor $f_{crit} = 1$
- Umgebungstemperatur $\vartheta_{amb} = 20$ °C

Die Simulation besteht aus insgesamt 6 aufeinander folgender Betriebsperioden und beginnt mit der Konditionierung V1. Der Speicher befindet sich anfangs im thermischen Gleichgewicht mit seiner Umgebung $\vartheta_{ini} = 20 \text{ }^\circ\text{C}$. Abb. 6.2 zeigt den Verlauf der Massenströme über die Zeit. \dot{m}_{down} steht für einen Massenstrom nach unten, einem Beladevorgang, \dot{m}_{up} für Fließrichtung nach oben, einem Entladevorgang. Die dabei auftretenden Ein- und Austrittstemperaturen sind Abb. 6.3 zu entnehmen.

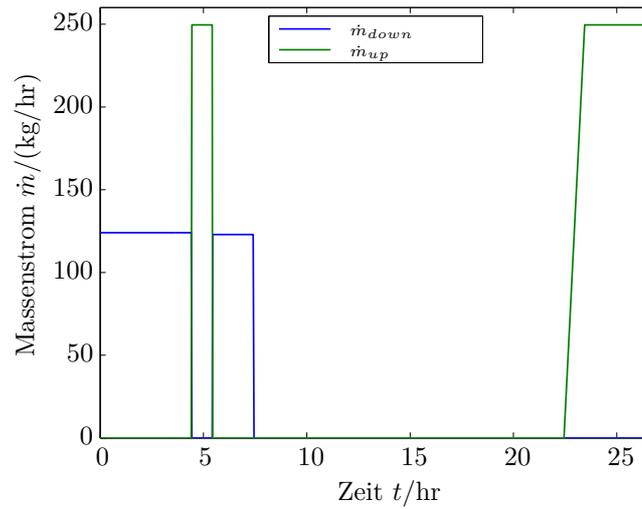


Abbildung 6.2: Be- und Entlademassenstrom

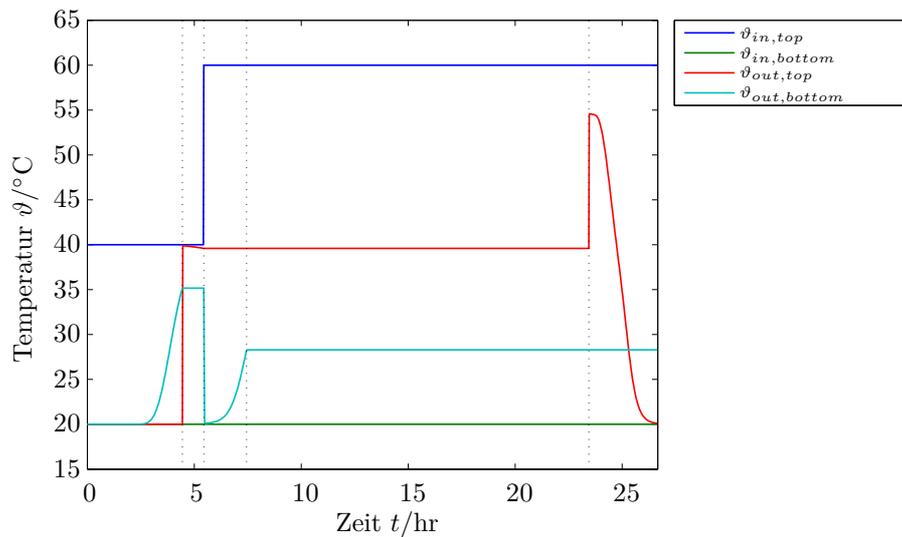


Abbildung 6.3: Ein- und Austrittstemperaturen

Zum Zeitpunkt 0 startet der erste Beladevorgang V2 mit einem über die Betriebsperiode konstanten Massenstrom $\dot{m}_{down} = 125 \text{ kg/hr}$ und einer Beladetemperatur von $\vartheta_{in,top} = 40 \text{ }^\circ\text{C}$.

Langsam bildet sich eine thermische Schichtung aus (vgl. Abb. 6.4), die sich mit fortschreitender Beladung immer weiter nach unten bewegt. Der anfangs recht große Temperaturgradient wird mit fortschreitender Zeit durch die Wärmeleitung im Fluid abgebaut. Durch die numerische Diffusion verstärkt sich dieser Effekt ein wenig, sie spielt aber gegenüber der Wärmeleitung eine untergeordnete Rolle. Nach einer Zeitdauer von $\Delta t_2 \approx 4,44$ hr ist das Abbruchkriterium $\vartheta_{out,bottom} = 35$ °C für den Beladevorgang erreicht. Es herrscht annähernd konstante Temperaturverteilung im gesamten Speicher.

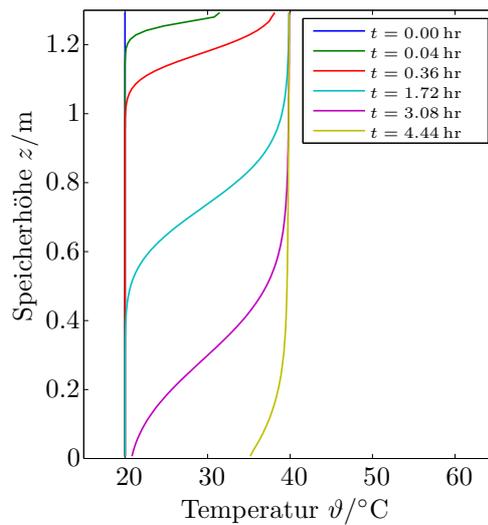


Abbildung 6.4: Temperaturverlauf beim ersten Beladevorgang

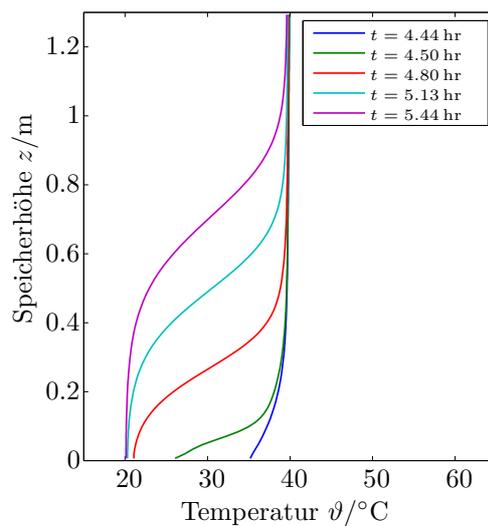


Abbildung 6.5: Temperaturverlauf beim ersten Entladevorgang

Nun folgte die Entladung V3 des halben Speichervolumens $V_{disch} = V/2$. Wie man Abb. 6.5 entnehmen kann, wird die Temperaturfront wieder nach oben gedrängt. Der Massenstrom ist mit $\dot{m}_{up} = 250$ kg/hr doppelt so groß wie beim vorhergehenden Beladen. Dieser Betriebsabschnitt ist daher bereits nach $\Delta t_3 \approx 1$ hr beendet.

Ein zweiter Beladevorgang V4, mit einem Massenstrom von $\dot{m}_{down} = 125$ kg/hr und einer Temperatur von $\vartheta_{in,top} = 60$ °C, wird bis zum Austausch des halben Speichervolumens $V_{ch} = V/2$ simuliert.

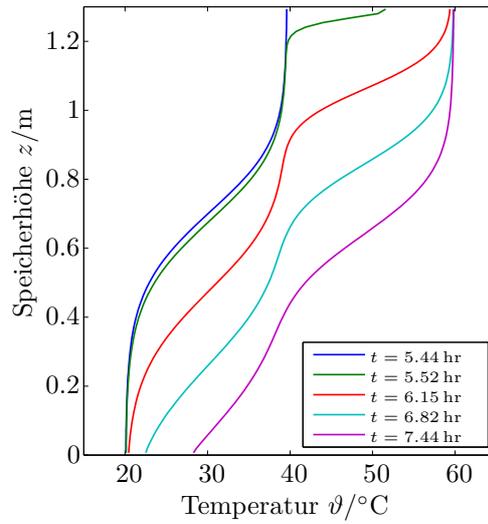


Abbildung 6.6: Temperaturverlauf beim zweiten Beladevorgang

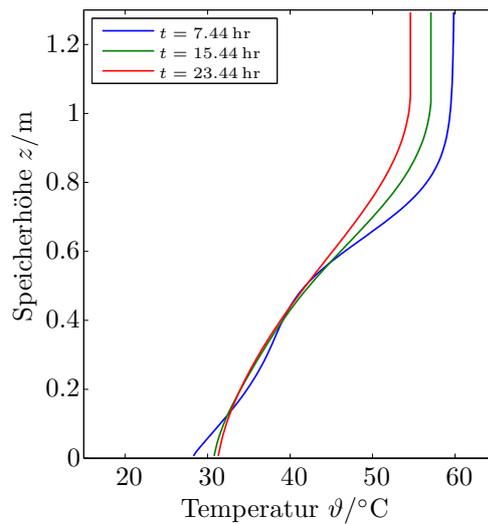


Abbildung 6.7: Temperaturverlauf im Stand-By Betrieb

Am Ende des zweiten Beladevorgangs sind zwei Temperaturschichtungen erkennbar, die im anschließenden Stand-By-Betrieb V5 zu einer verschmelzen. Der Speicher wird $\Delta t_5 = 16$ hr sich selbst überlassen. Die radialen Verluste am oberen Speichermantel und die axialen Verluste am Deckel, lassen die Speichertemperatur lokal sinken. Dieser Effekt wird durch den axialen Wärmetransport im Fluid verstärkt.

Aufgrund der niedrigen Temperaturunterschiede der unteren Regionen zur Umgebung, sind hier die radialen Verluste durch den Mantel nicht so stark ausgeprägt. Sie werden sogar vom axialen Wärmetransport übertrumpft und die Temperatur steigt an.

Es folgt die abschließende Entladung V6, mit einem Massenstrom von $\dot{m}_{up} = 250$ kg/hr und einer Entladetemperatur von $\vartheta_{in,bottom} = 20$ °C. Nach $\Delta t_6 \approx 3,23$ hr wird am Speicherdeckel die Austrittstemperatur $\vartheta_{out,top} \approx 20$ °C erreicht. Die Simulation ist damit beendet.

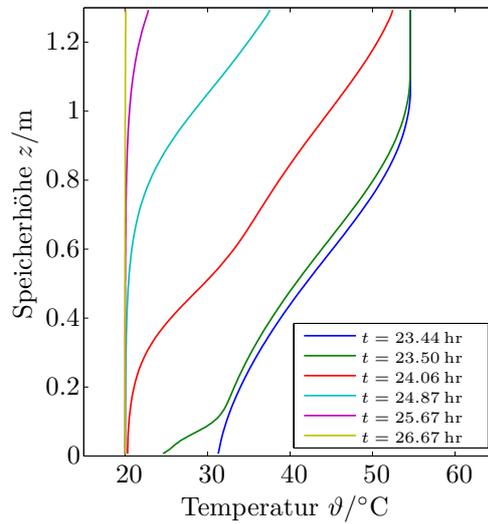


Abbildung 6.8: Temperaturverlauf beim vollständigen Entladen

Es ergeben sich insgesamt 26,67 hr Simulationszeit, die von einem Intel Core 2 Duo P8600 Prozessor mit einer Taktfrequenz von 2,40 GHz abgearbeitet wurden.

Tabelle 6.1: Rechenzeit

kritischer Zeitschrittfaktor f_{crit} :	1	2	3
Simulationszeit:	26,67 hr		
Rechenzeit:	3,019 s	1,697 s	1,271 s
	100 %	56,2 %	42 %
Simulationszeit/Rechenzeit:	31.803	56.577	75.541
Rechenzeitanteil für Stoffeigenschaften:	67,3 %	64,7 %	61,8 %

In Tab. 6.1 sind einige markante Werte zusammengefasst. Dazu wurden noch zwei weitere Simulationen des selben Speicherzyklusses, mit anderen kritischen Zeitschrittfaktoren durchgeführt. Das Erhöhen von f_{crit} ändert allerdings nichts an der Zeitschrittweite im Stand-By-Betrieb, der den Großteil der Simulationszeit einnimmt. Lediglich die zulässige Zeitschrittweite in massenstrombehafteten Betriebsperioden wird erhöht. Die dadurch enorme Reduktion der Rechenzeit lässt nun folgern, dass der Hauptanteil der Rechenzeit für massenstrombehaftete Betriebsperioden aufgewendet wird.

Eine weitere wichtige Aussage ist, dass bei allen drei Simulationen, der Rechenzeitanteil zur Ermittlung der Stoffeigenschaften, über 60 % der gesamten Rechenzeit ausmacht. Hier ist Einsparungspotential erkennbar.

Beurteilung von Speicher und Zyklus

Zur Beurteilung der Effizienz werden zunächst die in Kapitel 5 definierten Enthalpie-, Exergie- und Entropieänderungen betrachtet. Volllinien repräsentieren den oben simulierten Speicher, Strichlinien den vollständig durchmischten Referenzspeicher. Die vertikalen, punktierten Linien trennen die einzelnen Betriebsperioden.

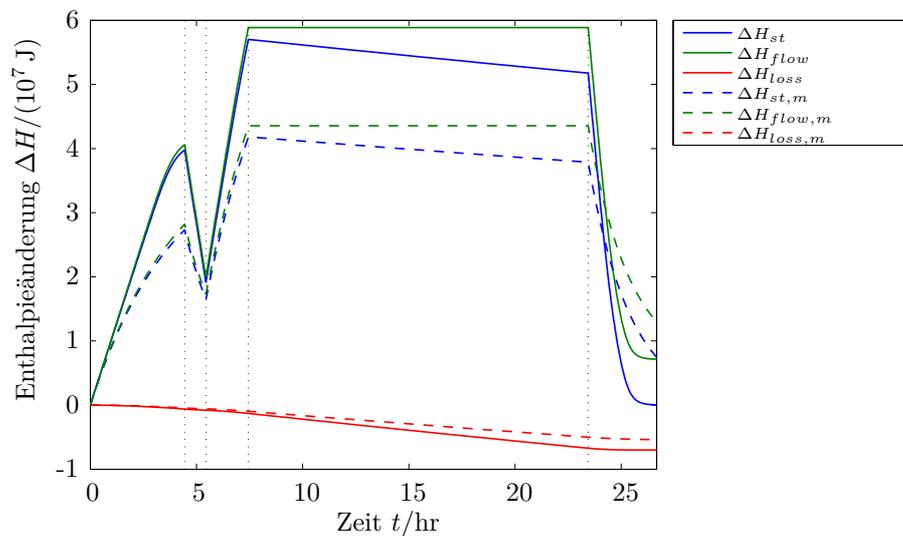


Abbildung 6.9: Enthalpieänderung

Abb. 6.9 zeigt den Verlauf der Enthalpieänderung über der Zeit. Man kann gut erkennen, wie während der Be- und Entladephasen der Energieinhalt des Speichers ΔH_{st} mit der konvektiv über die Systemgrenze transportierte Enthalpiemenge ΔH_{flow} wächst, bzw. fällt. Im Laufe der Zeit nehmen die Speicherverluste aufgrund von Wärmeleitung durch die Wand ΔH_{loss} zu. Die Zunahme ist jedoch in Abschnitten höherer Beladung, d.h. höheren Temperaturniveaus, stärker ausgeprägt. Der Referenzspeicher performt in diesem Zusammenhang besser.

Sein Temperaturniveau ist im allgemeinen, aufgrund des sofortigen Mischens des eintretenden Fluids mit dem gesamten Speicherinhalt, niedriger. Interne Effekte sind hier nicht erkennbar, Mischungsvorgänge gehen isenthalp von statten.

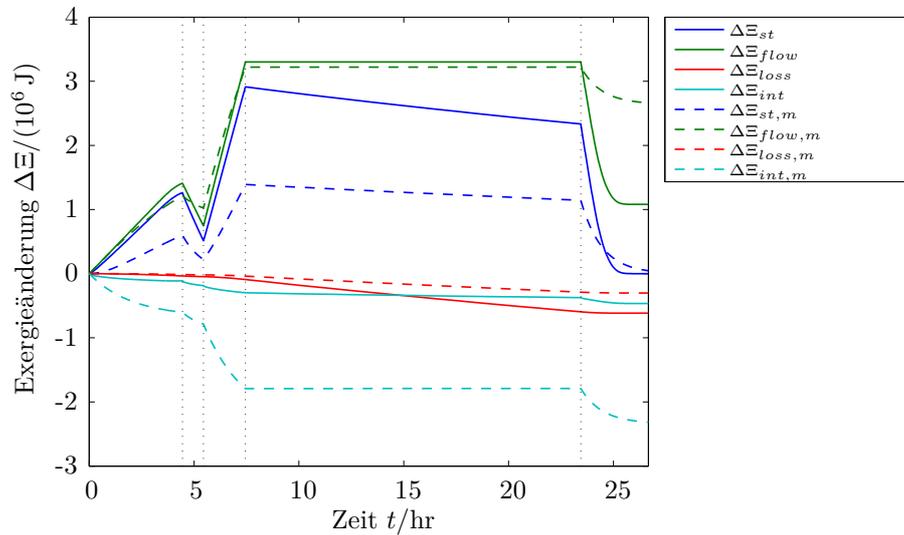


Abbildung 6.10: Exergieänderung

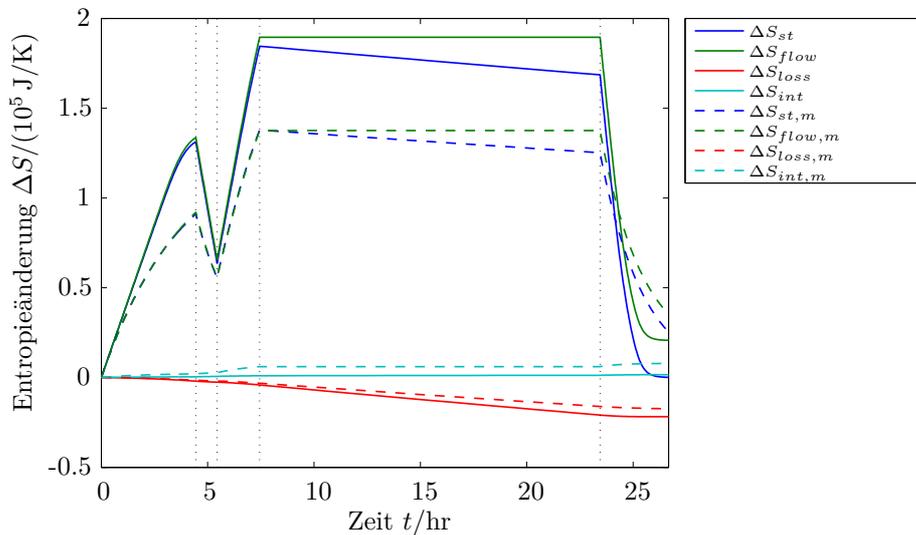


Abbildung 6.11: Entropieänderung

In Abb. 6.10 sind die Exergieänderungen dargestellt. Der Speicherinhalt $\Delta \Xi_{st}$ und der konvektive Anteil $\Delta \Xi_{flow}$ zeigen das gleiche Verhalten wie oben. Nun sind die konduktive Verluste $\Delta \Xi_{loss}$ und internen Irreversibilitäten $\Delta \Xi_{int}$ separat darstellbar. Aus Mischungsvorgängen und Wärmeleitung im Fluid resultiert ein Exergieverlust. Eine besonders starke Änderung ist immer zu Beginn von Be- bzw. Entladevorgängen zu erkennen. Die Temperaturunterschiede

zwischen Speichertemperatur am Eintritt und der des einströmenden Fluids, sind hier am größten. Mischungsvorgänge sind dann besonders stark ausgeprägt. Im Referenzspeicher, in dem sich der gesamte Speicherinhalt sofort mischt, sind diese Effekte natürlich am gravierendsten. $\Delta \Xi_{loss}$ nimmt mit der Zeit sukzessive zu, in Zeiten höheren Temperaturniveaus jedoch stärker.

Die Entropieänderungen sind in Abb. 6.11 zu sehen. Exergieverluste $\Delta \Xi_{int} < 0$ werden durch Entropieproduktion hervorgerufen $\Delta S_{int} > 0$. Entropieänderungen, aufgrund konduktiver Leitung durch die Wand, sind allerdings negativ $\Delta S_{loss} < 0$.

Die folgenden beiden Abb. veranschaulichen die in Kapitel 5 vorgestellten energetischen und exergetischen Wirkungsgrade. Sie sind das primäre Ziel dieser Speicheranalyse.

Im Initialzustand ist der Speicher im thermischen Gleichgewicht mit seiner Umgebung, d.h. die Definitionen A und C bzw. die Definitionen B und D liefern das gleiche Ergebnis.

η_C und ψ_C zeigen grob abweichendes Verhalten von den anderen Definitionen. Wie oben schon erwähnt, sind diese nur zur Beurteilung von Entladevorgängen geeignet, nicht aber für den gesamten Speicherzyklus. Sie sind hier nur zur Untermauerung der obigen Aussage eingezeichnet.

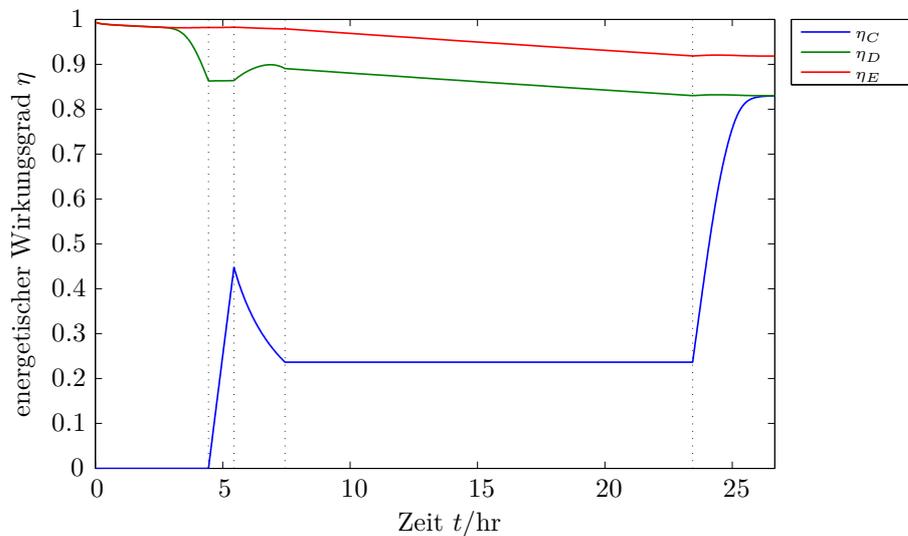


Abbildung 6.12: Energetischer Wirkungsgrad

Energetische Betrachtungen identifizieren keine Mischungsvorgänge und Wärmeleitungseffekte des Speicherinhalts. Ein solcher Ansatz würde also eine gesonderte Beurteilung, hinsichtlich der Speicherverluste über die Behälterwand, ermöglichen. An Stelle von Glg. 5.17a, zur Definition von η_E , könnte man ebenso schreiben (ΔH_{loss} ist unter realen Betriebsbedingungen stets negativ):

$$\eta_E = 1 + \frac{\Delta H_{loss}}{(\Delta H_{ch} - \Delta H_{down,amb}) + (\Delta H_{in,bottom} - \Delta H_{up,amb}) + (H_{ini} - H_{amb})} \quad (6.1)$$

Somit wird klar, dass es sich um eine isolierte Betrachtung der Verluste thermischer Energie an die Umgebung handelt. η_E in Abb. 6.12 bestätigt diesen Sachverhalt. Der Wirkungsgrad nimmt sukzessive ab und steigt zu keinem Zeitpunkt. Bestenfalls wäre ein horizontaler Verlauf, falls der Speicher im thermischen Gleichgewicht mit seiner Umgebung ist, möglich.

Zur Untersuchung der gesamten Verluste und internen Verluste im speziellen, sind exergetische Betrachtungen zielführender.

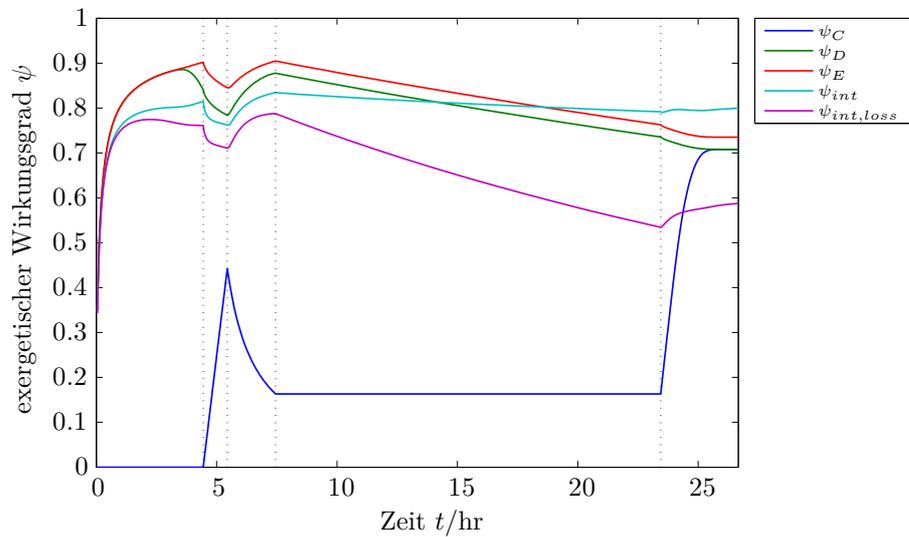


Abbildung 6.13: Exergetischer Wirkungsgrad

Zur Beurteilung der gesamten Speicherverluste bieten sich die Definitionen D und E an. Wie oben schon erwähnt, kommt in D eine ungünstige Betriebsführung stärker zur Geltung. Erst zum Ende des ersten Beladevorgangs kommt es in Abb. 6.13 zu merkbaren Unterschieden. Das resultiert aus dem andauernden Beladen, trotz sehr hoher Austrittstemperatur $\vartheta_{out,bottom}$ (vgl. Abb. 6.3). Hochwertige Exergie wird am Speicherboden hinaus gedrückt. Danach verlaufen die Kurven annähernd parallel, das spricht von einer guten Betriebsführung.

ψ_{int} gibt Aussage über die Irreversibilitäten im Speicherinneren. Es wird mit dem schlechtestmöglichen Referenzspeicher verglichen.

$\psi_{int,loss}$ ist zur Beurteilung der Gesamtverluste eher ungeeignet (vgl. Kapitel 5).

Kapitel 7

Zusammenfassung

In dieser Arbeit werden Modelle zur numerischen Simulation thermischer Wasserspeicher in der Hochsprache Matlab erstellt. Bei den Speichern handelt es sich um Schichtspeicher die sich durch direktes Be- bzw. Entladen, also durch Austausch des Speichermediums selbst, auszeichnen.

Kapitel 2 schafft einleitend einen groben Überblick zur thermischen Energiespeicherung. In weiterer Folge wird Wasser als Speichermedium hinsichtlich Vorteile, Nachteile, Einsatzmöglichkeiten (etc.) genauer betrachtet.

Ausgehend von den allgemeinen dreidimensionalen Bilanzgleichungen für Masse, Impuls und Energie, wird in Kapitel 3 Schritt für Schritt die Modellgleichung abgeleitet.

Aus der Betrachtung des vermeintlich ruhenden Speichers können wichtige Aussagen getätigt werden. Das Speichermedium befindet sich nur dann im mechanischen Gleichgewicht, wenn es sich auch im thermischen Gleichgewicht mit seiner Umgebung befindet. Andernfalls werden, durch vertikale Wärmeleitung in der Behälterwand und durch die Verluste an die Umgebung, frei Konvektionen hervorgerufen. Die stärkere Durchmischung, kann durch Einführung einer effektiven Wärmeleitfähigkeit, in ein eindimensionales Wärmeleitungsproblem reduziert werden.

Zur Beurteilung der Strömungsform im nicht ruhenden Speicher, eignet sich die Richardson-Zahl. Für $Ri > 0,25$ kann von idealer Kolbenströmung ausgegangen werden. Mit kleiner werdender Richardson-Zahl treten immer stärkere Verwirbelungen auf, die wiederum durch lokales Anheben der Wärmeleitfähigkeit quantifizierbar sind.

Das Ergebnis ist die eindimensionale Transportgleichung der thermischen Energie. Kapitel 4 zeigt den Ablauf der räumlichen Diskretisierung mithilfe der Finite-Volumen-Methode. Die Zeitintegration erfolgt durch das Crank-Nicolson-Verfahren. Dieses bietet einen guten Kompromiss aus aufzuwendender Rechenzeit und erzielbarer Genauigkeit. Das so gewonnene algebraische Gleichungssystem, samt programmiertechnischer Implementierung, stellt das fertige Speichermodell dar. Einige ausgewählte Funktionen werden in Kapitel 6 näher beschrieben.

Zur Beurteilung von thermischen Speichern bzw. Speicherzyklen, stellt Kapitel 5 einige Wirkungsgraddefinitionen vor. Energetische Betrachtungen lassen nur Aussagen hinsichtlich der Verluste durch die Behälterwand an die Umgebung zu. Mischungsvorgänge und Wärmeleitung im Fluid können nur durch exergetische Betrachtung erfasst werden.

Zur Demonstration wird abschließend anhand eines konkreten Beispiels ein Speicherzyklus simuliert, grafisch veranschaulicht und diskutiert. Das Auswerten der Rechenzeit ergibt, dass der Großteil der Rechenzeit zur Ermittlung der Stoffeigenschaften des Wassers aufgewendet wird.

Es steht nun ein Rechenmodell thermischer Wasserspeicher mit direkter Be- und Entladung für des Programmpaket Matlab zur Verfügung. Sowohl Simulationen am autonomen Speicher, als auch das Einbinden in ein bestehendes System sind auf sehr einfache Weise möglich.

Ein Verbesserungspotential ist durch Implementieren schnellerer bzw. weniger rechenintensiver Stoffwertfunktionen zu erkennen.

Außerdem wäre eine Erweiterung der Modelle denkbar. Die Energiezufuhr muss nicht notwendigerweise durch Be- bzw. Entladen des Speichermediums erfolgen. Ebenso kann Energie durch interne Wärmetauscher oder elektrische Heizstäbe transferiert werden.

Abbildungsverzeichnis

2.1	Einteilung thermischer Speicher	3
2.2	Latente und sensible Wärme	6
2.3	Einspeicher-, Zweispeichersystem	12
2.4	Warmwasserspeicher als Pufferspeicher mit direkter Be- und Entladung [Fis05]	13
2.5	Bewegung der Temperaturfront [Fis05]	14
2.6	Warmwasserspeicher mit indirekter Energiezufuhr über Wärmetauscher [Fis05]	14
2.7	Warmwasserspeicher mit idealer Einspeisung	15
2.8	Integration der Speicheranlage Simmering in das Fernwärmenetz	16
2.9	Aquifer-, Erdsonden- und Erdbeckenspeicher	17
3.1	Abbau der Schichtung durch Wärmeleitung des Fluids im adiabaten Speicher	23
3.2	Abbau der Schichtung durch Wärmeverluste an die Umgebung	24
3.3	Qualitatives Geschwindigkeitsprofil im kalten Speicherquerschnitt [New95] . .	26
3.4	Ermittlung der effektiven Wärmeleitfähigkeit aus experimentellen Daten . . .	27
3.5	Näherung der effektiven Wärmeleitfähigkeit	27
3.6	Kalorisches Mischen	28
3.7	Verwirbelungsfaktor $\epsilon = \epsilon(z, n)$	33
3.8	Einlassgeometrie [Zur91]	34
3.9	Verwirbelungsfaktor $\epsilon_{in} = \epsilon_{in}(Re/Ri, \text{Geometrie})$ [Zur91]	35
3.10	Verwirbelungsfaktor $\epsilon_{in} = \epsilon_{in}(Ri, \text{Geometrie})$ [Zur91]	35
3.11	Abkühlkurve des vollständig durchmischten Speichers	39
3.12	Modellierung des idealen Beladevorgangs in einem Schichtspeicher [Yoo96] . .	39
4.1	Zerlegung des Rechengebiets	46
4.2	Temperaturverlauf während des Zeitschritts	50
4.3	Einfluss der Integrationsmethode auf die numerische Diffusion	53
4.4	Einfluss der Knotenanzahl n auf die numerische Diffusion	53
4.5	Gegenüberstellung der Integrationsmethoden	54
5.1	Globale Energiebilanz	58

6.1	Zeitachse	65
6.2	Be- und Entlademassenstrom	67
6.3	Ein- und Austrittstemperaturen	67
6.4	Temperaturverlauf beim ersten Beladevorgang	68
6.5	Temperaturverlauf beim ersten Entladevorgang	68
6.6	Temperaturverlauf beim zweiten Beladevorgang	69
6.7	Temperaturverlauf im Stand-By Betrieb	69
6.8	Temperaturverlauf beim vollständigen Entladen	70
6.9	Enthalpieänderung	71
6.10	Exergieänderung	72
6.11	Entropieänderung	72
6.12	Energetischer Wirkungsgrad	73
6.13	Exergetischer Wirkungsgrad	74

Tabellenverzeichnis

2.1	Gegenüberstellung sensibler Speichermaterialien	5
2.2	Gegenüberstellung latenter Speichermaterialien	6
2.3	Reaktionspaare für die chemische Speicherung thermischer Energie [Fis05] . .	8
6.1	Rechenzeit	70

Literaturverzeichnis

- [AN97] AL-NAJEM, N.M. ; EL-REFAEE, M.M.: A Numerical Study for the Prediction of Turbulent Mixing Factor in Thermal Storage Tanks. In: *Applied Thermal Engineering* 17 (1997), Nr. 12, S. 1173–1181
- [Bec06] BECKER, R.: *Optimierung thermischer Systeme in dezentralen Energieversorgungsanlagen*, Universität Dortmund, Diss., 2006
- [Deu12] DEUTSCHES INSTITUT FÜR NORMUNG: *DIN EN 12977-3 Thermische Solaranlagen und ihre Bauteile - Kundenspezifisch gefertigte Anlagen - Teil 3: Leistungsprüfung von Warmwasserspeichern für Solaranlagen*. Juni 2012
- [Din11] DINÇER, İ. ; ROSEN, M.A.: *Thermal Energy Storage : Systems and Applications*. Wiley, 2011
- [Dru97] DRUECK, H. ; PETER, M. ; HAHNE, E.: Leistungsprüfung von Solaranlage zur Brauchwassererwärmung nach den zukünftigen CEN-Normen des TC 312. In: 7. *Symposium thermische Solarenergie*, 1997
- [Dru98] DRUECK, H. ; HAHNE, E.: Kombispeicher auf dem Prüfstand. In: 8. *Symposium thermische Solarenergie*, 1998
- [Dru06] DRUECK, H.: *Multiport Store-Model for TRNSYS - Type 340*. Version 1.99F. Stuttgart, Germany, März 2006
- [Fer08] FERZIGER, J.H. ; PERIC, M.: *Numerische Strömungsmechanik*. Springer London, Limited, 2008
- [Fis05] FISCH, N.: *Wärmespeicher*. 5. Auflage. Köln : TÜV-Verl., 2005 (BINE-Informationspaket)
- [Haf99] HAFNER, B. ; PLETTNER, J. ; WEMHÖNER, C. ; WENZEL, T. ; SOLAR-INSTITUT JÜLICH (Hrsg.): *CARNOT Blockset User's Guide*. Version 1.0. Jülich, Germany: Solar-Institut Jülich, November 1999

- [Hah98] HAHNE, E. ; CHEN, Y.: Numerical Study of Flow and Heat Transfer Characteristics in Hot Water Stores. In: *Solar Energy* 64 (1998), Nr. 3, S. 9–18
- [Hal10] HALLER, M.Y. ; YAZDANSHENAS, E. ; ANDERSEN, E. ; BALES, C. ; STREICHER, W. ; FURBO, S.: A Method to Determine Stratification Efficiency of Thermal Energy Storage Processes Independently from Storage Heat Losses. In: *Solar Energy* 84 (2010), S. 997–1007
- [Hau13] HAUER, A. ; HIEBLER, S. ; REUSS, M.: *Wärmespeicher*. 5. Auflage. Stuttgart : Fraunhofer IRB Verl., 2013 (BINE-Fachbuch)
- [Inc07] INCROPERA, F.P.: *Fundamentals of Heat and Mass Transfer*. 6. Auflage. John Wiley, 2007
- [Joh02] JOHN, R.: *Validierung eines Rechenmodells für Großspeicher zur Auslegung solarer Heizsysteme*, Philipps-Universität Marburg, Diss., 2002
- [Kle90] KLEINBACH, E.M.: *Performance Study of One-Dimensional Models for Stratified Thermal Storage Tank*, University of Wisconsin-Madison, Diplomarbeit, 1990
- [Kle06] KLEIN, S.A. ; BECKMAN, W.A. ; DUFFIE, J.A. ; SOLAR ENERGY LABORATORY, UNIVERSITY OF WISCONSIN-MADISON (Hrsg.): *TRNSYS - A Transient System Simulation Program*. Version 16. WI 53706 - USA: Solar Energy Laboratory, University of Wisconsin-Madison, 2006
- [Kuh10] KUHLMANN, H.C.: *Skriptum zur Vorlesung: Grundlagen der Numerischen Methoden der Strömungs- und Wärmetechnik*. Institut für Strömungslehre und Wärmeübertragung - Technische Universität Wien, 2010
- [Mat13a] THE MATHWORKS, INC. (Hrsg.): *Matlab Primer*. Version 8.2 (R2013b). Natick, USA: The MathWorks, Inc., September 2013
- [Mat13b] THE MATHWORKS, INC. (Hrsg.): *Simulink User's Guide*. Version 8.2 (R2013b). Natick, USA: The MathWorks, Inc., September 2013
- [New95] NEWTON, B.J.: *Modeling of Solar Storage Tanks*, University of Wisconsin-Madison, Diplomarbeit, 1995
- [Oer12] OERTEL, H.: *Prandtl - Führer durch die Strömungslehre*. 13. Auflage. Springer Vieweg, 2012
- [Opp86] OPPEL, F.J. ; GHAJAR, A.J. ; MORETTI, P.M.: A Numerical and Experimental Study of Stratified Thermal Storage. In: *ASHRAE Transactions* 92 (1986), S. 293–309

- [Pat80] PATANKAR, S.V.: *Numerical Heat Transfer and Fluid Flow*. Taylor & Francis, 1980 (Series in Computational Methods in Mechanics and Thermal Sciences)
- [Pow13] POWELL, K.M. ; EDGAR, T.F.: An Adaptive-Grid Model for Dynamic Simulation of Thermocline Thermal Energy Storage Systems. In: *Energy Conversion and Management* 76 (2013), S. 865–873
- [VDI10] VDI-GESELLSCHAFT VERFAHRENSTECHNIK UND CHEMIEINGENIEURWESEN, VDI: *VDI-Wärmeatlas*. 11. Auflage. Springer, 2010 (VDI-Buch)
- [Wag98] WAGNER, W. ; KRUSE, A.: *Properties of Water and Steam: The Industrial Standard IAPWS-IF97 for the Thermodynamic Properties and Supplementary Equations for Other Properties : Tables based on these Equation*. Springer-Verlag, 1998
- [Wie] WIEN ENERGIE: *Wien Energie eröffnet 1. Hochdruck-Wärmespeicher*. <http://www.wienenergie.at/eportal/ep/bvContentView.do?contentTypeId=1001&contentId=32323&programId=59635&channelId=-44429>, Abruf: 4. Aug. 2014. Online
- [Yoo96] YOO, H. ; PAK, E.: Analytical Solution to a One-Dimensional Finit-Domain Model for Stratified Thermal Storage Tanks. In: *Solar Energy* 56 (1996), Nr. 4, S. 315–322
- [Zur91] ZURIGAT, Y.H. ; LICHE, P.R. ; GHAJAR, A.J.: Influence of Inlet Geometry on Mixing in Thermocline Thermal Energy Storage. In: *International Journal of Heat and Mass Transfer* 34 (1991), Nr. 1, S. 115–125

Anhang A

Programmcode

```
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %%Falkner Stefan, 0826830                                     %%
3 %%Wien, 13.5.2014                                           %%
4 %%STEP_FILO                                                 %%
5 %%Zeitschritt mit dem Modell FILO (First In Last Out)      %%
6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7 %%Falkner Stefan, 22.5.2014                                  %%
8 %%zusaetzlicher In- u. Outputs fuer Austrittstemp.         %%
9 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10 %%Falkner Stefan, 17.6.20014                                 %%
11 %%Warning: dt zu gross                                     %%
12 %%Error: m_up und m_down ungleich 0                         %%
13 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14 %%                                     Input                 %%
15 %%H=Storage(1)...Speicherhoehe in m (Dim: 1x1)             %%
16 %%Dia=Storage(2)...Speicherdurchmesser in m (Dim: 1x1)     %%
17 %%k_rad=Storage(3)...radialer Waermedurchgang in W/m^2K    %%
18 %%k_ax=Storage(4)...axialer Waermedurchgang in W/m^2K     %%
19 %%lambda_eff=Storage(5)...effektive axiale Waermeleitfaehigkeit in W/mK %%
20 %% (Dim: 1x1)                                              %%
21 %%n=Geometry_node(1)...Anzahl der Knoten (Dim: 1x1)       %%
22 %%V_node=Geometry_node(2)...Knotenvolumen in m^3 (Dim: 1x1) %%
23 %%dz=Geometry_node(3)...Knotenhoehe dz in m (Dim: 1x1)   %%
24 %%A_ax=Geometry_node(4)...axialer Querschnitt in m^2 (Dim: 1x1) %%
25 %%A_rad=Geometry_node(5)...Mantelflaeche der Knoten in m^2 (Dim: 1x1) %%
26 %%d_in=Inlet(1)...Einstroemdurchmesser in m (Dim: 1x1)    %%
27 %%a=Inlet(2)...epsilon_in=a*(Re/Ri)^b (Dim: 1x1)          %%
28 %%b=Inlet(3) (Dim: 1x1)                                    %%
29 %%phi=Inlet(4)...Exponent fuer epsilon_eff;                %%
30 %%phi=0 --> hypberbolischer Verlauf von epsilon_eff (Dim: 1x1) %%
31 %%Inlet(5)==1...Beruecksichtigung der Einstroemverhaeltnisse (Dim: 1x1) %%
```

```

32 %%Inlet(5)==else...keine Beruecksichtigung der Einstroemverhaeltnisse %%
33 %%(Dim: 1x1) %%
34 %%m_up...Entlademassenstrom in kg/s (Dim: 1x1) %%
35 %%m_down...Belademassenstrom in kg/s --> einer der beiden bzw. alle %%
36 %%beide muessen 0 sein (Dim: 1x1) %%
37 %%T_up...Temp. des Entlademassenstroms in Grad C (Dim: 1x1) %%
38 %%T_down...Temp. des Belademassenstroms in Grad C (Dim: 1x1) %%
39 %%T_amb...Temp. der Umgebung in Grad C (Dim: 1x1) %%
40 %%T0...Speichertemp. zu Beginn des Zeitschritts in Grad C (Dim: nx1) %%
41 %%T0_out_1..Austrittstemp.unten vorangehender Zeitschr.in GradC(Dim:1x1)%%
42 %%T0_out_n...Austrittstemp.oben vorangehender Zeitschr.in GradC(Dim:1x1)%%
43 %%dt...vom Benutzer vorgegebene Zeitschrittlaenge in s (Dim: 1x1) %%
44 %%f_crit...Faktor fuer max. Zeitschrittlaenge dt_max=f_crit*dt_crit %%
45 %%(Dim: 1x1) %%
46 %%%                               Output %%
47 %%T1...Speichertemp. am Ende des Zeitschritts in Grad C (Dim: nx1) %%
48 %%T1_out_1...Austrittstemp.unten waehrend des Schritts inGrad C(Dim:1x1)%%
49 %%T1_out_n...Austrittstemp.oben waehrend des Schritts inGrad C(Dim: 1x1)%%
50 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
51
52
53 function [ T1,T1_out_1,T1_out_n ] = STEP_FILO(Storage,Geometry_node, ...
        Inlet,m_up,m_down,T_up,T_down,T_amb,T0,T0_out_1,T0_out_n,dt,f_crit)
54
55 %warning
56 if (m_up≠0 && m_down≠0)
57     error('m_up und/oder m_down muessen/muss gleich 0 sein');
58 end
59
60
61 %Zeitschritthandling
62 %kritischer Zeitschritt: Courant-Zahl=1=v*dt/dz
63 if m_up≠0
64     dt_crit=density(T_up)*Geometry_node(4)*Geometry_node(3)/m_up;
65     %Austrittstemp. unten bleibt gleich
66     T1_out_1=T0_out_1;
67 elseif m_down≠0
68     dt_crit=density(T_down)*Geometry_node(4)*Geometry_node(3)/m_down;
69     %Austrittstemp. oben bleibt gleich
70     T1_out_n=T0_out_n;
71 else
72     dt_crit=10^10; %-->infinity
73     %Austrittstemp. oben u. unten bleiben gleich
74     T1_out_1=T0_out_1;
75     T1_out_n=T0_out_n;
76 end
77 dt_max=dt_crit*f_crit;

```

```

78 %sollte der vom Benutzer vorgegebene Zeitschritt dt groesser als der
79 %maximal moegliche dt_max sein, so muessen funktionsintern mehrere
80 %Zeitschritte mit einer kleineren Zeitschrittdauer geloest werden, bis die
81 %gewuenscht Zeitschrittlaege dt erreicht ist.
82 if dt>ceil(dt_max)
83     nt=ceil(dt/dt_max); %Anzahl der noetigen Zeitschritte (aufgerundet)
84     dt = dt/nt;       %neue interne Zeitschrittlaege
85 else
86     nt=1;
87 end
88
89 %warning
90 if nt≠1
91     warning(['Zeitschrittweite zu gross! Es werden ',num2str(nt),' ...
92             Schritte mit einer Zeitschrittweite von ',num2str(dt),' s ...
93             berechnet.']);
94 end
95
96 %interne Zeitschritte
97 for j=1:nt
98     if Inlet(5)==1
99         %Einstroemverhaeltnisse werden beruecksichtigt
100        if m_up≠0
101            v_in = 4*m_up/(Inlet(1)^2*pi()*density(T_up));
102            nu = kinematic_viscosity(T_up);
103            Ri = 9.81*Storage(1)*0.5*(thermal_expansion(T_up) ...
104                +thermal_expansion(T0(1)))*abs(T_up-T0(1))/v_in^2;
105            Re = v_in*Inlet(1)/nu;
106        elseif m_down≠0
107            v_in = 4*m_down/(Inlet(1)^2*pi()*density(T_down));
108            nu = kinematic_viscosity(T_down);
109            Ri = 9.81*Storage(1)*0.5*(thermal_expansion(T_down) ...
110                +thermal_expansion(T0(Geometry_node(1)))) ...
111                *abs(T_down-T0(Geometry_node(1)))/v_in^2;
112            Re = v_in*Inlet(1)/nu;
113        else
114            Ri = 10^10; %Ri-->infinity
115            Re = 0;
116        end
117        epsilon_eff=EPSILON(Inlet(2)*(Re/Ri)^Inlet(3),Geometry_node(1), ...
118            m_up,m_down,Geometry_node(3),Inlet(4));
119    else
120        %Einstroemverhaeltnisse werden nicht beruecksichtigt
121        epsilon_eff=EPSILON(0,Geometry_node(1),0,0,Geometry_node(3),0);
122    end
123    %Koeffizienten der Energiebilanz

```

```

118     koeff = FILO( Geometry_node,Storage(3),Storage(4),Storage(5),m_up, ...
           m_down,T_up,T_down,T_amb,T0,epsilon_eff );
119     %Zeitschritt
120     T0 = CRANK_NICOLSON(koeff,T0,dt);
121     %Abbau inverser Schichtung
122     T0 = CALORIC_MIXING(T0);
123 end
124 T1=T0;
125
126 %Austrittstemp.falls Massenstroeme vorhanden (konsistent zu Crank-Nicolson)
127 if m_up≠0
128     T1_out_n=0.5*(T0(Geometry_node(1))+T1(Geometry_node(1)));
129 elseif m_down≠0
130     T1_out_l=0.5*(T0(1)+T1(1));
131 end
132 end
    
```

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %%Falkner Stefan, 0826830                                     %%
3  %%Wien, 13.5.2014                                           %%
4  %%STEP_VAR_IN                                              %%
5  %%Zeitschritt mit dem Modell VAR_IN (VARIABLE INlet)      %%
6  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7  %%Falkner Stefan, 22.5.2014                                  %%
8  %%zusaeztlicher In- u. Outputs fuer Austrittstemp.        %%
9  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10 %%Falkner Stefan, 17.6.20014                                 %%
11 %%Warning: dt zu gross                                     %%
12 %%Error: m_up und m_down ungleich 0                         %%
13 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14 %% Input                                                    %%
15 %%H=Storage(1)...Speicherhoehe in m (Dim: 1x1)             %%
16 %%Dia=Storage(2)...Speicherdurchmesser in m (Dim: 1x1)     %%
17 %%k_rad=Storage(3)...radialer Waermedurchgang in W/m^2K (Dim: 1x1) %%
18 %%k_ax=Storage(4)...axialer Waermedurchgang in W/m^2K (Dim: 1x1) %%
19 %%lambda_eff=Storage(5)...effektive axiale Waermeleitfaehigkeit in W/mK %%
20 %% (Dim: 1x1)                                               %%
21 %%n=Geometry_node(1)...Anzahl der Knoten (Dim: 1x1)       %%
22 %%V_node=Geometry_node(2)...Knotenvolumen in m^3 (Dim: 1x1) %%
23 %%dz=Geometry_node(3)...Knotenhoehe dz in m (Dim: 1x1)   %%
24 %%A_ax=Geometry_node(4)...axialer Querschnitt in m^2 (Dim: 1x1) %%
25 %%A_rad=Geometry_node(5)...Mantelflaeche der Knoten in m^2 (Dim: 1x1) %%
26 %%m_up...Entlademassenstrom in kg/s (Dim: 1x1)            %%
27 %%m_down...Belademassenstrom in kg/s --> einer der beiden bzw. alle %%
28 %%beide muessen 0 sein (Dim: 1x1)                          %%
29 %%T_up...Temp. des Entlademassenstroms in Grad C (Dim: 1x1) %%
    
```

```

30 %%T_down...Temp. des Belademassenstroms in Grad C (Dim: 1x1)      %%
31 %%T_amb...Temp. der Umgebung in Grad C (Dim: 1x1)                %%
32 %%T0...Speichertemp. zu Beginn des Zeitschritts in Grad C (Dim: nx1) %%
33 %%T0_out_1...Austrittstemp.unten vorangehender Zeitschr.in GradC(Dim:1x1)%%
34 %%T0_out_n...Austrittstemp.oben vorangehender Zeitschr.in GradC(Dim:1x1)%%
35 %%dt...vom Benutzer vorgegebene Zeitschrittlaenge in s (Dim: 1x1)  %%
36 %%f_crit...Faktor fuer max. Zeitschrittlaenge dt_max=f_crit*dt_crit  %%
37 %% (Dim: 1x1)                                                       %%
38 %%%                                                                    %%
39 %%T1...Speichertemp. am Ende des Zeitschritts in Grad C (Dim: nx1)  %%
40 %%T1_out_1...Austrittstemp.unten waehrend des Schritts inGrad C(Dim:1x1)%%
41 %%T1_out_n...Austrittstemp.oben waehrend des Schritts inGrad C(Dim: 1x1)%%
42 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
43
44
45 function [ T1,T1_out_1,T1_out_n ] = STEP_VAR_IN(Storage,Geometry_node, ...
         m_up,m_down,T_up,T_down,T_amb,T0,T0_out_1,T0_out_n,dt,f_crit)
46
47 %warning
48 if (m_up≠0 && m_down≠0)
49     error('m_up und/oder m_down muessen/muss gleich 0 sein');
50 end
51
52 %Zeitschritthandling
53 %kritischer Zeitschritt: Courant-Zahl=1=v*dt/dz
54 if m_up≠0
55     dt_crit=density(T_up)*Geometry_node(4)*Geometry_node(3)/m_up;
56     %Austrittstemp. unten bleibt gleich
57     T1_out_1=T0_out_1;
58 elseif m_down≠0
59     dt_crit=density(T_down)*Geometry_node(4)*Geometry_node(3)/m_down;
60     %Austrittstemp. oben bleibt gleich
61     T1_out_n=T0_out_n;
62 else
63     dt_crit=10^10; %-->infinity
64     %Austrittstemp. oben u. unten bleiben gleich
65     T1_out_1=T0_out_1;
66     T1_out_n=T0_out_n;
67 end
68 dt_max=dt_crit*f_crit;
69 %sollte der vom Benutzer vorgegebene Zeitschritt dt groesser als der
70 %maximal moegliche dt_max sein, so muessen funktionsintern mehrere
71 %Zeitschritte mit einer kleineren Zeitschrittdauer geloest werden, bis die
72 %gewuenscht Zeitschrittlaenge dt erreicht ist.
73 if dt>ceil(dt_max)
74     nt=ceil(dt/dt_max); %Anzahl der noetigen Zeitschritte (aufgerundet)
75     dt = dt/nt;        %neue interne Zeitschrittlaenge

```

```

76 else
77     nt=1;
78 end
79
80 %warning
81 if nt≠1
82     warning(['Zeitschrittweite zu gross! Es werden ',num2str(nt),' ...
            Schritte mit einer Zeitschrittweite von ',num2str(dt),' s ...
            berechnet.']);
83 end
84
85 %interne Zeitschritte
86 for j=1:nt
87     %Suche Schicht mit geringster Temp.abweichung zu T_down
88     i_in = GET_I_IN(T0,T_down);
89     %Koeffizienten der Energiebilanz
90     koeff = VAR_IN(Geometry_node,Storage(3),Storage(4),Storage(5), ...
            m_up,m_down,T_up,T_down,T_amb,T0,i_in);
91     %Zeitschritt
92     T0 = CRANK_NICOLSON(koeff,T0,dt);
93     %Abbau inverser Schichtung
94     T0 = CALORIC_MIXING(T0);
95 end
96 T1=T0;
97
98 %verbleibende Austrittstemp. falls Massenstroeme vorhanden
99 if m_up≠0
100     T1_out_n=0.5*(T0(Geometry_node(1))+T1(Geometry_node(1)));
101 elseif m_down≠0
102     T1_out_1=0.5*(T0(1)+T1(1));
103 end
104 end

```

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %%Falkner Stefan, 0826830                                             %%
3  %%Wien, 14.3.2014                                                  %%
4  %%FILO (First In Last Out)                                         %%
5  %%Funktion berechnet die Koeffizienten fuer ein DGL-System der Art %%
6  %%a(i,1)*(dT/dt)(i)=a(i,2)*T(i)+a(i,3)*T(i+1)+a(i,4)*T(i-1)+a(i,5) %%
7  %%fuer i = 1...n                                                  %%
8  %%Einschraenkungen:                                               %%
9  %%Beladen erfolgt am hoechsten Knoten n                           %%
10 %%Entladen erfolgt am tiefsten Knoten 1                            %%
11 %%konstante Knotengroesse                                          %%
12  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
13 %%Falkner Stefan, 16.04.2014                                       %%

```

```

14 %%Implementieren von epsilon_eff zum Erfassen von Turbulenzen, %%
15 %%aufgrund der Einstroemung-->Rueckfuehrung auf Waermeleitungsproblem %%
16 %%Falkner Stefan, 11.05.2014 %%
17 %%(1) Implementieren der Temperaturabhaengigkeit von cp u. rho. %%
18 %%(2) Anpassen der Inputs %%
19 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
20 %% Input %%
21 %%Geometry...Knotenabmessungen(Anzahl der Knoten n, Knotenvolumen V_node%%
22 %%in m^3, Knotenhoehe dz in m, axialer Querschnitt A_ax in m^2, %%
23 %%Mantelflaeche A_rad in m^2) %%
24 %%k_rad...radialer Waermedurchgang in W/m^2K (Dim: 1 x 1) %%
25 %%k_ax...axialer Waermedurchgang in W/m^2K (Dim: 1 x 1) %%
26 %%lambda_eff...axiale effektive Waermeleitfaehigkeit in W/mK (Dim: 1x1) %%
27 %%m_up...Entlademassenstrom in kg/s (positiv) (Dim: 1 x 1) %%
28 %%m_down...Belademassenstrom in kg/s (positiv) (Dim: 1 x 1) %%
29 %%T_up...Temp. des Entlademassenstroms in Grad C (Dim: 1 x 1) %%
30 %%T_down...Temp. des Belademassenstroms in Grad C (Dim: 1 x 1) %%
31 %%T_amb...Umgebungstemperatur in Grad C (Dim: 1 x 1) %%
32 %%T0...Temp. zu Beginn des Zeitschritts in Grad C (Dim: n x 1) %%
33 %%epsilon_eff...Faktor fuer Waermeleitung unter gegebenen Bedingungen %%
34 %%in - (Dim: (n-1)x1) %%
35 %% Output %%
36 %%a=[a1,a2,a3,a4,a5]...Spaltenvektoren der Koeffizienten (Dim: n x 5) %%
37 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
38
39
40 function [ a ] = FILO( Geometry,k_rad,k_ax,lambda_eff,m_up,m_down, ...
    T_up,T_down,T_amb,T0,epsilon_eff )
41
42 n=Geometry(1);
43 V_node=Geometry(2);
44 dz=Geometry(3);
45 A_ax=Geometry(4);
46 A_rad=Geometry(5);
47
48 a = zeros(n,5);
49
50
51 for i=1:n
52     %density(T in Grad C)
53     rho = density(T0(i));
54     cp = heat_capacity_int(T0(i));
55     a(i,1) = rho*cp;
56     if i==1
57         a(1,2) = -(A_rad*k_rad+A_ax*k_ax)/V_node ...
            -lambda_eff*epsilon_eff(1)/dz^2-m_up*cp/V_node-m_down*cp/V_node;
58         a(1,3) = lambda_eff*epsilon_eff(1)/dz^2+m_down*cp/V_node;

```

```

59     a(1,4) = 0;
60     a(1,5) = m_up*cp/V_node*T_up+(A_rad*k_rad+A_ax*k_ax)/V_node*T_amb;
61     elseif i==n
62         a(n,2) = -(A_rad*k_rad+A_ax*k_ax)/V_node ...
                -lambda_eff*epsilon_eff(n-1)/dz^2-m_up*cp/V_node-m_down*cp/V_node;
63         a(n,3) =0;
64         a(n,4) = lambda_eff*epsilon_eff(n-1)/dz^2+m_up*cp/V_node;
65         a(n,5) = m_down*cp/V_node*T_down+(A_rad*k_rad+A_ax*k_ax)/V_node*T_amb;
66     else
67         a(i,2) = -A_rad*k_rad/V_node-lambda_eff*(epsilon_eff(i) ...
                +epsilon_eff(i-1))/dz^2-m_up*cp/V_node-m_down*cp/V_node;
68         a(i,3) = lambda_eff*epsilon_eff(i)/dz^2+m_down*cp/V_node;
69         a(i,4) = lambda_eff*epsilon_eff(i-1)/dz^2+m_up*cp/V_node;
70         a(i,5) = A_rad*k_rad/V_node*T_amb;
71     end
72 end

```

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %%Falkner Stefan, 0826830                                     %%
3  %%Wien, 15.3.2014                                           %%
4  %%VAR_IN (Variable Inlet)                                   %%
5  %%Funktion berechnet die Koeffizienten fuer ein DGL-System der Art   %%
6  %%a(i,1)*(dT/dt)(i)=a(i,2)*T(i)+a(i,3)*T(i+1)+a(i,4)*T(i-1)+a(i,5)   %%
7  %%(i = 1...n)                                               %%
8  %%Einschraenkungen:                                         %%
9  %%Beladen erfolgt im Knoten i_in                             %%
10 %%Entladen erfolgt am tiefstem Knoten 1                     %%
11 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12 %%15.03 - Erweiterung von FILO                               %%
13 %%der Beladeknoten ist nun nicht mehr an den oberen Knoten gebunden, %%
14 %%sondern kann variiert werden (Knoten i_in). Uebergibt man fuer i_in=n,%%
15 %%so bekommt man die gleichen Ergebnisse wie mit der Fkt. FILO   %%
16 %%Falkner Stefan, 11.05.2014                                 %%
17 %%(1) Implementieren der Temperaturabhaengigkeit von cp u. rho.   %%
18 %%(2) Anpassen der Inputs                                     %%
19 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
20 %%                               Input                         %%
21 %%Geometry...Knotenabmessungen(Anzahl der Knoten n, Knotenvolumen V_node%%
22 %%in m^3, Knotenhoehe dz in m, axialer Querschnitt A_ax in m^2,   %%
23 %%Mantelflaeche A_rad in m^2)                                   %%
24 %%k_rad...radialer Waermedurchgang in W/m^2K (Dim: 1 x 1)      %%
25 %%k_ax...axialer Waermedurchgang in W/m^2K (Dim: 1 x 1)      %%
26 %%lambda_eff...effektive axiale Waermeleitfaehigkeit in W/mK(Dim: 1 x 1)%%
27 %%V_node...Knotenvolumen in m^3 (Dim: 1 x 1)                  %%
28 %%A_rad...Flaechen fuer radiale Verluste in m^2 (Dim: 1 x 1)  %%
29 %%A_ax...Flaechen fuer axiale Waermeleitung bzw. Verluste in m^2(Dim:1x1)%%

```

```

30 %%dz...Knotenhoehe in m (Dim: 1 x 1) %%
31 %%n...Knotenanzahl (Dim: 1 x 1) %%
32 %%m_up...Entlademassenstrom in kg/s (positiv) (Dim: 1 x 1) %%
33 %%m_down...Belademassenstrom in kg/s (positiv) (Dim: 1 x 1) %%
34 %%T_up...Temp. des Entlademassenstroms in Grad C (Dim: 1 x 1) %%
35 %%T_down...Temp. des Belademassenstroms in Grad C (Dim: 1 x 1) %%
36 %%T_amb...Umgebungstemperatur in Grad C (Dim: 1 x 1) %%
37 %%T0...Temp. zu Beginn des Zeitschritts in Grad C (Dim: n x 1) %%
38 %%i_in...Knotennummer der Beladeschicht %%
39 %%% Output %%
40 %%a=[a1,a2,a3,a4,a5]...Spaltenvektoren der Koeffizienten (Dim: n x 5) %%
41 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
42
43
44 function [ a ] = VAR_IN( ...
    Geometry,k_rad,k_ax,lambda_eff,m_up,m_down,T_up,T_down,T_amb,T0,i_in )
45
46 n=Geometry(1);
47 V_node=Geometry(2);
48 dz=Geometry(3);
49 A_ax=Geometry(4);
50 A_rad=Geometry(5);
51
52 a = zeros(n,5);
53
54 %Bestimmen der Koeffizient fuer Energiebilanz
55 if (m_up~=0) || ((i_in==n) && (m_down~=0)) || ((m_up==0) && (m_down==0))
56     %Entladevorgang
57     %Beladen im hoechsten Knoten i_in=n
58     %keine Massenstroeme
59     for i=2:n-1
60         rho = density(T0(i));
61         cp = heat_capacity_int(T0(i));
62         a(i,1) = rho*cp;
63         a(i,2) = -A_rad*k_rad/V_node-2*lambda_eff/dz^2-m_up*cp/V_node ...
            -m_down*cp/V_node;
64         a(i,3) = lambda_eff/dz^2+m_down*cp/V_node;
65         a(i,4) = lambda_eff/dz^2+m_up*cp/V_node;
66         a(i,5) = A_rad*k_rad/V_node*T_amb;
67     end
68     rho = density(T0(1));
69     cp = heat_capacity_int(T0(1));
70     a(1,1) = rho*cp;
71     a(1,2) = ...
        -(A_rad*k_rad+A_ax*k_ax)/V_node-lambda_eff/dz^2-m_up*cp/V_node ...
        -m_down*cp/V_node;
72     a(1,3) = lambda_eff/dz^2+m_down*cp/V_node;

```

```

73     a(1,4) = 0;
74     a(1,5) = m_up*cp/V_node*T_up+(A_rad*k_rad+A_ax*k_ax)/V_node*T_amb;
75     rho = density(T0(n));
76     cp = heat_capacity_int(T0(n));
77     a(n,1) = rho*cp;
78     a(n,2) = a(1,2);
79     a(n,3) =0;
80     a(n,4) = lambda_eff/dz^2+m_up*cp/V_node;
81     a(n,5) = m_down*cp/V_node*T_down+(A_rad*k_rad+A_ax*k_ax)/V_node*T_amb;
82 elseif (i_in==1)&&(m_down≠0)
83     %Beladen im tiefsten Knoten i_in=1
84     for i=2:n-1
85         rho = density(T0(i));
86         cp = heat_capacity_int(T0(i));
87         a(i,1) = rho*cp;
88         a(i,2) = -A_rad*k_rad/V_node-2*lambda_eff/dz^2;
89         a(i,3) = lambda_eff/dz^2;
90         a(i,4) = lambda_eff/dz^2;
91         a(i,5) = A_rad*k_rad/V_node*T_amb;
92     end
93     rho = density(T0(n));
94     cp = heat_capacity_int(T0(n));
95     a(n,1) = rho*cp;
96     a(n,2) = -(A_rad*k_rad+A_ax*k_ax)/V_node-lambda_eff/dz^2;
97     a(n,3) =0;
98     a(n,4) = lambda_eff/dz^2;
99     a(n,5) = (A_rad*k_rad+A_ax*k_ax)/V_node*T_amb;
100    rho = density(T0(1));
101    cp = heat_capacity_int(T0(1));
102    a(1,1) = rho*cp;
103    a(1,2) = -(A_rad*k_rad+A_ax*k_ax)/V_node-lambda_eff/dz^2-m_down*cp/V_node;
104    a(1,3) = lambda_eff/dz^2;
105    a(1,4) = 0;
106    a(1,5) = (A_rad*k_rad+A_ax*k_ax)/V_node*T_amb+m_down*cp/V_node*T_down;
107 else
108     %Beladen in einem der inneren Knoten i_in=(2 bis n-1)
109     for i=1:n
110         rho = density(T0(i));
111         cp = heat_capacity_int(T0(i));
112         if i==1
113             a(i,1) = rho*cp;
114             a(i,2) = -(A_rad*k_rad+A_ax*k_ax)/V_node-lambda_eff/dz^2 ...
                    -m_down*cp/V_node;
115             a(i,3) = lambda_eff/dz^2+m_down*cp/V_node;
116             a(i,4) = 0;
117             a(i,5) = (A_rad*k_rad+A_ax*k_ax)/V_node*T_amb;
118         elseif i==n

```

```

119         a(i,1) = rho*cp;
120         a(i,2) = -(A_rad*k_rad+A_ax*k_ax)/V_node-lambda_eff/dz^2;
121         a(i,3) =0;
122         a(i,4) = lambda_eff/dz^2;
123         a(i,5) = (A_rad*k_rad+A_ax*k_ax)/V_node*T_amb;
124     elseif i==i_in
125         a(i_in,1) = rho*cp;
126         a(i_in,2) = ...
127             -A_rad*k_rad/V_node-2*lambda_eff/dz^2-m_down*cp/V_node;
128         a(i_in,3) = lambda_eff/dz^2;
129         a(i_in,4) = lambda_eff/dz^2;
130         a(i_in,5) = A_rad*k_rad/V_node*T_amb+m_down*cp/V_node*T_down;
131     elseif i<i_in&&i≠1
132         a(i,1) = rho*cp;
133         a(i,2) = -A_rad*k_rad/V_node-2*lambda_eff/dz^2-m_down*cp/V_node;
134         a(i,3) = lambda_eff/dz^2+m_down*cp/V_node;
135         a(i,4) = lambda_eff/dz^2;
136         a(i,5) = A_rad*k_rad/V_node*T_amb;
137     elseif i>i_in&&i≠n
138         a(i,1) = rho*cp;
139         a(i,2) = -A_rad*k_rad/V_node-2*lambda_eff/dz^2;
140         a(i,3) = lambda_eff/dz^2;
141         a(i,4) = lambda_eff/dz^2;
142         a(i,5) = A_rad*k_rad/V_node*T_amb;
143     end
144 end
145 end

```

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %%Falkner Stefan, 0826830                                             %%
3  %%Wien, 13.5.2014                                                    %%
4  %%TESTING_L                                                          %%
5  %%Funktion simuliert Sequenz der Pruefung L aus DIN EN 12977-3:2012-06 %%
6  %%L1...Konditionierung bis ein stat. Zustand erreicht ist          %%
7  %%L2...Beladen bis T_ch_stop im unteren Bereich des Tanks erreicht wird %%
8  %%L3...Stand-By                                                    %%
9  %%L4...Entladen bis T_disch_stop im oberen Bereich erreicht wird   %%
10 %%Anmerkung: Uebergabe der Volumenstroeme, Temperaturen,... obliegen %%
11 %%dem Benutzer. Somit kann diese Fkt. auch Pruefung S u. Pruefung C %%
12 %%simulieren. Bzw. ist es moeglich andere aehnliche Szenarien     %%
13 %%nachzustellen                                                    %%
14 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
15 %%                               Input                                %%
16 %%H=Storage(1)...Speicherhoehe in m (Dim: 1x1)                      %%
17 %%Dia=Storage(2)...Speicherdurchmesser in m (Dim: 1x1)              %%

```

```

18 %%k_rad=Storage(3)...radialer Waermedurchgang in W/m^2K (Dim: 1x1)      %%
19 %%k_ax=Storage(4)...axialer Waermedurchgang in W/m^2K (Dim: 1x1)      %%
20 %%lambda_eff=Storage(5)...effektive axiale Waermeleitfaehigkeit in W/mK %%
21 %% (Dim: 1x1)                                                           %%
22 %%n...Anzahl der Knoten (Dim: 1x1)                                     %%
23 %%d_in=Inlet(1)...Einstroemdurchmesser in m (Dim: 1x1)                %%
24 %%a=Inlet(2)...epsilon_in=a*(Re/Ri)^b (Dim: 1x1)                     %%
25 %%b=Inlet(3) (Dim: 1x1)                                               %%
26 %%phi=Inlet(4)...Exponent fuer epsilon_eff;                           %%
27 %%phi=0 --> hypberbolischer Verlauf von epsilon_eff (Dim: 1x1)        %%
28 %%Inlet(5)==1...Beruecksichtigung der Einstroemverhaeltnisse (Dim: 1x1) %%
29 %%Inlet(5)==else...keine Beruecksichtigung der Einstroemverhaeltnisse %%
30 %% (Dim: 1x1)                                                           %%
31 %%dV_ch...Beladevolumenstrom in m^3/s (Dim: 1x1)                      %%
32 %%T_ch...Beladetemp. in Grad C (Dim: 1x1)                             %%
33 %%T_ch_stop...zu erreichende Austrittstemp. in Grad C (Dim: 1x1)      %%
34 %%t_stby...Dauer Stand by in s (Dim: 1x1)                             %%
35 %%dV_disch...Entladevolumenstrom in m^3/s (Dim: 1x1)                 %%
36 %%T_disch...Entladetemp. in Grad C (Dim: 1x1)                         %%
37 %%T_disch_stop...zu erreichende Austrittstemp. in Grad C (Dim: 1x1)   %%
38 %%T_amb...Temp. der Umgebung in Grad C (Dim: 1x1)                     %%
39 %%T_ini...Anfangstemp. in Grad C (Dim: 1x1)                           %%
40 %%f_crit...Faktor fuer max. Zeitschrittlaenge dt_max=f_crit*dt_crit  %%
41 %% (Dim: 1x1)                                                           %%
42 %%%                               Output                                %%
43 %%T...Speichertemperatur in Grad C (Dim: n x nt_ges+1)                 %%
44 %%z...z-Koordinaten der Knoten in m (Dim: n x 1)                       %%
45 %%time...Zeitvektor in s (Dim: 1 x nt_ges+1)                           %%
46 %%m_down...Belademassenstrom ueber Zeit in kg/s (Dim: 1 x nt_ges)     %%
47 %%m_up...Entlademassenstrom ueber Zeit in kg/s (Dim: 1 x nt_ges)      %%
48 %%T_down...Beladetemp. ueber Zeit in Grad C (Dim: 1 x nt_ges)         %%
49 %%T_up...Entladetemp. ueber Zeit in Grad C (Dim: 1 x nt_ges)          %%
50 %%T_out_1...Austrittstemp. unten in Grad C (Dim: 1 x nt_ges+1)         %%
51 %%T_out_n...Austrittstemp. oben in Grad C (Dim: 1 x nt_ges+1)         %%
52 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
53
54
55 function [ T,z,time,m_down,m_up,T_down,T_up,T_out_1,T_out_n ] = ...
    TESTING_L(Storage,n,Inlet,dV_ch,T_ch,T_ch_stop,t_stby,dV_disch, ...
    T_disch,T_disch_stop,T_amb,T_ini,f_crit)
56
57 Geometry_node = GEOMETRY(n,Storage(1),Storage(2)); ...
    %Knotengeometrie des Speichers
58 z = GET_Z(Geometry_node(3),Geometry_node(1)); %zugehoerige z-Koordinaten
59
60 %Anfangszustand-Knoditionierung
61 j=1;

```

```

62 T=zeros(Geometry_node(1),j);
63 T(:,j)=T_ini;
64 %Austrittstemp.
65 T_out_1=T(1,1);
66 T_out_n=T(Geometry_node(1),1);
67
68 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Beladevorgang
69 dt_crit_ch = Geometry_node(2)/dV_ch; %kritischer Zeitschritt in s beim ...
    Beladen
70 dt_ch = f_crit*dt_crit_ch; %Zeitschrittdauer in s beim Beladen
71 m_ch=dV_ch*density(T_ch); %Massenstrom beim Beladen in kg/s
72 m_disch=0; %muss somit 0 sein
73 %bis die Austrittstemp. (unten) T_ch_stop erreicht hat
74 while T(1,j)<T_ch_stop
75     [T(:,j+1),T_out_1(1,j+1),T_out_n(1,j+1)] = ...
        STEP_FILO(Storage,Geometry_node,Inlet,m_disch,m_ch, ...
            T_disch,T_ch,T_amb,T(:,j),T_out_1(1,j),T_out_n(1,j),dt_ch,f_crit);
76     j = j+1;
77 end
78 nt_ch=j-1; %Anzahl der Beladezeitschritte
79
80
81
82 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Stand-By
83 if t_stby≠0
84     dt_stby = 3600; %Zeitschritt im Stand-By, soll ca. eine Stunde betragen
85     nt_stby = ceil(t_stby/dt_stby); %aufrunden
86     dt_stby = t_stby/nt_stby; %Zeitschritt im Stand-By in s
87     m_ch = 0; %im StandBy gleich 0
88     m_disch = 0; %im StandBy gleich 0
89     for j=nt_ch+1:nt_ch+nt_stby
90         [T(:,j+1),T_out_1(1,j+1),T_out_n(1,j+1)] = ...
            STEP_FILO(Storage,Geometry_node,Inlet,m_disch,m_ch,T_disch, ...
                T_ch,T_amb,T(:,j),T_out_1(1,j),T_out_n(1,j),dt_stby,f_crit);
91     end
92 else
93     nt_stby=0;
94 end
95
96 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Entladevorgang
97 dt_crit_disch = Geometry_node(2)/dV_disch; %kritischer Zeitschritt in s ...
    beim Entladen
98 dt_disch = f_crit*dt_crit_disch; %Zeitschrittdauer in s beim ...
    Entladen
99 m_disch=dV_disch*density(T_disch); %Massenstrom beim Entladen in kg/s
100 m_ch=0; %muss somit 0 sein
101 %bis die Austrittstemp. (oben) annaehrnd T_disch_stop erreicht hat

```

```

102 j=nt_ch+nt_stby+1;
103 while (T(Geometry_node(1),j)-T_disch_stop)>0.1
104     [T(:,j+1),T_out_1(1,j+1),T_out_n(1,j+1)] = ...
        STEP_FILO(Storage,Geometry_node,Inlet,m_disch,m_ch, ...
        T_disch,T_ch,T_amb,T(:,j),T_out_1(1,j),T_out_n(1,j),dt_disch,f_crit);
105     j = j+1;
106 end
107 nt_ges=j-1;
108 nt_disch=nt_ges-(nt_ch+nt_stby);
109
110 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Rueckgabewerte
111 m_down=zeros(1,nt_ges);
112 m_down(1,1:nt_ch)=dV_ch*density(T_ch);
113
114 T_down=zeros(1,nt_ges);
115 T_down(1,:)=T_ch;
116
117 m_up=zeros(1,nt_ges);
118 m_up(1,nt_ch+nt_stby+1:nt_ges)=dV_disch*density(T_disch);
119
120 T_up=zeros(1,nt_ges);
121 T_up(1,:)=T_disch;
122
123 time=zeros(1,nt_ges+1);
124 for j=1:nt_ch
125     time(j+1)=time(j)+dt_ch;
126 end
127 for j=nt_ch+1:nt_ch+nt_stby
128     time(j+1)=time(j)+dt_stby;
129 end
130 for j=nt_ch+nt_stby+1:nt_ges
131     time(j+1)=time(j)+dt_disch;
132 end
133 end

```

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %%Falkner Stefan, 0826830 %%
3 %%Wien, 14.5.2014 %%
4 %%TESTING_NB %%
5 %%Funktion simuliert Sequenz der Pruefung NB aus DIN EN 12977-3:2012-06 %%
6 %%NB1...Konditionierung bis ein stat. Zustand erreicht ist %%
7 %%NB2...Beladen eines bestimmten Volumenanteils %%
8 %%NB3...Stand-By %%
9 %%NB4...Entladen bis T_disch_stop im oberen Bereich erreicht wird %%
10 %%Anmerkung: Uebergabe der Volumenstroeme, Temperaturen,... obliegen %%
11 %%dem Benutzer. Somit kann diese Fkt. auch Pruefung NA simulieren. Bzw. %%

```

```

12 %%ist es moeglich andere aehnliche Szenarien nachzustellen.          %%
13 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14 %%                               Input                                %%
15 %%H=Storage(1)...Speicherhoehe in m (Dim: 1x1)                       %%
16 %%Dia=Storage(2)...Speicherdurchmesser in m (Dim: 1x1)              %%
17 %%k_rad=Storage(3)...radialer Waermedurchgang in W/m^2K (Dim: 1x1)  %%
18 %%k_ax=Storage(4)...axialer Waermedurchgang in W/m^2K (Dim: 1x1)  %%
19 %%lambda_eff=Storage(5)...effektive axiale Waermeleitfaehigkeit in W/mK %%
20 %% (Dim: 1x1)                                                       %%
21 %%n...Anzahl der Knoten (Dim: 1x1)                                   %%
22 %%d_in=Inlet(1)...Einstroemdurchmesser in m (Dim: 1x1)             %%
23 %%a=Inlet(2)...epsilon_in=a*(Re/Ri)^b (Dim: 1x1)                   %%
24 %%b=Inlet(3) (Dim: 1x1)                                             %%
25 %%phi=Inlet(4)...Exponent fuer epsilon_eff;                          %%
26 %%phi=0 --> hypberbolischer Verlauf von epsilon_eff (Dim: 1x1)     %%
27 %%Inlet(5)==1...Beruecksichtigung der Einstroemverhaeltnisse (Dim: 1x1) %%
28 %%Inlet(5)==else...keine Beruecksichtigung der Einstroemverhaeltnisse %%
29 %% (Dim: 1x1)                                                       %%
30 %%dV_ch...Beladevolumenstrom in m^3/s (Dim: 1x1)                   %%
31 %%T_ch...Beladetemp. in Grad C (Dim: 1x1)                           %%
32 %%vol_ch_stop...Volumenanteil der beladen werden soll (Dim: 1x1)  %%
33 %%t_stby...Dauer Stand by in s (Dim: 1x1)                           %%
34 %%dV_disch...Entladevolumenstrom in m^3/s (Dim: 1x1)              %%
35 %%T_disch...Entladetemp. in Grad C (Dim: 1x1)                       %%
36 %%T_disch_stop...zu erreichende Austrittstemp. in Grad C (Dim: 1x1) %%
37 %%T_amb...Temp. der Umgebung in Grad C (Dim: 1x1)                  %%
38 %%T_ini...Anfangstemp. in Grad C (Dim: 1x1)                         %%
39 %%f_crit...Faktor fuer max. Zeitschrittlaenge dt_max=f_crit*dt_crit %%
40 %% (Dim: 1x1)                                                       %%
41 %%                               Output                               %%
42 %%T...Speichertemperatur in Grad C (Dim: n x nt_ges+1)              %%
43 %%z...z-Koordinaten der Knoten in m (Dim: n x 1)                   %%
44 %%time...Zeitvektor in s (Dim: 1 x nt_ges+1)                        %%
45 %%m_down...Belademassenstrom ueber Zeit in kg/s (Dim: 1 x nt_ges) %%
46 %%m_up...Entlademassenstrom ueber Zeit in kg/s (Dim: 1 x nt_ges)  %%
47 %%T_down...Beladetemp. ueber Zeit in Grad C (Dim: 1 x nt_ges)     %%
48 %%T_up...Entladetemp. ueber Zeit in Grad C (Dim: 1 x nt_ges)      %%
49 %%T_out_1...Austrittstemp. unten in Grad C (Dim: 1 x nt_ges+1)    %%
50 %%T_out_n...Austrittstemp. oben in Grad C (Dim: 1 x nt_ges+1)     %%
51 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
52
53
54 function [ T, z, time, m_down, m_up, T_down, T_up, T_out_1, T_out_n ] = ...
    TESTING_NB(Storage, n, Inlet, dV_ch, T_ch, vol_ch_stop, ...
    t_stby, dV_disch, T_disch, T_disch_stop, T_amb, T_ini, f_crit)
55
56 Geometry_node = GEOMETRY(n, Storage(1), Storage(2)); ...

```

```

        %Knotengeometrie des Speichers
57 z = GET_Z(Geometry_node(3),Geometry_node(1)); %zugehoerige z-Koordinaten
58 V = Geometry_node(1)*Geometry_node(2); %Speichervolumen in m^3 (V=n*V_node)
59
60 %Anfangszustand-Knoditionierung
61 j=1;
62 T=zeros(Geometry_node(1),j);
63 T(:,j)=T_ini;
64 %Austrittstemp.
65 T_out_1=T(1,1);
66 T_out_n=T(Geometry_node(1),1);
67
68 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Beladevorgang
69 dt_crit_ch = Geometry_node(2)/dV_ch; %kritischer Zeitschritt in s beim ...
        Beladen
70 dt_ch = f_crit*dt_crit_ch; %Schaetzung des Zeitschritts beim ...
        Beladevorgang in s
71 t_ch = vol_ch_stop*V/dV_ch; %Beladedauer in s
72 nt_ch = ceil(t_ch/dt_ch); %Anzahl der Zeitschritte beim ...
        Beladen (aufgerundet)
73 dt_ch = t_ch/nt_ch; %Zeitschritt beim Beladen in s
74 m_ch=dV_ch*density(T_ch); %Massenstrom beim Beladen in kg/s
75 m_disch=0; %muss somit 0 sein
76
77 for j=1:nt_ch
78     [T(:,j+1),T_out_1(1,j+1),T_out_n(1,j+1)] = ...
        STEP_FILO(Storage,Geometry_node,Inlet,m_disch,m_ch, ...
        T_disch,T_ch,T_amb,T(:,j),T_out_1(1,j),T_out_n(1,j),dt_ch,f_crit);
79 end
80
81
82 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Stand-By
83 if t_stby≠0
84     dt_stby = 3600; %Zeitschritt im Stand-By, soll ca. eine Stunde betragen
85     nt_stby = ceil(t_stby/dt_stby); %aufrunden
86     dt_stby = t_stby/nt_stby; %Zeitschritt im Stand-By in s
87     m_ch = 0; %im StandBy gleich 0
88     m_disch = 0; %im StandBy gleich 0
89     for j=nt_ch+1:nt_ch+nt_stby
90         [T(:,j+1),T_out_1(1,j+1),T_out_n(1,j+1)] = ...
            STEP_FILO(Storage,Geometry_node,Inlet,m_disch,m_ch,T_disch, ...
            T_ch,T_amb,T(:,j),T_out_1(1,j),T_out_n(1,j),dt_stby,f_crit);
91     end
92 else
93     nt_stby=0;
94 end
95

```

```

96 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Entladevorgang
97 dt_crit_disch = Geometry_node(2)/dV_disch;    %kritischer Zeitschritt in s ...
        beim Entladen
98 dt_disch = f_crit*dt_crit_disch;            %Zeitschrittdauer in s beim ...
        Entladen
99 m_disch=dV_disch*density(T_disch);    %Massenstrom beim Entladen in kg/s
100 m_ch=0;                                    %muss somit 0 sein
101 %bis die Austrittstemp. (oben) annaeherd T_disch_stop erreicht hat
102 j=nt_ch+nt_stby+1;
103 while (T(Geometry_node(1),j)-T_disch_stop)>0.1
104     [T(:,j+1),T_out_1(1,j+1),T_out_n(1,j+1)] = ...
        STEP_FILO(Storage,Geometry_node,Inlet,m_disch,m_ch, ...
        T_disch,T_ch,T_amb,T(:,j),T_out_1(1,j),T_out_n(1,j),dt_disch,f_crit);
105     j = j+1;
106 end
107 nt_ges=j-1;
108 nt_disch=nt_ges-(nt_ch+nt_stby);
109
110 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Rueckgabewerte
111 m_down=zeros(1,nt_ges);
112 m_down(1,1:nt_ch)=dV_ch*density(T_ch);
113
114 T_down=zeros(1,nt_ges);
115 T_down(1,:)=T_ch;
116
117 m_up=zeros(1,nt_ges);
118 m_up(1,nt_ch+nt_stby+1:nt_ges)=dV_disch*density(T_disch);
119
120 T_up=zeros(1,nt_ges);
121 T_up(1,:)=T_disch;
122
123 time=zeros(1,nt_ges+1);
124 for j=1:nt_ch
125     time(j+1)=time(j)+dt_ch;
126 end
127 for j=nt_ch+1:nt_ch+nt_stby
128     time(j+1)=time(j)+dt_stby;
129 end
130 for j=nt_ch+nt_stby+1:nt_ges
131     time(j+1)=time(j)+dt_disch;
132 end
133 end

```

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %%Falkner Stefan, 0826830                %%
3 %%Wien, 14.5.2014                        %%

```

```

4 %%TESTING_V %%
5 %%Funktion simuliert Sequenz der Pruefung V aus DIN EN 12977-3:2012-06 %%
6 %%V1...Konditionierung bis ein stat. Zustand erreicht ist %%
7 %%V2...Beladen bis T_ch_stop im unteren Bereich des Tanks erreicht wird %%
8 %%V3...Entladen von eines bestimmten Volumenanteils %%
9 %%V4...Beladen eines bestimmten Volumenanteils %%
10 %%V5...Stand-By %%
11 %%V6...Entladen bis T_disch_stop im oberen Bereich erreicht wird %%
12 %%Anmerkung: Uebergabe der Volumenstroeme, Temperaturen,... obliegen %%
13 %%dem Benutzer. Somit ist es moeglich andere aehnliche Szenarien %%
14 %%nachzustellen. %%
15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16 %% Input %%
17 %%H=Storage(1)...Speicherhoehe in m (Dim: 1x1) %%
18 %%Dia=Storage(2)...Speicherdurchmesser in m (Dim: 1x1) %%
19 %%k_rad=Storage(3)...radialer Waermedurchgang in W/m^2K (Dim: 1x1) %%
20 %%k_ax=Storage(4)...axialer Waermedurchgang in W/m^2K (Dim: 1x1) %%
21 %%lambda_eff=Storage(5)...effektive axiale Waermeleitfaehigkeit in W/mK %%
22 %% (Dim: 1x1) %%
23 %%n...Anzahl der Knoten (Dim: 1x1) %%
24 %%d_in=Inlet(1)...Einstroemdurchmesser in m (Dim: 1x1) %%
25 %%a=Inlet(2)...epsilon_in=a*(Re/Ri)^b (Dim: 1x1) %%
26 %%b=Inlet(3) (Dim: 1x1) %%
27 %%phi=Inlet(4)...Exponent fuer epsilon_eff; %%
28 %%phi=0 --> hypberbolischer Verlauf von epsilon_eff (Dim: 1x1) %%
29 %%Inlet(5)==1...Beruecksichtigung der Einstroemverhaeltnisse (Dim: 1x1) %%
30 %%Inlet(5)==else...keine Beruecksichtigung der Einstroemverhaeltnisse %%
31 %% (Dim: 1x1) %%
32 %%dV_ch2...Beladevolumenstorm V2 in m^3/s (Dim: 1x1) %%
33 %%T_ch2...Beladetemp. V2 in Grad C (Dim: 1x1) %%
34 %%T_ch_stop2...zu erreichende Austrittstemp. V2 in Grad C (Dim: 1x1) %%
35 %%dV_disch3...Entladevolumenstorm V3 in m^3/s (Dim: 1x1) %%
36 %%T_disch3...Entladetemp. V3 in Grad C (Dim: 1x1) %%
37 %%vol_disch_stop3...Volumenanteil der entladen werden soll V3 (Dim: 1x1) %%
38 %%dV_ch4...Beladevolumenstorm V4 in m^3/s (Dim: 1x1) %%
39 %%T_ch4...Beladetemp. V4 in Grad C (Dim: 1x1) %%
40 %%vol_ch_stop4...Volumenanteil der betladen werden soll V4 (Dim: 1x1) %%
41 %%t_stby5...Dauer Stand by V5 in s (Dim: 1x1) %%
42 %%dV_disch6...Entladevolumenstorm V6 in m^3/s (Dim: 1x1) %%
43 %%T_disch6...Entladetemp. V6 in Grad C (Dim: 1x1) %%
44 %%T_disch_stop6...zu erreichende Austrittstemp. V6 in Grad C (Dim: 1x1) %%
45 %%T_amb...Temp. der Umgebung in Grad C (Dim: 1x1) %%
46 %%T_ini...Anfangstemp. in Grad C (Dim: 1x1) %%
47 %%f_crit...Faktor fuer max. Zeitschrittlaenge dt_max=f_crit*dt_crit %%
48 %% (Dim: 1x1) %%
49 %% Output %%
50 %%T...Speichertemperatur in Grad C (Dim: n x nt_ges+1) %%

```

```

51 %%z...z-Koordinaten der Knoten in m (Dim: n x 1) %%
52 %%time...Zeitvektor in s (Dim: 1 x nt_ges+1) %%
53 %%m_down...Belademassenstrom ueber Zeit in kg/s (Dim: 1 x nt_ges) %%
54 %%m_up...Entlademassenstrom ueber Zeit in kg/s (Dim: 1 x nt_ges) %%
55 %%T_down...Beladetemp. ueber Zeit in Grad C (Dim: 1 x nt_ges) %%
56 %%T_up...Entladetemp. ueber Zeit in Grad C (Dim: 1 x nt_ges) %%
57 %%T_out_1...Austrittstemp. unten in Grad C (Dim: 1 x nt_ges+1) %%
58 %%T_out_n...Austrittstemp. oben in Grad C (Dim: 1 x nt_ges+1) %%
59 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
60
61
62 function [ T,z,time,m_down,m_up,T_down,T_up,T_out_1,T_out_n ] = ...
    TESTING_V(Storage,n,Inlet,dV_ch2,T_ch2,T_ch_stop2,dV_disch3, ...
    T_disch3,vol_disch_stop3,dV_ch4,T_ch4,vol_ch_stop4,t_stby5, ...
    dV_disch6,T_disch6,T_disch_stop6,T_amb,T_ini,f_crit)
63
64 %Knotengeometrie des Speichers
65 Geometry_node = GEOMETRY(n,Storage(1),Storage(2));
66 z = GET_Z(Geometry_node(3),Geometry_node(1)); %zugehoerige z-Koordinaten
67 V = Geometry_node(1)*Geometry_node(2); %Speichervolumen in m^3 (V=n*V_node)
68
69 %Anfangszustand-Knoditionierung V1
70 j=1;
71 T=zeros(Geometry_node(1),j);
72 T(:,j)=T_ini;
73 %Austrittstemp.
74 T_out_1=T(1,1);
75 T_out_n=T(Geometry_node(1),1);
76
77 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Beladevorgang V2
78 dt_crit_ch2 = Geometry_node(2)/dV_ch2; %kritischer Zeitschritt in s beim ...
    Beladen
79 dt_ch2 = f_crit*dt_crit_ch2; %Zeitschrittdauer in s beim Beladen
80 m_ch2=dV_ch2*density(T_ch2); %Massenstrom beim Beladen in kg/s
81 m_disch2=0; %muss somit 0 sein
82 %bis die Austrittstemp. (unten) T_ch_stop2 erreicht hat
83 while T(1,j)<T_ch_stop2
84     [T(:,j+1),T_out_1(1,j+1),T_out_n(1,j+1)] = ...
        STEP_FILO(Storage,Geometry_node,Inlet,m_disch2,m_ch2,0,T_ch2, ...
        T_amb,T(:,j),T_out_1(1,j),T_out_n(1,j),dt_ch2,f_crit);
85     j = j+1;
86 end
87 nt_ch2=j-1; %Anzahl der Beladezeitschritte
88
89 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Entladevorgang V3
90 dt_crit_disch3 = Geometry_node(2)/dV_disch3; %kritischer Zeitschritt ...
    in s beim Entladen

```

```

91 dt_disch3 = f_crit*dt_crit_disch3;           %Schaetzung des ...
    Zeitschritts beim Entladevorgang in s
92 t_disch3 = vol_disch_stop3*V/dV_disch3;     %Entladedauer in s
93 nt_disch3 = ceil(t_disch3/dt_disch3);       %Anzahl der ...
    Zeitschritte beim Entladen (aufgerundet)
94 dt_disch3 = t_disch3/nt_disch3;           %Zeitschritt beim ...
    Entladen in s
95 m_disch3=dV_disch3*density(T_disch3);      %Massenstrom beim Entladen in kg/s
96 m_ch3=0;                                   %muss somit 0 sein
97 for j=nt_ch2+1:(nt_ch2+nt_disch3)
98     [T(:,j+1),T_out_1(1,j+1),T_out_n(1,j+1)] = ...
        STEP_FILO(Storage,Geometry_node,Inlet,m_disch3,m_ch3,T_disch3,0, ...
        T_amb,T(:,j),T_out_1(1,j),T_out_n(1,j),dt_disch3,f_crit);
99 end
100
101 %%%%%%%%%%%%%%Beladevorgang V4
102 dt_crit_ch4 = Geometry_node(2)/dV_ch4;     %kritischer Zeitschritt in s beim ...
    Beladen
103 dt_ch4 = f_crit*dt_crit_ch4;              %Schaetzung des Zeitschritts beim ...
    Beladevorgang in s
104 t_ch4 = vol_ch_stop4*V/dV_ch4;            %Beladedauer in s
105 nt_ch4 = ceil(t_ch4/dt_ch4);              %Anzahl der Zeitschritte beim ...
    Beladen (aufgerundet)
106 dt_ch4 = t_ch4/nt_ch4;                    %Zeitschritt beim Beladen in s
107 m_ch4=dV_ch4*density(T_ch4);             %Massenstrom beim Beladen in kg/s
108 m_disch4=0;                               %muss somit 0 sein
109 for j=(nt_ch2+nt_disch3+1):(nt_ch2+nt_disch3+nt_ch4)
110     [T(:,j+1),T_out_1(1,j+1),T_out_n(1,j+1)] = ...
        STEP_FILO(Storage,Geometry_node,Inlet,m_disch4,m_ch4,0,T_ch4, ...
        T_amb,T(:,j),T_out_1(1,j),T_out_n(1,j),dt_ch4,f_crit);
111 end
112
113 %%%%%%%%%%%%%%Stand-By V5
114 if t_stby5≠0
115     dt_stby5 = 3600;%Zeitschritt im Stand-By, soll ca. eine Stunde betragen
116     nt_stby5 = ceil(t_stby5/dt_stby5);     %aufrunden
117     dt_stby5 = t_stby5/nt_stby5;         %Zeitschritt im Stand-By in s
118     m_ch5 = 0;                            %im StandBy gleich 0
119     m_disch5 = 0;                          %im StandBy gleich 0
120     for j=(nt_ch2+nt_disch3+nt_ch4+1):(nt_ch2+nt_disch3+nt_ch4+nt_stby5)
121         [T(:,j+1),T_out_1(1,j+1),T_out_n(1,j+1)] = ...
            STEP_FILO(Storage,Geometry_node,Inlet,m_disch5,m_ch5,0,0, ...
            T_amb,T(:,j),T_out_1(1,j),T_out_n(1,j),dt_stby5,f_crit);
122     end
123 else
124     nt_stby5=0;
125 end

```

```

126
127 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Entladevorgang V6
128 dt_crit_disch6 = Geometry_node(2)/dV_disch6;      %kritischer Zeitschritt in ...
           s beim Entladen
129 dt_disch6 = f_crit*dt_crit_disch6;                %Zeitschrittdauer in s ...
           beim Entladen
130 m_disch6=dV_disch6*density(T_disch6);  %Massenstrom beim Entladen in kg/s
131 m_ch6=0;                                         %muss somit 0 sein
132 %bis die Austrittstemp. (oben) annaehern T_disch_stop erreicht hat
133 j=nt_ch2+nt_disch3+nt_ch4+nt_stby5+1;
134 while (T(Geometry_node(1),j)-T_disch_stop6)>0.1
135     [T(:,j+1),T_out_1(1,j+1),T_out_n(1,j+1)] = ...
           STEP_FILO(Storage,Geometry_node,Inlet,m_disch6,m_ch6,T_disch6,0, ...
           T_amb,T(:,j),T_out_1(1,j),T_out_n(1,j),dt_disch6,f_crit);
136     j = j+1;
137 end
138 nt_ges=j-1;
139 nt_disch6=nt_ges-(nt_ch2+nt_disch3+nt_ch4+nt_stby5);
140
141
142 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Rueckgabewerte
143 m_down=zeros(1,nt_ges);
144 T_down=zeros(1,nt_ges);
145 m_up=zeros(1,nt_ges);
146 T_up=zeros(1,nt_ges);
147
148 m_down(1,1:nt_ch2)=m_ch2;
149 m_down(1,(nt_ch2+nt_disch3+1):(nt_ch2+nt_disch3+nt_ch4))=m_ch4;
150 T_down(1,1:nt_ch2+nt_disch3)=T_ch2;
151 T_down(1,(nt_ch2+nt_disch3+1):nt_ges)=T_ch4;
152
153 m_up((nt_ch2+1):(nt_ch2+nt_disch3))=m_disch3;
154 m_up((nt_ch2+nt_disch3+nt_ch4+nt_stby5+1):(nt_ges))=m_disch6;
155 T_up(1:(nt_ch2+nt_disch3+nt_ch4+nt_stby5))=T_disch3;
156 T_up((nt_ch2+nt_disch3+nt_ch4+nt_stby5+1):(nt_ges))=T_disch6;
157
158 time=zeros(1,nt_ges+1);
159 for j=1:nt_ch2
160     time(j+1)=time(j)+dt_ch2;
161 end
162 for j=(nt_ch2+1):(nt_ch2+nt_disch3)
163     time(j+1)=time(j)+dt_disch3;
164 end
165 for j=(nt_ch2+nt_disch3+1):(nt_ch2+nt_disch3+nt_ch4)
166     time(j+1)=time(j)+dt_ch4;
167 end
168 for j=(nt_ch2+nt_disch3+nt_ch4+1):(nt_ch2+nt_disch3+nt_ch4+nt_stby5)

```

```

169     time(j+1)=time(j)+dt_stby5;
170 end
171 for j=(nt_ch2+nt_disch3+nt_ch4+nt_stby5+1):(nt_ges)
172     time(j+1)=time(j)+dt_disch6;
173 end
174
175 end

```

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %%Falkner Stefan, 0826830                                           %%
3  %%Wien, 24.7.2014                                                 %%
4  %%ANALYSE_CYCLE                                                  %%
5  %%Funktion analysiert einen gegebenen Speicherzyklus hinsichtlich Exergie%%
6  %%Enthalpie und Entropie                                          %%
7  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8  %%%                                                                %%
9  %%T...Temperaturverteilung im Speicher in Grad C (Dim: n x (nt+1)) %%
10 %%time...Zeitvektor in s (Dim: 1 x (nt+1))                       %%
11 %%m_down...Massenstrom nach unten in kg/s (Dim: 1 x nt)          %%
12 %%m_up...Massenstrom nach oben in kg/s (Dim: 1 x nt)             %%
13 %%T_down...Beladetemperatur in Grad C (Dim: 1 x nt)              %%
14 %%T_up...Entladetemperatur in Grad C (Dim: 1 x nt)               %%
15 %%T_out_1...Austrittstemp. unten in Grad C (Dim: 1 x nt+1)      %%
16 %%T_out_n...Austrittstemp. oben in Grad C (Dim: 1 x nt+1)       %%
17 %%Storage...Speicherparameter (Dim: 5 x 1)                       %%
18 %%H=Storage(1)...Speicherhoehe in m (Dim: 1x1)                   %%
19 %%Dia=Storage(2)...Speicherdurchmesser in m (Dim: 1x1)           %%
20 %%k_rad=Storage(3)...radialer Waermedurchgang in W/m^2K (Dim: 1x1) %%
21 %%k_ax=Storage(4)...axialer Waermedurchgang in W/m^2K (Dim: 1x1) %%
22 %%lambda_eff=Storage(5)...effektive axiale Waermeleitfaehigkeit in W/mK %%
23 %% (Dim: 1x1)                                                     %%
24 %%T_amb...Umgebungstemperatur in Grad C (Dim: 1 x 1)            %%
25 %%%                                                                %%
26 %%%                                                                %%
27 %%Enth...Enthalpie in J (Dim: 8 x 1)                               %%
28 %%Enth_st (nt+1), DeltaEnth_st (nt+1), DeltaEnth_disch (nt+1),  %%
29 %%DeltaEnth_ch (nt+1), DeltaEnth_in_bottom (nt+1), DeltaEnth_out_bottom %%
30 %% (nt+1), DeltaEnth_flow (nt+1), DeltaEnth_loss (nt+1)          %%
31 %%Enth_mix...Enthalpie Referenzspeicher in J (Dim: 8 x 1)       %%
32 %%Enth_st_mix (nt+1), DeltaEnth_st_mix (nt+1), DeltaEnth_disch_mix %%
33 %% (nt+1), DeltaEnth_ch_mix (nt+1), DeltaEnth_in_bottom_mix (nt+1), %%
34 %%DeltaEnth_out_bottom_mix (nt+1), DeltaEnth_flow_mix (nt+1),    %%
35 %%DeltaEnth_loss_mix (nt+1)                                       %%
36 %%Ex...Exergie in J (Dim: 9 x 1)                                  %%
37 %%Ex_st (nt+1), DeltaEx_st (nt+1), DeltaEx_disch (nt+1), DeltaEx_ch %%
38 %% (nt+1), DeltaEx_in_bottom (nt+1), DeltaEx_out_bottom (nt+1),  %%
39 %%DeltaEx_flow (nt+1), DeltaEx_loss (nt+1), DeltaEx_int (nt+1)   %%

```

```

39 %%Ex_mix...Exergie Referenzspeicher in J (Dim: 9 x 1) %%
40 %%Ex_st_mix (nt+1), DeltaEx_st_mix (nt+1), DeltaEx_disch_mix (nt+1), %%
41 %%DeltaEx_ch_mix (nt+1), DeltaEx_in_bottom_mix (nt+1), %%
42 %%DeltaEx_out_bottom_mix (nt+1), DeltaEx_flow_mix (nt+1), %%
43 %%DeltaEx_loss_mix (nt+1), DeltaEx_int_mix (nt+1) %%
44 %%S...Entropie in J/K (Dim: 9 x 1) %%
45 %%S_st (nt+1), DeltaS_st (nt+1), DeltaS_disch (nt+1), DeltaS_ch (nt+1), %%
46 %%DeltaS_in_bottom (nt+1), DeltaS_out_bottom (nt+1), DeltaS_flow (nt+1), %%
47 %%DeltaS_loss (nt+1), DeltaS_int (nt+1) %%
48 %%S...Entropie_mix Referenzspeicher in J/K (Dim: 9 x 1) %%
49 %%S_st_mix (nt+1), DeltaS_st_mix (nt+1), DeltaS_disch_mix (nt+1), %%
50 %%DeltaS_ch_mix (nt+1), DeltaS_in_bottom_mix (nt+1), %%
51 %%DeltaS_out_bottom_mix (nt+1), DeltaS_flow_mix (nt+1), DeltaS_loss_mix %%
52 %% (nt+1), DeltaS_int_mi (nt+1) %%
53 %%energetischer Wirkungsgrad: %%
54 %%eta[eta_Aeta_B (nt+1); eta_C (nt+1); eta_D (nt+1); eta_E (nt+1)] %%
55 %%exergetischer Wirkungsgrad: %%
56 %%psi[{psi_A (nt+1); psi_B (nt+1); psi_C (nt+1); psi_D (nt+1); psi_E %%
57 %% (nt+1); psi_int (nt+1); psi_int_loss (nt+1)] %%
58 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
59 %%Kuerzel: st...Speicherinhalt, Delta...Differenz zw. betrachteten %%
60 %%Zeitpunkt und Zeitpunkt 0, disch...Entladen bzw. Austritt oben, ch... %%
61 %%Beladen bzw. Eintritt oben, in_bottom...Eintritt unten, out_bottom... %%
62 %%Austritt unten, flow...resultierender Fluss, loss...Verluste ueber %%
63 %%Behaelterwand, int...interner Verlust durch Mischen, mix... %%
64 %%Referenzspeicher - vollstaendig durchmischt %%
65 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
66 %%Voraussetzungen: uniform Grid %%
67 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
68
69
70 function [ Enth,Enth_mix,Ex,Ex_mix,S,S_mix,T_mix,eta,psi ] = ...
    ANALYSE_CYCLE(T,time,m_down,m_up, ...
    T_down,T_up,T_out_1,T_out_n,Storage,T_amb)
71
72 dT_abs=273.15;
73
74 [n,nt]=size(T);
75 nt=nt-1;
76
77 H=Storage(1);
78 Dia=Storage(2);
79 k_rad=Storage(3);
80 k_ax=Storage(4);
81 A_ax=Dia^2*pi()*0.25;
82 V=A_ax*H;
83 V_node=V/n;

```

```

84 A_rad=Dia*pi()*H/n;
85 kA_rad=k_rad*A_rad;
86 kA_ax=k_ax*A_ax;
87 enth_amb=enthalpy(T_amb);
88
89
90 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
91 %%                               Zeit                               %%
92 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
93 dt=zeros(1,nt);
94 for j=1:nt
95     dt(j)=time(j+1)-time(j);
96 end
97
98
99 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
100 %%                               Speicherinhalt                       %%
101 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
102 Enth_st = zeros(1,nt+1);
103 Ex_st = zeros(1,nt+1);
104 S_st = zeros(1,nt+1);
105 DeltaEnth_st = zeros(1,nt);
106 DeltaEx_st = zeros(1,nt);
107 DeltaS_st = zeros(1,nt);
108 for j=1:nt+1
109     for i=1:n
110         mass = V_node*density(T(i,j));
111         Enth_st(j) = Enth_st(j) + mass*enthalpy(T(i,j));
112         Ex_st(j) = Ex_st(j) + mass*exergy(T(i,j),T_amb);
113         S_st(j) = S_st(j) + mass*entropy(T(i,j));
114     end
115     DeltaEnth_st(j) = Enth_st(j)-Enth_st(1);
116     DeltaEx_st(j) = Ex_st(j)-Ex_st(1);
117     DeltaS_st(j) = S_st(j)-S_st(1);
118 end
119 Enth_amb=V*density(T_amb)*enth_amb;
120
121 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
122 %%                               Referenz - voll durchmischter Speicher           %%
123 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
124 T_mix=zeros(1,nt+1);
125
126 T_mix(1)=T_M(Enth_st(1),min(T(:,1)),max(T(:,1)),V);
127
128 kA=2*k_ax*A_ax+n*k_rad*A_rad;
129 for j=1:nt
130     C_st=heat_capacity_int(T_mix(j))*density(T_mix(j))*V;

```

```

131     if m_up(j)≠0
132         dC_flow = ...
133         m_up(j) * (enthalpy(T_mix(j)) - enthalpy(T_up(j))) / (T_mix(j) - T_up(j));
134         T_inf = (kA*T_amb + dC_flow*T_up(j)) / (kA + dC_flow);
135     elseif m_down(j)≠0
136         dC_flow = ...
137         m_down(j) * (enthalpy(T_mix(j)) - enthalpy(T_down(j))) / (T_mix(j) - T_down(j));
138         T_inf = (kA*T_amb + dC_flow*T_down(j)) / (kA + dC_flow);
139     else
140         dC_flow = 0;
141         T_inf = T_amb;
142     end
143     a = (kA + dC_flow) / C_st;
144     T_mix(j+1) = T_inf + (T_mix(j) - T_inf) * exp(-a*dt(j));
145 end
146
147 Enth_st_mix = zeros(1, nt+1);
148 Ex_st_mix = zeros(1, nt+1);
149 S_st_mix = zeros(1, nt+1);
150 DeltaEnth_st_mix = zeros(1, nt);
151 DeltaEx_st_mix = zeros(1, nt);
152 DeltaS_st_mix = zeros(1, nt);
153 for j=1:nt+1
154     mass = V*density(T_mix(j));
155     Enth_st_mix(j) = mass*enthalpy(T_mix(j));
156     Ex_st_mix(j) = mass*exergy(T_mix(j), T_amb);
157     S_st_mix(j) = mass*entropy(T_mix(j));
158     DeltaEnth_st_mix(j) = Enth_st_mix(j) - Enth_st_mix(1);
159     DeltaEx_st_mix(j) = Ex_st_mix(j) - Ex_st_mix(1);
160     DeltaS_st_mix(j) = S_st_mix(j) - S_st_mix(1);
161 end
162
163 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
164 %%                                     konvektiver Transport                                     %%
165 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
166
167 %Stroeme-Speicher
168 dEnth_disch = zeros(1, nt);
169 dEnth_ch = zeros(1, nt);
170 dEnth_in_bottom = zeros(1, nt);
171 dEnth_out_bottom = zeros(1, nt);
172 dEnth_flow = zeros(1, nt);
173 dEx_disch = zeros(1, nt);
174 dEx_ch = zeros(1, nt);

```

```

175 dEx_in_bottom = zeros(1,nt);
176 dEx_out_bottom = zeros(1,nt);
177 dEx_flow = zeros(1,nt);
178 dS_disch = zeros(1,nt);
179 dS_ch = zeros(1,nt);
180 dS_in_bottom = zeros(1,nt);
181 dS_out_bottom = zeros(1,nt);
182 dS_flow = zeros(1,nt);
183 %Stroeme-Referenz
184 dEnth_disch_mix = zeros(1,nt);
185 dEnth_ch_mix = zeros(1,nt);
186 dEnth_in_bottom_mix = zeros(1,nt);
187 dEnth_out_bottom_mix = zeros(1,nt);
188 dEnth_flow_mix = zeros(1,nt);
189 dEx_disch_mix = zeros(1,nt);
190 dEx_ch_mix = zeros(1,nt);
191 dEx_in_bottom_mix = zeros(1,nt);
192 dEx_out_bottom_mix = zeros(1,nt);
193 dEx_flow_mix = zeros(1,nt);
194 dS_disch_mix = zeros(1,nt);
195 dS_ch_mix = zeros(1,nt);
196 dS_in_bottom_mix = zeros(1,nt);
197 dS_out_bottom_mix = zeros(1,nt);
198 dS_flow_mix = zeros(1,nt);
199 for j=1:nt
200     %Stroeme-Speicher
201     dEnth_disch(j) = m_up(j)*enthalpy(T_out_n(j+1));
202     dEnth_ch(j) = m_down(j)*enthalpy(T_down(j));
203     dEnth_in_bottom(j) = m_up(j)*enthalpy(T_up(j));
204     dEnth_out_bottom(j) = m_down(j)*enthalpy(T_out_1(j+1));
205     dEnth_flow(j) = ...
        dEnth_ch(j)+dEnth_in_bottom(j)-dEnth_disch(j)-dEnth_out_bottom(j);
206     dEx_disch(j) = m_up(j)*exergy(T_out_n(j+1),T_amb);
207     dEx_ch(j) = m_down(j)*exergy(T_down(j),T_amb);
208     dEx_in_bottom(j) = m_up(j)*exergy(T_up(j),T_amb);
209     dEx_out_bottom(j) = m_down(j)*exergy(T_out_1(j+1),T_amb);
210     dEx_flow(j) = dEx_ch(j)+dEx_in_bottom(j)-dEx_disch(j)-dEx_out_bottom(j);
211     dS_disch(j) = m_up(j)*entropy(T_out_n(j+1));
212     dS_ch(j) = m_down(j)*entropy(T_down(j));
213     dS_in_bottom(j) = m_up(j)*entropy(T_up(j));
214     dS_out_bottom(j) = m_down(j)*entropy(T_out_1(j+1));
215     dS_flow(j) = dS_ch(j)+dS_in_bottom(j)-dS_disch(j)-dS_out_bottom(j);
216     %Stroeme-Referenz
217     dEnth_disch_mix(j) = m_up(j)*enthalpy(T_mix(j));
218     dEnth_ch_mix(j) = m_down(j)*enthalpy(T_down(j));
219     dEnth_in_bottom_mix(j) = m_up(j)*enthalpy(T_up(j));
220     dEnth_out_bottom_mix(j) = m_down(j)*enthalpy(T_mix(j));

```

```

221     dEnth_flow_mix(j) = ...
           dEnth_ch_mix(j)+dEnth_in_bottom_mix(j)-dEnth_disch_mix(j) ...
           -dEnth_out_bottom_mix(j);
222     dEx_disch_mix(j) = m_up(j)*exergy(T_mix(j),T_amb);
223     dEx_ch_mix(j) = m_down(j)*exergy(T_down(j),T_amb);
224     dEx_in_bottom_mix(j) = m_up(j)*exergy(T_up(j),T_amb);
225     dEx_out_bottom_mix(j) = m_down(j)*exergy(T_mix(j),T_amb);
226     dEx_flow_mix(j) = dEx_ch_mix(j)+dEx_in_bottom_mix(j)-dEx_disch_mix(j) ...
           -dEx_out_bottom_mix(j);
227     dS_disch_mix(j) = m_up(j)*entropy(T_mix(j));
228     dS_ch_mix(j) = m_down(j)*entropy(T_down(j));
229     dS_in_bottom_mix(j) = m_up(j)*entropy(T_up(j));
230     dS_out_bottom_mix(j) = m_down(j)*entropy(T_mix(j));
231     dS_flow_mix(j) = ...
           dS_ch_mix(j)+dS_in_bottom_mix(j)-dS_disch_mix(j)-dS_out_bottom_mix(j);
232 end
233
234 %Aufintegriert - Speicher
235 DeltaEnth_disch = zeros(1,nt+1);
236 DeltaEnth_ch = zeros(1,nt+1);
237 DeltaEnth_in_bottom = zeros(1,nt+1);
238 DeltaEnth_out_bottom = zeros(1,nt+1);
239 DeltaEnth_flow = zeros(1,nt+1);
240 DeltaEx_disch = zeros(1,nt+1);
241 DeltaEx_ch = zeros(1,nt+1);
242 DeltaEx_in_bottom = zeros(1,nt+1);
243 DeltaEx_out_bottom = zeros(1,nt+1);
244 DeltaEx_flow = zeros(1,nt+1);
245 DeltaS_disch = zeros(1,nt+1);
246 DeltaS_ch = zeros(1,nt+1);
247 DeltaS_in_bottom = zeros(1,nt+1);
248 DeltaS_out_bottom = zeros(1,nt+1);
249 DeltaS_flow = zeros(1,nt+1);
250 %Aufintegriert - Referenz
251 DeltaEnth_disch_mix = zeros(1,nt+1);
252 DeltaEnth_ch_mix = zeros(1,nt+1);
253 DeltaEnth_in_bottom_mix = zeros(1,nt+1);
254 DeltaEnth_out_bottom_mix = zeros(1,nt+1);
255 DeltaEnth_flow_mix = zeros(1,nt+1);
256 DeltaEx_disch_mix = zeros(1,nt+1);
257 DeltaEx_ch_mix = zeros(1,nt+1);
258 DeltaEx_in_bottom_mix = zeros(1,nt+1);
259 DeltaEx_out_bottom_mix = zeros(1,nt+1);
260 DeltaEx_flow_mix = zeros(1,nt+1);
261 DeltaS_disch_mix = zeros(1,nt+1);
262 DeltaS_ch_mix = zeros(1,nt+1);
263 DeltaS_in_bottom_mix = zeros(1,nt+1);

```

```

264 DeltaS_out_bottom_mix = zeros(1,nt+1);
265 DeltaS_flow_mix = zeros(1,nt+1);
266 for j=1:nt
267     %Aufintegrieren – Speicher
268     DeltaEnth_disch(j+1) = DeltaEnth_disch(j) + dEnth_disch(j)*dt(j);
269     DeltaEnth_ch(j+1) = DeltaEnth_ch(j) + dEnth_ch(j)*dt(j);
270     DeltaEnth_in_bottom(j+1) = DeltaEnth_in_bottom(j) + ...
        dEnth_in_bottom(j)*dt(j);
271     DeltaEnth_out_bottom(j+1) = DeltaEnth_out_bottom(j) + ...
        dEnth_out_bottom(j)*dt(j);
272     DeltaEnth_flow(j+1) = DeltaEnth_flow(j) + dEnth_flow(j)*dt(j);
273     DeltaEx_disch(j+1) = DeltaEx_disch(j) + dEx_disch(j)*dt(j);
274     DeltaEx_ch(j+1) = DeltaEx_ch(j) + dEx_ch(j)*dt(j);
275     DeltaEx_in_bottom(j+1) = DeltaEx_in_bottom(j) + dEx_in_bottom(j)*dt(j);
276     DeltaEx_out_bottom(j+1) = DeltaEx_out_bottom(j) + dEx_out_bottom(j)*dt(j);
277     DeltaEx_flow(j+1) = DeltaEx_flow(j) + dEx_flow(j)*dt(j);
278     DeltaS_disch(j+1) = DeltaS_disch(j) + dS_disch(j)*dt(j);
279     DeltaS_ch(j+1) = DeltaS_ch(j) + dS_ch(j)*dt(j);
280     DeltaS_in_bottom(j+1) = DeltaS_in_bottom(j) + dS_in_bottom(j)*dt(j);
281     DeltaS_out_bottom(j+1) = DeltaS_out_bottom(j) + dS_out_bottom(j)*dt(j);
282     DeltaS_flow(j+1) = DeltaS_flow(j) + dS_flow(j)*dt(j);
283     %Aufintegrieren – Referenz
284     DeltaEnth_disch_mix(j+1) = DeltaEnth_disch_mix(j) + ...
        dEnth_disch_mix(j)*dt(j);
285     DeltaEnth_ch_mix(j+1) = DeltaEnth_ch_mix(j) + dEnth_ch_mix(j)*dt(j);
286     DeltaEnth_in_bottom_mix(j+1) = DeltaEnth_in_bottom_mix(j) + ...
        dEnth_in_bottom_mix(j)*dt(j);
287     DeltaEnth_out_bottom_mix(j+1) = DeltaEnth_out_bottom_mix(j) + ...
        dEnth_out_bottom_mix(j)*dt(j);
288     DeltaEnth_flow_mix(j+1) = DeltaEnth_flow_mix(j) + dEnth_flow_mix(j)*dt(j);
289     DeltaEx_disch_mix(j+1) = DeltaEx_disch_mix(j) + dEx_disch_mix(j)*dt(j);
290     DeltaEx_ch_mix(j+1) = DeltaEx_ch_mix(j) + dEx_ch_mix(j)*dt(j);
291     DeltaEx_in_bottom_mix(j+1) = DeltaEx_in_bottom_mix(j) + ...
        dEx_in_bottom_mix(j)*dt(j);
292     DeltaEx_out_bottom_mix(j+1) = DeltaEx_out_bottom_mix(j) + ...
        dEx_out_bottom_mix(j)*dt(j);
293     DeltaEx_flow_mix(j+1) = DeltaEx_flow_mix(j) + dEx_flow_mix(j)*dt(j);
294     DeltaS_disch_mix(j+1) = DeltaS_disch_mix(j) + dS_disch_mix(j)*dt(j);
295     DeltaS_ch_mix(j+1) = DeltaS_ch_mix(j) + dS_ch_mix(j)*dt(j);
296     DeltaS_in_bottom_mix(j+1) = DeltaS_in_bottom_mix(j) + ...
        dS_in_bottom_mix(j)*dt(j);
297     DeltaS_out_bottom_mix(j+1) = DeltaS_out_bottom_mix(j) + ...
        dS_out_bottom_mix(j)*dt(j);
298     DeltaS_flow_mix(j+1) = DeltaS_flow_mix(j) + dS_flow_mix(j)*dt(j);
299 end
300
301

```

```

302 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
303 %                                konduktiver Transport – Wand                                %%
304 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
305
306 %Stroeme – Speicher
307 dEnth_loss=zeros(1,nt);
308 dEx_loss=zeros(1,nt);
309 dS_loss=zeros(1,nt);
310 %Stroeme – Referenz
311 dEnth_loss_mix=zeros(1,nt);
312 dEx_loss_mix=zeros(1,nt);
313 dS_loss_mix=zeros(1,nt);
314 for j=1:nt
315     %Stroeme – Speicher
316     dEnth_loss(j)=- (kA_rad+kA_ax) * (0.5*(T(1,j)+T(1,j+1))-T_amb);
317     dS_loss(j)=- (kA_rad+kA_ax) * 0.5 * ((T(1,j)-T_amb) / (dT_abs+T(1,j)) ...
318         + (T(1,j+1)-T_amb) / (dT_abs+T(1,j+1)));
319     for i=2:n-1
320         dEnth_loss(j)=dEnth_loss(j)-kA_rad*(0.5*(T(i,j)+T(i,j+1))-T_amb);
321         dS_loss(j)=dS_loss(j)-kA_rad*0.5*((T(i,j)-T_amb) / (dT_abs+T(i,j)) ...
322             + (T(i,j+1)-T_amb) / (dT_abs+T(i,j+1)));
323     end
324     dEnth_loss(j)=dEnth_loss(j)-(kA_rad+kA_ax) * (0.5*(T(n,j)+T(n,j+1))-T_amb);
325     dS_loss(j)=dS_loss(j)-(kA_rad+kA_ax) * 0.5 * ((T(n,j)-T_amb) / ...
326         (dT_abs+T(n,j)) + (T(n,j+1)-T_amb) / (dT_abs+T(n,j+1)));
327     dEx_loss(j)=dEnth_loss(j)-(T_amb+dT_abs)*dS_loss(j);
328     %Stroeme – Referenz
329     dEnth_loss_mix(j)=-kA*(T_mix(j)-T_amb);
330     dS_loss_mix(j)=-kA*(T_mix(j)-T_amb) / (dT_abs+T_mix(j));
331     dEx_loss_mix(j)=dEnth_loss_mix(j)-(T_amb+dT_abs)*dS_loss_mix(j);
332 end
333
334 %Aufintegriert – Speicher
335 DeltaEnth_loss = zeros(1,nt+1);
336 DeltaEx_loss = zeros(1,nt+1);
337 DeltaS_loss = zeros(1,nt+1);
338 %Aufintegriert – Referenz
339 DeltaEnth_loss_mix = zeros(1,nt+1);
340 DeltaEx_loss_mix = zeros(1,nt+1);
341 DeltaS_loss_mix = zeros(1,nt+1);
342 for j=1:nt
343     %Aufintegrieren – Speicher
344     DeltaEnth_loss(j+1) = DeltaEnth_loss(j) + dEnth_loss(j)*dt(j);
345     DeltaEx_loss(j+1) = DeltaEx_loss(j) + dEx_loss(j)*dt(j);
346     DeltaS_loss(j+1) = DeltaS_loss(j) + dS_loss(j)*dt(j);
347     %Aufintegrieren – Referenz
348     DeltaEnth_loss_mix(j+1) = DeltaEnth_loss_mix(j) + dEnth_loss_mix(j)*dt(j);

```



```

391 psi_S_int=zeros(1,nt+1);
392 psi_Ex_int=zeros(1,nt+1);
393 psi_Ex_int_loss=zeros(1,nt+1);
394
395 for j=1:nt+1
396     psi_S_int(j)=1-DeltaS_int(j)/DeltaS_int_mix(j);
397     psi_Ex_int(j)=1-DeltaEx_int(j)/DeltaEx_int_mix(j);
398     psi_Ex_int_loss(j)=1-(DeltaEx_st(j)-DeltaEx_flow(j))/(DeltaEx_st_mix(j) ...
        -DeltaEx_flow_mix(j));
399 end
400
401 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
402 %%                               Wirkungsgrade – Dincer                               %%
403 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
404
405 eta_A=zeros(1,nt+1);
406 eta_B=zeros(1,nt+1);
407 eta_C=zeros(1,nt+1);
408 eta_D=zeros(1,nt+1);
409 eta_E=zeros(1,nt+1);
410 psi_A=zeros(1,nt+1);
411 psi_B=zeros(1,nt+1);
412 psi_C=zeros(1,nt+1);
413 psi_D=zeros(1,nt+1);
414 psi_E=zeros(1,nt+1);
415
416 %Enthalpiestroeme abzueglich der Enthalpie unter Umgebungsbedingungen
417 dEnth_up_amb = zeros(1,nt);
418 dEnth_down_amb = zeros(1,nt);
419 for j=1:nt
420     dEnth_up_amb(j) = m_up(j)*enth_amb;
421     dEnth_down_amb(j) = m_down(j)*enth_amb;
422 end
423 DeltaEnth_up_amb = zeros(1,nt+1);
424 DeltaEnth_down_amb = zeros(1,nt+1);
425
426 for j=1:nt
427     DeltaEnth_up_amb(j+1) = DeltaEnth_up_amb(j) + dEnth_up_amb(j)*dt(j);
428     DeltaEnth_down_amb(j+1) = DeltaEnth_down_amb(j) + dEnth_down_amb(j)*dt(j);
429 end
430
431 for j=1:nt+1
432     eta_A(j) = (DeltaEnth_disch(j)-DeltaEnth_up_amb(j)) / ...
        (DeltaEnth_ch(j)-DeltaEnth_down_amb(j));
433
434     eta_B(j) = (DeltaEnth_disch(j)-DeltaEnth_up_amb(j)+DeltaEnth_st(j)) / ...
        (DeltaEnth_ch(j)-DeltaEnth_down_amb(j));

```

```

435
436   eta_C(j) = (DeltaEnth_disch(j)-DeltaEnth_up_amb(j)) / ...
           (DeltaEnth_ch(j)-DeltaEnth_down_amb(j)+Enth_st(1)-Enth_amb);
437
438   eta_D(j) = ...
           (DeltaEnth_disch(j)-DeltaEnth_up_amb(j)+Enth_st(j)-Enth_amb) / ...
           (DeltaEnth_ch(j)-DeltaEnth_down_amb(j)+Enth_st(1)-Enth_amb);
439
440   eta_E(j) = ...
           (DeltaEnth_disch(j)-DeltaEnth_up_amb(j)+DeltaEnth_out_bottom(j) ...
           -DeltaEnth_down_amb(j)+Enth_st(j)-Enth_amb) / ...
           (DeltaEnth_ch(j)-DeltaEnth_down_amb(j)+DeltaEnth_in_bottom(j) ...
           -DeltaEnth_up_amb(j)+Enth_st(1)-Enth_amb);
441
442   psi_A(j)=DeltaEx_disch(j)/DeltaEx_ch(j);
443   psi_B(j)=(DeltaEx_disch(j)+DeltaEx_st(j))/DeltaEx_ch(j);
444   psi_C(j)=DeltaEx_disch(j)/(DeltaEx_ch(j)+Ex_st(1));
445   psi_D(j)=(DeltaEx_disch(j)+Ex_st(j))/(DeltaEx_ch(j)+Ex_st(1));
446   psi_E(j)=(DeltaEx_disch(j)+DeltaEx_out_bottom(j)+Ex_st(j))/ ...
           (DeltaEx_ch(j)+DeltaEx_in_bottom(j)+Ex_st(1));
447 end
448
449 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
450 %%                               Staffelung der Rueckgabewerte                               %%
451 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
452 Enth=[Enth_st;DeltaEnth_st;DeltaEnth_disch;DeltaEnth_ch; ...
        DeltaEnth_in_bottom;DeltaEnth_out_bottom;DeltaEnth_flow;DeltaEnth_loss];
453 Enth_mix=[Enth_st_mix;DeltaEnth_st_mix;DeltaEnth_disch_mix; ...
           DeltaEnth_ch_mix;DeltaEnth_in_bottom_mix;DeltaEnth_out_bottom_mix; ...
           DeltaEnth_flow_mix;DeltaEnth_loss_mix];
454
455 Ex=[Ex_st;DeltaEx_st;DeltaEx_disch;DeltaEx_ch;DeltaEx_in_bottom; ...
     DeltaEx_out_bottom;DeltaEx_flow;DeltaEx_loss;DeltaEx_int];
456 Ex_mix=[Ex_st_mix;DeltaEx_st_mix;DeltaEx_disch_mix;DeltaEx_ch_mix; ...
          DeltaEx_in_bottom_mix;DeltaEx_out_bottom_mix;DeltaEx_flow_mix; ...
          DeltaEx_loss_mix;DeltaEx_int_mix];
457
458 S=[S_st;DeltaS_st;DeltaS_disch;DeltaS_ch;DeltaS_in_bottom; ...
     DeltaS_out_bottom;DeltaS_flow;DeltaS_loss;DeltaS_int];
459 S_mix=[S_st_mix;DeltaS_st_mix;DeltaS_disch_mix;DeltaS_ch_mix; ...
         DeltaS_in_bottom_mix;DeltaS_out_bottom_mix;DeltaS_flow_mix; ...
         DeltaS_loss_mix;DeltaS_int_mix];
460
461 eta=[eta_A;eta_B;eta_C;eta_D;eta_E];
462 psi=[psi_A;psi_B;psi_C;psi_D;psi_E;psi_Ex_int;psi_Ex_int_loss];
463 end

```

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %%Falkner Stefan, 0826830                                     %%
3  %%Wien, 15.3.2014                                           %%
4  %%GET_I_IN                                                  %%
5  %%Funktion sucht jenen Wert im Vektor T der am naechsten bei T_in liegt %%
6  %%und gibt den Zeilenindex zurueck                          %%
7  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8  %%Falkner Stefan – Wien, 29.05.2014                          %%
9  %%Da beim kalorischen Mischen ein negativer Temp.grad. von %%
10 %%0.0001K/Knoten zugelassen wird, muessen die ersten beiden if %%
11 %%Anweisungen gecheckt werden, da sonst der falsche Index ermittelt %%
12 %%werden koennte.                                           %%
13 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14 %%                               Input                          %%
15 %%T...Temperaturverteilung in Grad C (Dim: n x 1)           %%
16 %%T_in...Temperatur in Grad C (Dim: 1 x 1)                  %%
17 %%                               Output                        %%
18 %%i_in...Zeilenindex der gesucht Temp. (Dim: 1 x 1)        %%
19 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
20
21 function [ i_in ] = GET_I_IN( T , T_in )
22
23 n=size(T);
24 n=n(1,1);
25 %tritt T_max od. T_min mehrfach auf, so wird der tiefste Index zurueckgeg.
26 [T_max,i_max]=max(T);
27 [T_min,i_min]=min(T);
28 if T_in>=T_max
29     i_in=n;
30 elseif T_in<=T_min
31     i_in=1;
32 else %T_min<T<T_max
33     dT = abs(T-T_in);
34     i_in = i_max;
35     %Grundsatzlich soll am hoechst moeglichem Knoten Beladen werden
36     dT_min = dT(i_max);
37     %sollte der kleinste Temp.unterschied mehrfach auftreten, dann wird die
38     %hoechste Schicht zurueckgegeben
39     for i=i_max:-1:i_min
40         if dT(i) < dT_min
41             dT_min = dT(i);
42             i_in=i;
43         end
44     end
45 end
46 end

```



```

8 %%Voraussetzungen: uniform Grid %%
9 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10 %% Input %%
11 %%Enth_exakt...exakter Wert in J (Dim: 1x1) %%
12 %%T_min...untere Grenze des Startintervalls in Grad C (Dim: 1x1) %%
13 %%T_max...obere Grenze des Startintervalls in Grad C (Dim: 1x1) %%
14 %%V...Gesamtvolumen des Speichers in m^3 (Dim: 1x1) %%
15 %% Output %%
16 %%T_m...Temp.des vollstaendig durchmischtem Speichers in GradC(Dim: 1x1)%%
17 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
18
19
20 function [ T_m ] = T_M(Enth_exakt,T_min,T_max,V)
21
22 i_iter=0; %Zaehler fuer Iterationen
23 n_iter=1e4; %Obergrenze an Iterationen
24 T_m=0;
25 error=1;
26 while error>1e-6 && i_iter<n_iter
27     T_m=(T_min+T_max)*0.5;
28     Enth_m=V*density(T_m)*T_m*heat_capacity_int(T_m);
29     error=abs(Enth_exakt-Enth_m)/Enth_exakt;
30     if Enth_m > Enth_exakt
31         T_max=T_m;
32     elseif Enth_m < Enth_exakt
33         T_min=T_m;
34     else
35         error=0;
36     end
37     i_iter=i_iter+1;
38 end
39 end

```

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %%Falkner Stefan, 0826830 %%
3 %%Wien, 16.4.2014 %%
4 %%EPSILON %%
5 %%Funktion liefert den effektiven Waermeleitungsfaktor fuer die %%
6 %%uebergebenen Einstroembedingungen (Eddy-Conductivity) %%
7 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
8 %%Falkner Stefan, 8.5.2014 %%
9 %%Zusaetzlich zum hyperbolischem Ansatz kann jetzt auch ein %%
10 %%exponentieller verwendet werden %%
11 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12 %%Bemerkung: %%
13 %%(a) uebergibt man m_up und m_down gleich 0, so liefert die Fkt. %%

```

```

14 %%fuer alle Stellen e=1. D.h.: (1) Es sind keine Massenstroeme vorhanden%%
15 %%bzw. (2) es kann eine turbulenzfreie Einstroemung fuer die Berechnung %%
16 %%vom Benutzer erzwungen werden. %%
17 %%(b) Nur fuer gleiche Knotengroessen %%
18 %%(c) wird phi≠0 uebergeben, so wird der exponentielle Ansatz verwendet%%
19 %%andernfalls der hyperbolische %%
20 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
21 %% Input %%
22 %%epsilon_in = a*(Re/Ri)^b... Waermeleitungsfaktor am Eintritt(Dim: 1x1)%%
23 %%n...Anzahl der Knoten %%
24 %%m_up...Massenstrom nach oben in kg/s (Dim: 1x1) %%
25 %%m_down...Massenstrom nach unten in kg/s (Dim: 1x1) %%
26 %%dz...Knotenhoehe in m (Dim: 1x1) %%
27 %%phi...exponentieller Ansatz (Dim: 1x1) %%
28 %% Output %%
29 %%e...Faktor fuer effektive Waermeleitung unter gegebene %%
30 %%Einstroembedingungen (Dim: (n-1)x1) %%
31 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
32
33
34 function [ e ] = EPSILON( epsilon_in,n,m_up,m_down,dz,phi )
35
36 e = zeros(n-1,1);
37 %sollte kein Massenstrom vorhanden sein nimmt e an jeder Stelle den Wert 1
38 %an
39 e(:)=1;
40
41 if m_up≠0 || m_down≠0 %wird Speicher durchstroemt
42     if phi==0 %hyperbolischer Ansatz
43         A=(epsilon_in-1)/(1-1/n);
44         B=epsilon_in-A;
45         for i=1:n-1;
46             e(i)=A/i+B;
47         end
48     else %exponentieller Ansatz
49         for i=1:n-1
50             e(i)=epsilon_in*exp(-phi*dz*i);
51         end
52     end
53 %sollte der Einlass unten sein (m_up≠0) sein, dann stimmt das schon so
54 %sollte der Einlas oben sein, wird Vektor gespiegelt
55 if m_down≠0
56     e=flipud(e);
57 end
58 end
59
60 end

```



```

15 function [ z ] = GET_Z( dz,n )
16
17 m = size(dz);
18 m = m(1,1);          %Anzahl der Eintraege in dz
19
20 z=zeros(n,1);
21 if m==1
22     %konstante Knotengroesse
23     for i=1:n
24         z(i)=dz*(i-0.5);
25     end
26 elseif m==3 && m~=n
27     %innere Knoten mit konstanter Groesse
28     z(1)=dz(1)/2;
29     z(2)=dz(1)+dz(2)/2;
30     for i=3:n-1
31         z(i)=z(i-1)+dz(2);
32     end
33     z(n)=z(n-1)+(dz(2)+dz(3))/2;
34 elseif n==m
35     %unterschiedliche Knotengroessen
36     z(1)=dz(1)/2;
37     for i=2:n
38         z(i)=z(i-1)+(dz(i-1)+dz(i))/2;
39     end
40 end

```

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %%Falkner Stefan, 0826830                                           %%
3  %%Wien, 14.3.2014                                                  %%
4  %%Funktion loest ein DGL-System der Gestalt                         %%
5  %%a(i,1)*(dT/dt)(i)=a(i,2)*T(i)+a(i,3)*T(i+1)+a(i,4)*T(i-1)+a(i,5) %%
6  %%fuer i = 1...n                                                  %%
7  %%mit Vorwaerts-Euler                                             %%
8  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9  %%%                                                                %%
10 %%[a1,a2,a3,a4,a5]...Spaltenvektoren der Koeffizienten (Dim: n x 5) %%
11 %%T0...Loesung zu Beginn des Zeitschritts (Dim: n x 1)           %%
12 %%dt...Dauer des Zeitschritts (Dim: 1 x 1)                       %%
13 %%%                                                                %%
14 %%T1...Loesung zum Ende des Zeitschritts (Dim: n x 1)           %%
15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16
17 function [ T1 ] = EULER( a , T0, dt )
18
19 n=size(a);

```

```

20 n=n(1,1);
21
22 T1=zeros(n,1);
23
24 T1(1)=(a(1,2)*T0(1)+a(1,3)*T0(2)+a(1,5))*dt/a(1,1)+T0(1);
25 for i=2:n-1
26     T1(i)=(a(i,2)*T0(i)+a(i,3)*T0(i+1)+a(i,4)*T0(i-1)+a(i,5))*dt/a(i,1)+T0(i);
27 end
28 T1(n)=(a(n,2)*T0(n)+a(n,4)*T0(n-1)+a(n,5))*dt/a(n,1)+T0(n);
29 end

```

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %%Falkner Stefan, 0826830                                     %%
3  %%Wien, 14.3.2014                                           %%
4  %%Funktion loest ein DGL-System der Gestalt                  %%
5  %%a(i,1)*(dT/dt)(i)=a(i,2)*T(i)+a(i,3)*T(i+1)+a(i,4)*T(i-1)+a(i,5) %%
6  %%fuer i = 1...n                                           %%
7  %%mit der Fully-Implicit-Methode                            %%
8  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9  %%%                                                         Input %%
10 %%[a1,a2,a3,a4,a5]...Spaltenvektoren der Koeffizienten (Dim: n x 5) %%
11 %%T0...Loesung zu Beginn des Zeitschritts (Dim: n x 1)    %%
12 %%dt...Dauer des Zeitschritts (Dim: 1 x 1)                %%
13 %%%                                                         Output %%
14 %%T1...Loesung zum Ende des Zeitschritts (Dim: n x 1)    %%
15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16
17 function [ T1 ] = FULLY_IMPLICIT( a, T0, dt )
18
19 T1 = TDMA(a(:,1)/dt-a(:,2), a(:,3), a(:,4), a(:,5)+(a(:,1)/dt).*T0);
20
21 end

```

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %%Falkner Stefan, 0826830                                     %%
3  %%Wien, 14.3.2014                                           %%
4  %%Funktion loest ein DGL-System der Gestalt                  %%
5  %%a(i,1)*(dT/dt)(i)=a(i,2)*T(i)+a(i,3)*T(i+1)+a(i,4)*T(i-1)+a(i,5) %%
6  %%fuer i = 1...n                                           %%
7  %%mit dem Crank-Nicolson-Verfahren                          %%
8  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9  %%%                                                         Input %%
10 %%[a1,a2,a3,a4,a5]...Spaltenvektoren der Koeffizienten (Dim: n x 5) %%
11 %%T0...Loesung zu Beginn des Zeitschritts (Dim: n x 1)    %%

```

```

12 %%dt...Dauer des Zeitschritts (Dim: 1 x 1)                %%
13 %%                                                    %%      Output                %%
14 %%T1...Loesung zum Ende des Zeitschritts (Dim: n x 1)   %%
15 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16
17 function [ T1 ] = CRANK_NICOLSON( a, T0, dt )
18
19 n=size(a);
20 n=n(1,1);
21
22 b = zeros(n,1);
23 b(1)=a(1,3)/2*T0(2)+(a(1,2)/2+a(1,1)/dt)*T0(1);
24 for i=2:n-1
25     b(i)=a(i,3)/2*T0(i+1)+a(i,4)/2*T0(i-1)+(a(i,2)/2+a(i,1)/dt)*T0(i);
26 end
27 b(n)=a(n,4)/2*T0(n-1)+(a(n,2)/2+a(n,1)/dt)*T0(n);
28
29 T1 = TDMA(a(:,1)/dt-a(:,2)/2, a(:,3)/2, a(:,4)/2, a(:,5)+b);
30
31 end

```

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %%Falkner Stefan, 0826830                                %%
3 %%Wien, 22.6.2013                                       %%
4 %%Funktion loest ein tridiagonales Glg.system der Gestalt %%
5 %%a(i)*x(i)=b(i)*x(i+1)+c(i)*x(i-1)+d(i) fuer i = 1...n %%
6 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7 %%                                                    %%      Input                %%
8 %%a,b,c,d...Spaltenvektoren der Koeffizienten (Dim: n x 1) %%
9 %%                                                    %%      Output                %%
10 %%x...Loesungsvektor (Dim: n x 1)                       %%
11 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12
13 function [ x ] = TDMA( a, b, c, d )
14
15 n=size(a);
16 n=n(1,1);
17
18 P = zeros(n,1);
19 Q = zeros(n,1);
20 x = zeros(n,1);
21
22 P(1) = b(1)/a(1);
23 Q(1) = d(1)/a(1);
24
25 for i=2:n

```

```

26     P(i) = b(i)/(a(i)-c(i)*P(i-1));
27     Q(i) = (d(i)+c(i)*Q(i-1))/(a(i)-c(i)*P(i-1));
28 end;
29
30 x(n) = Q(n);
31 for i=(n-1):-1:1
32     x(i)=P(i)*x(i+1)+Q(i);
33 end;
34 end

```

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %%Falkner Stefan, 0826830                                     %%
3  %%Wien, 28.5.2014                                           %%
4  %%STEP_FILO_modfsl (First In Last Out modified for simulink) %%
5  %%Level-2 MATLAB S-Function, zum Einbinden des Modells FILO in Simulink %%
6  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7  %%Wichtig: fixed TimeStep Solver verwenden                 %%
8  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9
10
11
12 function STEP_FILO_modfsl(block)
13
14 setup(block);
15 %end function
16
17
18
19 %-----
20 %Initialisierung des Blocks
21 %-----
22 function setup(block)
23
24 %Anzahl der Eingangs- und Ausgangsports
25 block.NumInputPorts = 8;
26 %m_up,m_down,T_up,T_down,T_amb,T0,T0_out_1,T0_out_n
27 block.NumOutputPorts = 3;
28 %T1,T1_out_1,T1_out_n
29
30 %Anzahl der Parameter
31 block.NumDialogPrms = 13;
32 %H,Dia,k_rad,k_ax,lambda_eff,n,d_in,a,b,phi,Einst,T_ini,f_crit
33
34 %Dimensionen der Eingangsports
35 %Flag DirectFeedthrough kennzeichnet, ob ein Eingang direkt an einem
36 %Ausgang auftritt, d.h. y=f(u)

```

```
37 block.InputPort(1).Dimensions      = 1;
38 block.InputPort(1).SamplingMode = 'Sample';
39 block.InputPort(1).DirectFeedthrough = false;
40 block.InputPort(2).Dimensions      = 1;
41 block.InputPort(2).SamplingMode = 'Sample';
42 block.InputPort(2).DirectFeedthrough = false;
43 block.InputPort(3).Dimensions      = 1;
44 block.InputPort(3).SamplingMode = 'Sample';
45 block.InputPort(3).DirectFeedthrough = false;
46 block.InputPort(4).Dimensions      = 1;
47 block.InputPort(4).SamplingMode = 'Sample';
48 block.InputPort(4).DirectFeedthrough = false;
49 block.InputPort(5).Dimensions      = 1;
50 block.InputPort(5).SamplingMode = 'Sample';
51 block.InputPort(5).DirectFeedthrough = false;
52 block.InputPort(6).Dimensions      = block.DialogPrm(6).Data;
53 block.InputPort(6).SamplingMode = 'Sample';
54 block.InputPort(6).DirectFeedthrough = false;
55 block.InputPort(7).Dimensions      = 1;
56 block.InputPort(7).SamplingMode = 'Sample';
57 block.InputPort(7).DirectFeedthrough = false;
58 block.InputPort(8).Dimensions      = 1;
59 block.InputPort(8).SamplingMode = 'Sample';
60 block.InputPort(8).DirectFeedthrough = false;
61
62 %Dimensionen der Ausgangsports
63 block.OutputPort(1).Dimensions      = block.DialogPrm(6).Data;
64 block.OutputPort(1).SamplingMode = 'Sample';
65 block.OutputPort(2).Dimensions      = 1;
66 block.OutputPort(2).SamplingMode = 'Sample';
67 block.OutputPort(3).Dimensions      = 1;
68 block.OutputPort(3).SamplingMode = 'Sample';
69
70 %Einstellen der Abtastzeit: [0 0] wird verwendet fuer die
71 %zeitkontinuierliche Simulation.
72 block.SampleTimes = [0 0];
73
74 %Registrieren der einzelnen Methoden
75 %InitializeConditions ... Initialisierung
76 %Outputs ... Berechnung der Ausgaenge
77 %Terminate ... Konsistentes Beenden der Simulation
78
79 block.RegBlockMethod('InitializeConditions', @InitConditions);
80 block.RegBlockMethod('Outputs', @Output);
81 block.RegBlockMethod('Terminate', @Terminate);
82 %end function
83
```

```
84
85
86
87
88 %-----
89 %Setzen der Anfangstemperaturen
90 %-----
91 function InitConditions(block)
92 block.InputPort(6).Data(:)=block.DialogPrm(12).Data;
93 block.InputPort(7).Data=block.DialogPrm(12).Data;
94 block.InputPort(8).Data=block.DialogPrm(12).Data;
95 %end function
96
97
98
99 %-----
100 %Berechnen der Ausgaenge
101 %-----
102 function Output(block)
103 %H=Storage(1),Dia=Storage(2),k_rad=Storage(3),k_ax=Storage(4), ...
    lambda_eff=Storage(5)
104 %n=Geometry_node(1),V_node=Geometry_node(2),dz=Geometry_node(3), ...
    A_ax=Geometry_node(4),A_rad=Geometry_node(5)
105 %d_in=Inlet(1),a=Inlet(2),b=Inlet(3),phi=Inlet(4),Inlet(5)==1
106
107 %Parameter
108 Storage=[block.DialogPrm(1).Data,block.DialogPrm(2).Data, ...
    block.DialogPrm(3).Data,block.DialogPrm(4).Data,block.DialogPrm(5).Data]';
109 Inlet=[block.DialogPrm(7).Data,block.DialogPrm(8).Data, ...
    block.DialogPrm(9).Data,block.DialogPrm(10).Data, ...
    block.DialogPrm(11).Data]';
110 Geometry_node=GEOMETRY(block.DialogPrm(6).Data,block.DialogPrm(1).Data, ...
    block.DialogPrm(2).Data);
111 f_crit=block.DialogPrm(13).Data;
112
113 %Inputs
114 m_up=block.InputPort(1).Data;
115 m_down=block.InputPort(2).Data;
116 T_up=block.InputPort(3).Data;
117 T_down=block.InputPort(4).Data;
118 T_amb=block.InputPort(5).Data;
119 T0=block.InputPort(6).Data;
120 T0_out_1=block.InputPort(7).Data;
121 T0_out_n=block.InputPort(8).Data;
122
123 %warning
124 if (m_up~=0 && m_down~=0)
```

```
125     error('m_up und/oder m_down muessen/muss gleich 0 sein');
126 end
127
128 if strcmp(get_param(bdroot, 'SolverType'), 'Variable-step') % //Returns ...
    'Variable-step' or 'Fixed-step'
129     error('!!Fixed-step-Solver verwenden!!');
130 end
131
132 %globale Variable von Solver - FixedStep
133 dt = str2double(get_param(bdroot, 'FixedStep'));
134
135 %Zeitschritthandling
136 %kritischer Zeitschritt: Courant-Zahl=1=v*dt/dz
137 if m_up≠0
138     dt_crit=density(T_up)*Geometry_node(4)*Geometry_node(3)/m_up;
139     %Austrittstemp. unten bleibt gleich
140     T1_out_1=T0_out_1;
141 elseif m_down≠0
142     dt_crit=density(T_down)*Geometry_node(4)*Geometry_node(3)/m_down;
143     %Austrittstemp. oben bleibt gleich
144     T1_out_n=T0_out_n;
145 else
146     dt_crit=10^10; %-->infinity
147     %Austrittstemp. oben u. unten bleiben gleich
148     T1_out_1=T0_out_1;
149     T1_out_n=T0_out_n;
150 end
151 dt_max=dt_crit*f_crit;
152 %sollte der vom Benutzer vorgegebene Zeitschritt dt groesser als der
153 %maximal moegliche dt_max sein, so muessen funktionsintern mehrere
154 %Zeitschritte mit einer kleineren Zeitschrittdauer geloest werden, bis die
155 %gewuenscht Zeitschrittlaege dt erreicht ist.
156 if dt>ceil(dt_max)
157     nt=ceil(dt/dt_max); %Anzahl der noetigen Zeitschritte (aufgerundet)
158     dt = dt/nt; %neue interne Zeitschrittlaege
159 else
160     nt=1;
161 end
162
163 %warning
164 if nt≠1
165     warning(['Zeitschrittweite zu gross! Es werden ', num2str(nt), ' ...
        Schritte mit einer Zeitschrittweite von ', num2str(dt), ' s ...
        berechnet.']);
166 end
167
168 %interne Zeitschritte
```

```

169 for j=1:nt
170     if Inlet(5)==1
171         %Einstroemverhaeltnisse werden beruecksichtigt
172         if m_up≠0
173             v_in = 4*m_up/(Inlet(1)^2*pi()*density(T_up));
174             nu = kinematic_viscosity(T_up);
175             Ri = 9.81*Storage(1)*0.5*(thermal_expansion(T_up) ...
176                 +thermal_expansion(T0(1)))*abs(T_up-T0(1))/v_in^2;
177             Re = v_in*Inlet(1)/nu;
178         elseif m_down≠0
179             v_in = 4*m_down/(Inlet(1)^2*pi()*density(T_down));
180             nu = kinematic_viscosity(T_down);
181             Ri = 9.81*Storage(1)*0.5*(thermal_expansion(T_down) ...
182                 +thermal_expansion(T0(Geometry_node(1)))) ...
183                 *abs(T_down-T0(Geometry_node(1)))/v_in^2;
184             Re = v_in*Inlet(1)/nu;
185         else
186             Ri = 10^10; %Ri-->infinity
187             Re = 0;
188         end
189         epsilon_eff=EPSILON(Inlet(2)*(Re/Ri)^Inlet(3),Geometry_node(1), ...
190             m_up,m_down,Geometry_node(3),Inlet(4));
191     else
192         %Einstroemverhaeltnisse werden nicht beruecksichtigt
193         epsilon_eff=EPSILON(0,Geometry_node(1),0,0,Geometry_node(3),0);
194     end
195     %Koeffizienten der Energiebilanz
196     koeff = FILO( ...
197         Geometry_node,Storage(3),Storage(4),Storage(5),m_up,m_down,T_up, ...
198         T_down,T_amb,T0,epsilon_eff );
199     %Zeitschritt
200     T0 = CRANK_NICOLSON(koeff,T0,dt);
201     %Abbau inverser Schichtung
202     T0 = CALORIC_MIXING(T0);
203 end
204 T1=T0;
205
206 %verbleibende Austrittstemp. falls Massenstroeme vorhanden
207 if m_up≠0
208     T1_out_n=0.5*(T0(Geometry_node(1))+T1(Geometry_node(1)));
209 elseif m_down≠0
210     T1_out_1=0.5*(T0(1)+T1(1));
211 end
212
213 %Outputs
214 block.OutputPort(1).Data = T1;

```

```

210 block.OutputPort(2).Data = T1_out_1;
211 block.OutputPort(3).Data = T1_out_n;
212 %end function
213
214
215
216 %-----
217 %Zeitschritt Ende
218 %-----
219 %Die function Terminate wird hier nicht verwendet,
220 %muss aber vorhanden sein!
221 function Terminate(block)

```

```

1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %%Falkner Stefan, 0826830                                           %%
3  %%Wien, 28.5.2014                                                  %%
4  %%STEP_VAR_IN_modfsl (VARIABLE INlet modified for simulink)      %%
5  %%Level-2 MATLAB S-Function, zum Einbinden des Modells VAR_IN in Simulink%%
6  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7  %%Wichtig: fixed TimeStep Solver verwenden                        %%
8  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
9
10
11
12 function STEP_VAR_IN_modfsl(block)
13
14 setup(block);
15 %end function
16
17
18
19 %-----
20 %Initialisierung des Blocks
21 %-----
22 function setup(block)
23
24 %Anzahl der Eingangs- und Ausgangsports
25 block.NumInputPorts = 8;
26 %m_up,m_down,T_up,T_down,T_amb,T0,T0_out_1,T0_out_n
27 block.NumOutputPorts = 3;
28 %T1,T1_out_1,T1_out_n
29
30 %Anzahl der Parameter
31 block.NumDialogPrms = 8;
32 %H,Dia,k_rad,k_ax,lambda_eff,n,T_ini,f_crit
33

```

```
34 %Dimensionen der Eingangsports
35 %Flag DirectFeedthrough kennzeichnet, ob ein Eingang direkt an einem
36 %Ausgang auftritt, d.h.  $y=f(u)$ 
37 block.InputPort(1).Dimensions      = 1;
38 block.InputPort(1).SamplingMode = 'Sample';
39 block.InputPort(1).DirectFeedthrough = false;
40 block.InputPort(2).Dimensions      = 1;
41 block.InputPort(2).SamplingMode = 'Sample';
42 block.InputPort(2).DirectFeedthrough = false;
43 block.InputPort(3).Dimensions      = 1;
44 block.InputPort(3).SamplingMode = 'Sample';
45 block.InputPort(3).DirectFeedthrough = false;
46 block.InputPort(4).Dimensions      = 1;
47 block.InputPort(4).SamplingMode = 'Sample';
48 block.InputPort(4).DirectFeedthrough = false;
49 block.InputPort(5).Dimensions      = 1;
50 block.InputPort(5).SamplingMode = 'Sample';
51 block.InputPort(5).DirectFeedthrough = false;
52 block.InputPort(6).Dimensions      = block.DialogPrm(6).Data;
53 block.InputPort(6).SamplingMode = 'Sample';
54 block.InputPort(6).DirectFeedthrough = false;
55 block.InputPort(7).Dimensions      = 1;
56 block.InputPort(7).SamplingMode = 'Sample';
57 block.InputPort(7).DirectFeedthrough = false;
58 block.InputPort(8).Dimensions      = 1;
59 block.InputPort(8).SamplingMode = 'Sample';
60 block.InputPort(8).DirectFeedthrough = false;
61
62 %Dimensionen der Ausgangsports
63 block.OutputPort(1).Dimensions      = block.DialogPrm(6).Data;
64 block.OutputPort(1).SamplingMode = 'Sample';
65 block.OutputPort(2).Dimensions      = 1;
66 block.OutputPort(2).SamplingMode = 'Sample';
67 block.OutputPort(3).Dimensions      = 1;
68 block.OutputPort(3).SamplingMode = 'Sample';
69
70 %Einstellen der Abtastzeit: [0 0] wird verwendet fuer die
71 %zeitkontinuierliche Simulation.
72 block.SampleTimes = [0 0];
73
74 %Registrieren der einzelnen Methoden
75 %InitializeConditions ... Initialisierung
76 %Outputs ... Berechnung der Ausgaenge
77 %Terminate ... Konsistentes Beenden der Simulation
78
79 block.RegBlockMethod('InitializeConditions', @InitConditions);
80 block.RegBlockMethod('Outputs', @Output);
```

```

81 block.RegBlockMethod('Terminate', @Terminate);
82 %end function
83
84
85
86
87
88 %-----
89 %Setzen der Anfangstemperaturen
90 %-----
91 function InitConditions(block)
92 block.InputPort(6).Data(:)=block.DialogPrm(7).Data;
93 block.InputPort(7).Data=block.DialogPrm(7).Data;
94 block.InputPort(8).Data=block.DialogPrm(7).Data;
95 %end function
96
97
98
99 %-----
100 %Berechnen der Ausgaenge
101 %-----
102 function Output(block)
103 %H=Storage(1),Dia=Storage(2),k_rad=Storage(3),k_ax=Storage(4), ...
    lambda_eff=Storage(5)
104 %n=Geometry_node(1),V_node=Geometry_node(2),dz=Geometry_node(3), ...
    A_ax=Geometry_node(4),A_rad=Geometry_node(5)
105
106
107 %Parameter
108 Storage=[block.DialogPrm(1).Data,block.DialogPrm(2).Data, ...
    block.DialogPrm(3).Data,block.DialogPrm(4).Data,block.DialogPrm(5).Data]';
109 Geometry_node=GEOMETRY(block.DialogPrm(6).Data,block.DialogPrm(1).Data, ...
    block.DialogPrm(2).Data);
110 f_crit=block.DialogPrm(8).Data;
111
112 %Inputs
113 m_up=block.InputPort(1).Data;
114 m_down=block.InputPort(2).Data;
115 T_up=block.InputPort(3).Data;
116 T_down=block.InputPort(4).Data;
117 T_amb=block.InputPort(5).Data;
118 T0=block.InputPort(6).Data;
119 T0_out_1=block.InputPort(7).Data;
120 T0_out_n=block.InputPort(8).Data;
121
122 %warning
123 if (m_up~=0 && m_down~=0)

```

```
124     error('m_up und/oder m_down muessen/muss gleich 0 sein');
125 end
126
127 if strcmp(get_param(bdroot, 'SolverType'), 'Variable-step') % //Returns ...
    'Variable-step' or 'Fixed-step'
128     error('!!Fixed-step-Solver verwenden!!');
129 end
130
131 %globale Variable von Solver - FixedStep
132 dt = str2double(get_param(bdroot, 'FixedStep'));
133
134 %Zeitschritthandling
135 %kritischer Zeitschritt: Courant-Zahl=1=v*dt/dz
136 if m_up≠0
137     dt_crit=density(T_up)*Geometry_node(4)*Geometry_node(3)/m_up;
138     %Austrittstemp. unten bleibt gleich
139     T1_out_1=T0_out_1;
140 elseif m_down≠0
141     dt_crit=density(T_down)*Geometry_node(4)*Geometry_node(3)/m_down;
142     %Austrittstemp. oben bleibt gleich
143     T1_out_n=T0_out_n;
144 else
145     dt_crit=10^10; %-->infinity
146     %Austrittstemp. oben u. unten bleiben gleich
147     T1_out_1=T0_out_1;
148     T1_out_n=T0_out_n;
149 end
150 dt_max=dt_crit*f_crit;
151 %sollte der vom Benutzer vorgegebene Zeitschritt dt groesser als der
152 %maximal moegliche dt_max sein, so muessen funktionsintern mehrere
153 %Zeitschritte mit einer kleineren Zeitschrittdauer geloest werden, bis die
154 %gewuenscht Zeitschrittlaenge dt erreicht ist.
155 if dt>ceil(dt_max)
156     nt=ceil(dt/dt_max); %Anzahl der noetigen Zeitschritte (aufgerundet)
157     dt = dt/nt; %neue interne Zeitschrittlaenge
158 else
159     nt=1;
160 end
161
162 %warning
163 if nt≠1
164     warning(['Zeitschrittweite zu gross! Es werden ', num2str(nt), ' ...
        Schritte mit einer Zeitschrittweite von ', num2str(dt), ' s ...
        berechnet.']);
165 end
166
167 %interne Zeitschritte
```

```
168 for j=1:nt
169     %Suche Schicht mit geringster Temp.abweichung zu T_down
170     i_in = GET_I_IN(T0,T_down);
171     %Koeffizienten der Energiebilanz
172     koeff = ...
            VAR_IN(Geometry_node,Storage(3),Storage(4),Storage(5),m_up,m_down, ...
            T_up,T_down,T_amb,T0,i_in);
173     %Zeitschritt
174     T0 = CRANK_NICOLSON(koeff,T0,dt);
175     %Abbau inverser Schichtung
176     T0 = CALORIC_MIXING(T0);
177 end
178 T1=T0;
179
180 %verbleibende Austrittstemp. falls Massenstroeme vorhanden
181 if m_up≠0
182     T1_out_n=0.5*(T0(Geometry_node(1))+T1(Geometry_node(1)));
183 elseif m_down≠0
184     T1_out_1=0.5*(T0(1)+T1(1));
185 end
186
187
188 %Outputs
189 block.OutputPort(1).Data = T1;
190 block.OutputPort(2).Data = T1_out_1;
191 block.OutputPort(3).Data = T1_out_n;
192 %end function
193
194
195
196 %-----
197 %Zeitschritt Ende
198 %-----
199 %Die function Terminate wird hier nicht verwendet,
200 %muss aber vorhanden sein!
201 function Terminate(block)
```