

**Thermodynamische Berechnungen mit
MATLAB – Entwicklung einer Programm-
sammlung**

Von der Gemeinsamen Naturwissenschaftlichen
Fakultät der Technischen Universität
Carolo-Wilhelmina
zu Braunschweig

zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

genehmigte

D i s s e r t a t i o n

von
Olaf Klein
aus Schloß Neuhaus

1. Referent: Prof. Dr. A. Löwe
2. Referent: Prof. Dr. A. Schumpe
eingereicht am: 8.2.2001
mündliche Prüfung am: 10.5.2001
Druckjahr: 2001

Danksagung

Die vorliegende Arbeit wurde in der Zeit von Juni 1997 bis Februar 2001 am Institut für Technische Chemie, Abteilung Reaktionstechnik, der Technischen Universität Carolo-Wilhelmina zu Braunschweig angefertigt.

Allen, die zum Gelingen dieser Arbeit beigetragen haben, danke ich herzlich, insbesondere

meinem Doktorvater, Herrn Univ.-Prof. Dr. A. Löwe, für die Anregung zu dieser Arbeit, seine Betreuung und seine Geduld,

Herrn Univ.-Prof. Dr. A. Schumpe für die Übernahme des Korreferats,

allen Mitarbeitern der Arbeitsgruppe Prof. Dr. Löwe und allen Angehörigen des Instituts für Technische Chemie für die ständige Diskussions- und Hilfsbereitschaft.

Zusammenfassung

Die schnell fortschreitende Entwicklung immer leistungsfähigerer Computer ermöglichte in den letzten Jahren einen verstärkten Einsatz rechnergestützter Modellierungen und Simulationen in der chemischen Technik [64]. Mögliche Einsatzgebiete sind unter anderem die Planung und Auslegung von chemischen Reaktoren oder die Verfahrensoptimierung. Vorteile der Anwendung von Simulationsverfahren gegenüber dem alleinigen Einsatz experimenteller Verfahren ist eine Aufwandsoptimierung und damit verbunden eine mögliche Kostenreduktion. Die Weiterentwicklung der rechnergestützten Prozeßsimulation als leistungsfähiges Werkzeug ist unter anderem vom Fortschritt der Computertechnik und der Entwicklung bzw. Verbesserung mathematischer Methoden abhängig. Hier sind zum Beispiel Algorithmen zur Behandlung von Differentialgleichungen zu nennen. Daneben sind zuverlässige Verfahren zur Abschätzung thermodynamischer Basisdaten von Bedeutung um die Simulation in dem umrissenen Themengebiet zu einem erfolgversprechenden Werkzeug zu machen.

Ein Programmsystem, das bereits erfolgreich auf dem Gebiet der Simulation eingesetzt wurde, ist MATLAB der Firma MATHWOKRS[®] [42, 49, 51, 64, 71]. Es umfaßt in seiner Grundversion verschiedene Algorithmen für numerische Hochleistungsrechnungen und Funktionen, die weit über den Umfang einer normalen Programmiersprache hinausgehen. Die MATLAB Grundversion kann weiterhin noch mit verschiedenen Funktionsbibliotheken - sogenannten *toolboxes* - erweitert werden. Jedoch stellt MATLAB keine explizite Funktionsbibliothek für Berechnungen auf dem Gebiet der chemischen Technik zur Verfügung [67].

Ziel der vorliegenden Arbeit ist es daher, eine Programmsammlung zu erstellen, die den leistungsfähigen numerischen Verfahren von MATLAB Funktionen an die Seite stellt, die es erlauben, thermodynamische Basisdaten für chemisch-technische Simulationen zu berechnen. Neben der Realisierung der angesprochenen Program-

me ist die Untersuchung von MATLAB, bezüglich der Eignung für die gestellten Aufgaben, integraler Bestandteil der Arbeit.

Im Rahmen der vorliegenden Dissertation wurden Programme zur Berechnung von thermodynamischen Eigenschaften idealer und realer Gase, zur Betrachtung von Dampf-Flüssig-Gleichgewichten und zur Abschätzung von molekularen Transportkoeffizienten implementiert. Darüber hinaus sind unter Anwendung der in MATLAB gegebenen numerischen Möglichkeiten Methoden zur Berechnung von chemischen Simultangleichgewichten programmiert worden. Des weiteren ist ein System zur benutzerfreundlichen Verwaltung von Daten, in Form eines leicht zugänglichen Datensatzes, unter MATLAB erstellt worden.

Dabei hat sich MATLAB für die Behandlung obiger Zusammenhänge als gut geeignet erwiesen. Diese gute Eignung resultiert aus der Kombination einer hoch flexiblen, mathematisch orientierten Programmiersprache mit umfassenden Funktionsbibliotheken für technisch-wissenschaftliche Hochleistungsrechnungen. Dabei ist hervorzuheben, daß die *toolboxes* konzeptionell in das Softwaresystem integriert, damit schnell verfügbar und problemlos in bestehende Strukturen einzubeziehen sind.

1	<u>EINLEITUNG IN DIE PROGRAMMSAMMLUNG</u>	12
1.1	VORSTELLUNG VON MATLAB	12
1.2	AUFBAU DER PROGRAMMSAMMLUNG	13
2	<u>DATENBASIS</u>	18
2.1	DATENORGANISATION	18
2.2	INHALT DES DATENSATZES	22
2.3	DATEN AUS DER ON-LINE DATENBANK DETHERM	26
2.4	SUCHFUNKTIONEN	30
3	<u>THERMODYNAMISCHE EIGENSCHAFTEN IDEALER GASE</u>	34
3.1	$\Delta_R H$, $\Delta_R G$ UND $\Delta_R S$	35
3.2	TEMPERATUREINFLUß AUF H, G UND S	45
3.3	TEMPERATUREINFLUß AUF DIE WÄRMEKAPAZITÄT IDEALER GASE	53
4	<u>THERMODYNAMISCHE EIGENSCHAFTEN REALER GASE</u>	55
4.1	ZUSTANDSGLEICHUNGEN	55
4.2	ABWEICHUNGSFUNKTIONEN	72
4.3	DRUCKEINFLUß AUF H, G UND S	76
4.4	FUGAZITÄTSKOEFFIZIENTEN	78
4.5	WÄRMEKAPAZITÄT REALER GASE	81
5	<u>CHEMISCHE GLEICHGEWICHTE</u>	85
5.1	GLEICHGEWICHTSKONSTANTEN	85
5.2	STÖCHIOMETRISCHE BERECHNUNGEN	88
5.2.1	SCHLÜSSELKOMponentEN	88
5.2.2	ELEMENT-SPEZIES-MATRIX	91
5.2.3	SCHLÜSSELREAKTIONEN	92
5.3	BERECHNUNG VON SIMULTANGLEICHGEWICHTEN	98
5.3.1	REAKTIONSFORTSCHRITT	99
5.3.2	LÖSEN VON NICHT LINEAREN ALGEBRAISCHEN GLEICHUNGEN	101
5.3.3	MINIMIERUNG DER GIBBSCHEN ENTHALPIE	107

5.3.4	BERECHNUNG MIT MASSENWIRKUNGSKINETIK	109
6	DAMPF-FLÜSSIG-GLEICHGEWICHTE	123
7	MOLEKULARE TRANSPORTVORGÄNGE	129
7.1	THERMISCHE LEITFÄHIGKEIT	129
7.1.1	METHODE NACH ELY-HANLEY	129
7.1.2	METHODE NACH CHUNG	130
7.1.3	ABSCHÄTZUNG DER THERMISCHEN LEITFÄHIGKEIT VON FLÜSSIGKEITEN	131
7.2	VISSKOSITÄT	136
7.2.1	METHODE NACH LUCAS	137
7.2.2	METHODE NACH CHUNG	138
7.2.3	ABSCHÄTZUNG VON FLÜSSIGKEITSVISSKOSITÄTEN	142
7.3	DIFFUSIONSKOEFFIZIENTEN	147
7.3.1	METHODE NACH CHAPMAN UND ENSKOG	147
7.3.2	METHODE NACH WILKE UND LEE	149
7.3.3	METHODE NACH FULLER, SCHEITLER UND GIDDINGS	150
8	DISKUSSION	153
8.1	DISKUSSION DER PROGRAMMSAMMLUNG	153
8.2	BEURTEILUNG DER EIGNUNG VON MATLAB FÜR DIE IMPLEMENTIERUNG DER PROGRAMMSAMMLUNG	155
9	AUSBLICK	158
10	VERZEICHNIS DER WICHTIGSTEN FUNKTIONEN	159
11	SYMBOLVERZEICHNIS	163
12	ANHANG	168

12.1	ANHANG A: THERMISCHE LEITFÄHIGKEIT NACH ELY-HANLEY	168
12.2	ANHANG B: THERMISCHE LEITFÄHIGKEIT NACH CHUNG	174
12.3	ANHANG C: GAS-VISKOSITÄT NACH LUCAS	177
12.4	ANHANG D: GAS-VISKOSITÄT NACH CHUNG	181
13	LITERATUR	187

Abbildungsverzeichnis

Abb. 1:	Darstellung des Programmverzeichnisses der Programmsammlung unter Windows	14
Abb. 2:	Von dem Programm DTREAD erzeugtes structure array nach Einlesen des obigen Datenfiles	28
Abb. 3:	Arbeitsfenster des GUI <i>SELGUI</i>	33
Abb. 4:	Fließschema zum Programm <i>HRx</i>	38
Abb. 5:	Fließschema des Programmes <i>Rx2N</i>	39
Abb. 6.:	Fließschema des Programmes <i>Htemp</i>	50
Abb. 7:	Schematischer Verlauf der Isothermen im PV-Diagramm einer kubischen Gleichung im Zweiphasengebiet	59
Abb. 8:	Fließschema des Programmes <i>VOLCUBIC</i>	62
Abb. 9:	Verlauf der P/ ρ -Kurve für ein Gemisch aus Methan, Ethan und n-Dekan (230 K/40 bar), gemäß der BWRS-Gleichung	68
Abb. 10:	Vergleich unter Anwendung von <i>WILSON</i> berechneter und experimenteller (Punkte) Dampfdrücke des Systems Cyclohexan/Anilin bei 50.2°C	126
Abb. 11:	Intermolekulare Potentialkurve	141

Tabellenverzeichnis

Tab 1:	In <i>data</i> enthaltene stoffspezifische Größen	22
Tab 2:	I/O-Charakteristik der Funktion <i>PVAP</i>	25
Tab 3:	I/O-Charakteristik der Funktion <i>DTREAD</i>	26
Tab 4:	I/O-Charakteristik der Funktion <i>DATASEARCH</i>	30
Tab 5:	I/O-Charakteristik der Funktionen <i>Rx2N</i> und <i>Equ2Str</i>	38
Tab 6:	I/O-Charakteristik der Funktionen <i>HRx</i> , <i>SRx</i> und <i>GRx</i>	41

Tab 7: I/O-Charakteristik der Funktion <i>EMPTY</i>	44
Tab 8: I/O-Charakteristik der Funktionen <i>Htemp</i> , <i>Stemp</i> und <i>Gtemp</i>	46
Tab 9: Bildungsenthalpie von Methan, Ethan und Propan bei einer Temperatur von 900 K	52
Tab 10: I/O-Charakteristik der Funktion <i>CPIDEAL</i>	54
Tab 11: Kubische Zustandsgleichungen	58
Tab 12: Zuordnung der Reinstoffparameter der BWR-Gleichung für die Mischungsregel [25]	61
Tab 13: I/O-Charakteristik der Funktion <i>VOLCUB</i> und <i>VOLCUBIC</i>	63
Tab 14: Über tausend Funktionsaufrufe gemittelte Zeit [s] für einen Programmaufruf	64
Tab 15: I/O-Charakteristik der Funktion <i>ABVdW</i> , <i>ABRK</i> , <i>ABSRK</i> , <i>ABPR</i> und <i>ABMIX</i>	64
Tab 16: Vergleich durch <i>VBWR</i> berechneter Werte mit Literaturwerten	68
Tab 17: I/O-Charakteristik der Funktion <i>VBWR</i>	69
Tab 18: Parameter zur Berechnung der BWRS-Parameter nach Han-Starling [6]	71
Tab 19: I/O-Charakteristik der Funktionen <i>ABBWR</i> und <i>ABBWRMIX</i>	71
Tab 20: I/O-Charakteristik der Funktionen <i>DEPFUNCCUB</i> und <i>DEPFUNCBWR</i>	75
Tab 21: Vergleich mit der Funktion <i>DEPFUNCCUB</i> berechneter Werte mit Literaturwerten [1]	76
Tab 22: I/O-Charakteristik der Funktionen <i>HP/Hpres</i> und <i>HPresBWR</i>	77
Tab 23: I/O-Charakteristik der Funktionen <i>FUGCUB</i> und <i>FUGCUBIC</i>	79
Tab 24: I/O-Charakteristik der Funktion <i>FUGBWR</i> .	80
Tab 25: Vergleich berechneter Fugazitätskoeffizienten mit Literaturdaten [80] für H ₂ /NH ₃	81
Tab 26: I/O-Charakteristik der Funktionen <i>DEPARTCPCUB</i> und <i>DEPARTCPCUBIC</i>	82
Tab 27: I/O-Charakteristik der Funktion <i>DEPARTCPBWR</i>	84
Tab 28: Vergleich berechneter Exzeßwärmekapazitäten ΔC_p	

[J/(mol*K)] mit Literaturwerten [16]	84
Tab 29: I/O-Charakteristik der Funktion <i>KT</i>	86
Tab 30: I/O-Charakteristik von <i>KEYESM</i>	90
Tab 31: Ergebnis nach <i>KEYESM</i> Aufruf, (SK ≡ Schlüsselkomponenten, NSK ≡ Nichtschlüsselkomponenten)	91
Tab 32: I/O-Charakteristik von <i>ELSPEMA</i>	91
Tab 33: I/O-Charakteristik der Funktion <i>STOIMA</i>	94
Tab 34: I/O-Charakteristiken von <i>ESM2EQU</i> und <i>N2EQU</i>	95
Tab 35: I/O-Charakteristik der Funktion <i>KEYREAC</i>	95
Tab 36: I/O-Charakteristik der Funktion <i>EENL</i>	103
Tab 37: Vergleich zweier <i>EENL</i> -Aufrufe durch <i>FSOLVE</i> mit veränderten Startwerten mit Literaturwerten [47]	106
Tab 38: I/O-Charakteristik der Funktion <i>N2RXGLMAT</i>	111
Tab 39: I/O-Charakteristik der Funktion <i>EE</i>	112
Tab 40: Vergleich der Berechnung der Verbrennung von Propan bei 2200 K mit Literaturwerten [36]	112
Tab 41: I/O-Charakteristik der Funktion <i>EEM</i>	121
Tab 42: Gleichgewichtszusammensetzungen der Synthese von Ammoniak bei 450°C und 300 bar, berechnete Werte und Literaturwerte [83]	121
Tab 43: Tabelle der implementierten Gleichungen zur Berechnung von Aktivitätskoeffizienten [24]	124
Tab 44: I/O-Charakteristik der Funktionen <i>MARGULES</i> , <i>VANLAAR</i> , <i>UNIQUAC</i> , <i>NRTL</i> und <i>WILSON</i>	127
Tab 45: I/O-Charakteristik der Funktion <i>INMATRIX</i>	128
Tab 46: Zusammenfassung der Transportgrößen bei Gasen [12]	129
Tab 47: I/O-Charakteristik der Funktion <i>TCELYHANLEY</i>	130
Tab 48: I/O-Charakteristik der Funktion <i>TCCHUNG</i>	131
Tab 49: I/O-Charakteristik der Funktion <i>TCLATINI</i>	132
Tab 50: I/O-Charakteristik der Funktion <i>TCSATORIEDEL</i>	133
Tab 51: I/O-Charakteristik der Funktion <i>TCLI</i>	134
Tab 52: I/O-Charakteristik der Funktion <i>TCROWLEY</i>	135
Tab 53: I/O-Charakteristik der Funktion <i>VILUCAS</i>	138
Tab 54: I/O-Charakteristik der Funktion <i>VICHUNG</i>	139
Tab 55: I/O-Charakteristik der Funktion <i>LENJOPOT</i>	140
Tab 56: I/O-Charakteristik der Funktion <i>VISLIQUID</i>	143
Tab 57: Auf die Abschätzung der Viskosität bezogene Daten-	

felder in <i>data</i>	143
Tab 58: I/O-Charakteristik der Funktion <i>VITEJARICE</i>	145
Tab 59: I/O-Charakteristik der Funktion <i>DCHAPENS</i>	148
Tab 60: I/O-Charakteristik der Funktion <i>DWILKELEE</i>	150
Tab 61: Atomare Diffusionsvolumen nach Fuller, Schettler und Giddings	151
Tab 62: I/O-Charakteristik der Funktion <i>DFULSCHEGID</i>	151
Tab 63: Diffusionsvolumen einfacher Moleküle	152
Tab 64: Numerische Werte der Parameter in Gleichung [161]	176
Tab 65: Konstanten für die Abschätzung der Viskosität nach Lucas	179
Tab 66: Korrekturfaktor κ für Gleichung [191]	183
Tab 67: Koeffizienten für E_i in Gleichung [198]	184

Verzeichnis häufig verwendeter Abkürzungen

BWR-Gleichung	Benedict-Webb-Rubin-Gleichung
BWRS-Gleichung	Benedict-Webb-Rubin-Starling-Gleichung
DAGl	Differential-Algebraische-Gleichung
DGl	Differential-Gleichung
GUI	Graphische Benutzer Schnittstelle <u>G</u> raphical <u>U</u> ser <u>I</u> nterface
I/O-Charakteristik	Ein-/Ausgabe-Charakteristik, <u>I</u> n-/ <u>O</u> utput-characteristic
ODE-Solver	Löser gewöhnlicher Differentialgleichungen <u>O</u> rdinary- <u>D</u> ifferential- <u>E</u> quation-Solver

Windows ist ein eingetragenes Warenzeichen der Microsoft GmbH.
MATLAB ist ein eingetragenes Warenzeichen der The MathWorks GmbH.

1 Einleitung in die Programmsammlung

1.1 Vorstellung von MATLAB

MATLAB ist ein Softwaresystem für technisch-wissenschaftliche Hochleistungsrechnungen, das aus den ehemaligen FORTRAN-Bibliotheken EISPACK und LINPACK hervorgegangen ist [66]. Der Name MATLAB setzt sich aus den Worten MATRix-LABoratory zusammen und beschreibt somit schon eine wesentliche Stärke des Programmsystems. MATLAB ermöglicht die einfache, schnelle und sehr effiziente Bearbeitung von Matrix- und Vektoroperationen [46]. Die direkte Ausführung von Berechnungen ist dabei an eine Arbeitsoberfläche gekoppelt. Diese Arbeitsoberfläche stellt sich unter Windows als ein eigenes Fenster auf dem Desktop dar. Hier können sowohl direkte Berechnungen durchgeführt als auch Programme aufgerufen werden. Die Ergebnisse der Berechnungen wirken sich auf den Arbeitsspeicher, die *workspace*, aus. Mit einem gut zu handhabenden Editor können Programme in einer mathematisch orientierten Programmiersprache erstellt und als sogenannte M-Files abgelegt werden. Der Aufruf dieser M-Files kann dann wieder über das MATLAB-Fenster erfolgen.

Folgt man beim Programmieren eines M-Files einem bestimmten Muster, so können Funktionen erstellt und in die Funktionenarchitektur von MATLAB eingefügt werden. Eine MATLAB-Funktion ist dadurch gekennzeichnet, daß ihr beim Aufruf eine Parameterliste übergeben wird und sie eine vorher festgelegte Anzahl von Ergebnissen zurückliefert. Ein weiteres entscheidendes Merkmal einer Funktion ist, daß die in ihr ablaufenden Berechnungen lokal begrenzt sind. Sie haben bis auf ausdrücklich als global definierte Variablen keine Auswirkungen auf andere Funktionen oder die *workspace*, d.h. sie sind gekapselt.

Der Aufruf von M-Files ist nicht nur von dem MATLAB-Fenster aus möglich, sondern kann auch aus einem anderen M-File heraus erfolgen. Somit ist der Aufbau großer, verschachtelter Programmsysteme aus kleineren Einheiten möglich.

MATLAB bietet weiterhin die Möglichkeit, graphische Benutzerschnittstellen zu programmieren. Diesen graphischen Konstrukten, kurz GUI (von Graphical User Interface) genannt, können einfache MATLAB-Befehle oder ganze M-Files hinterlegt werden. Somit kann zum Beispiel die Handhabung von komplexen Programmen für den Benutzer erleichtert werden [50].

Für die vorliegende Arbeit wurden *toolboxen* angewendet. Dies sind die *Optimization Toolbox*, die weiterführende Optimierverfahren (s. Kap. 5.3.2) zur Verfügung stellt [45], und die *Symbolic Math Toolbox*, die die symbolische Bearbeitung mathematischer Gleichungen und einen Zugriff auf MAPLE ermöglicht [48].

Ab der MATLAB-Version 5 sind *cell arrays* und *structure arrays* verfügbar. Dabei handelt es sich um Datenkonstrukte, in denen prinzipiell jede Art von Daten nebeneinander gespeichert werden kann; das heißt, es ist problemlos möglich in einem *array* nebeneinander beispielsweise den Namen einer Größe, ihren numerischen Wert und den zugehörigen Gültigkeitsbereich abzulegen. Der Unterschied zwischen den beiden Datenkonstrukten besteht darin, daß ein *cell array* lediglich numeriert werden kann; die Datenfelder eines *structure array* können jedoch auch mit Namen versehen werden.

1.2 Aufbau der Programmsammlung

Die vorliegende Programmsammlung gliedert sich in verschiedene Bereiche. Dieser Einteilung trägt die Organisation der einzelnen Programme Rechnung. Sie sind inhaltlich geordnet, in mehreren Unterverzeichnissen abgelegt (s. Abb. 1), auch wenn sich dadurch verschiedene thematische Überschneidungen ergeben. So ist die Funktion *LENJOPOT* zur Berechnung des Lennard-Jones-Potentials in einem Unterverzeichnis zur Behandlung von Diffusionskoeffizienten abgelegt, obwohl es auch bei der Berechnung der Viskosität und der thermischen Leitfähigkeit benötigt wird. Um einen reibungslosen Gebrauch der Programmsammlung zu gewährleisten, ist es erforderlich, MATLAB-Pfade zu setzen, die auf die

neuen Ordner verweisen. Bei einer Neuinstallation kann der Benutzer dies durch den Aufruf des M-Files *INSTALL* von dem MATLAB-Fenster aus tun. *INSTALL* kopiert alle Files der Programmsammlung auf die Festplatte des Rechners und aktualisiert die MATLAB-Pfade.

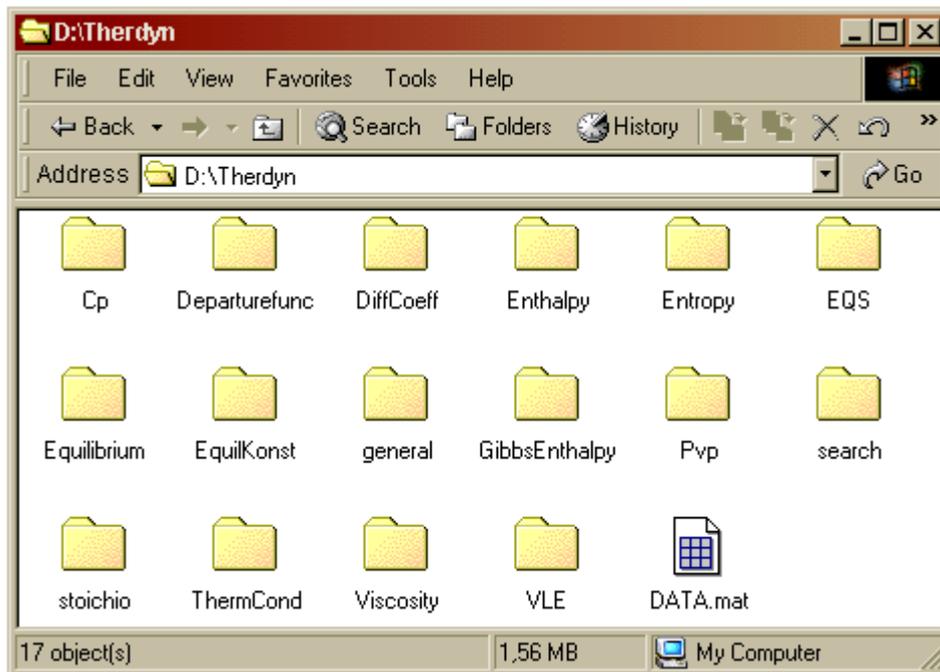


Abb. 1: Darstellung des Programmverzeichnisses der Programmsammlung unter Windows

Die Programmsammlung ist mit MATLAB 5.2 geschrieben und im Bereich der Behandlung von Differential-Algebraischen Gleichungen in Kapitel 5.3.5 auf MATLAB 5.3 angepaßt worden. Die Programmsammlung stellt keine gänzlich eigenständige Ergänzung zu MATLAB dar, da zu ihrer uneingeschränkten Benutzung die *Optimization Toolbox* notwendig ist.

Die Auswahl der hier betrachteten Themengebiete der Thermodynamik lehnt sich an Erfordernisse bei reaktionstechnisch orientierten Simulationen an. Dabei erhebt die Programmsammlung keinen Anspruch auf Vollständigkeit bezüglich relevanter Größen und Verfahren. Sie bietet lediglich Möglichkeiten an, Fragestellungen in dem genannten Themengebiet zu bearbeiten.

Thematisch läßt sich die Programmsammlung in vier große Bereiche einteilen:

- **Datenbasis**
Als Grundlage für thermodynamische Berechnungen beinhaltet die Programmsammlung Informationen von 721 Substanzen. Diese in Form eines *structure array* abgelegten Daten sind durch verschiedene Programme zugänglich.
- **Thermodynamische Berechnungen**
In der Programmsammlung sind Funktionen für grundlegende thermodynamische Berechnungen idealer und nicht idealer fluider Systeme enthalten. Dabei spielen Zustandsgleichungen eine wichtige Rolle.
- **Chemische Gleichgewichte**
Neben Funktionen zur Berechnung von Gleichgewichtskonstanten stehen verschiedene numerische Lösungsmöglichkeiten für die Betrachtung chemischer Gleichgewichte zur Verfügung.
- **Molekulare Transportprozesse**
Die Programmsammlung beinhaltet Abschätzungsverfahren für die Viskosität, für thermische Leitfähigkeiten und für Diffusionskoeffizienten von Fluiden.

Innerhalb dieser Einteilung sind zwei Themengebiete enthalten, die übergeordnete Bedeutung für andere Teile der Programmsammlung haben. Das sind die Behandlung von Zustandsgleichungen und die Organisation der Datenbasis. Beide Bereiche haben für nahezu alle anderen Themen der Programmsammlung eine zentrale Bedeutung und können daher auch als Schlüsselfunktionen betrachtet werden.

Die meisten der realisierten Methoden sind als Funktionen implementiert. Somit reiht sich die Programmsammlung nahtlos in das Funktionen-Konzept von MATLAB ein. Zum Aufruf einer MATLAB-Funktion muß neben dem Funktionsnamen eine Argumenteliste eingegeben werden. Durch die in der MATLAB-Funktion abgelegten Formeln wird dann der Wert berechnet, den die Funktion als Ergebnis zurückliefert. Dabei können die Parameter aus der Argu-

menteliste in diese Berechnung einfließen. Um die Verwendung von MATLAB-Funktionen in der Programmsammlung zu verdeutlichen, werden die Argumentelisten der Ein- und Ausgaben bei der Beschreibung der jeweiligen Funktion tabellarisch aufgeführt. Die dafür eingeführte Konvention soll hier anhand der Funktion *VOLCUBIC* verdeutlicht werden. Die Funktion berechnet das molare Volumen von Gasen durch die Lösung kubischer Zustandsgleichungen (weitere Informationen zu der Funktion *VOLCUBIC* siehe Kapitel 4.1).

```
VOLCUBIC(T,P,x,'string',val,key,Wij,swi1,swi2,R,Tc,Pc,Omega)
```

In der Argumenteliste sind alle Argumente wiedergegeben, die der Funktion *VOLCUBIC* übergeben werden können. Jedoch nicht jedes dieser Argumente müssen verwendet werden.

- Die Argumente T , P und x sind zur Ausführung der Funktion notwendig. Das macht die Zuweisung von Werten für jede Berechnung erforderlich.
- Weiterhin benötigt die Funktion Informationen über die Substanz, deren molares Volumen sie berechnen soll. Mit den Argumenten 'string' oder val werden in der Datenbasis enthaltene Substanzen identifiziert. Eines der beiden Argumente muß mit Werten belegt werden, wenn die Funktion die notwendigen Daten aus der Datenbasis entnehmen soll. Sollen hingegen andere als in der Datenbasis enthaltene Daten für die Berechnung verwendet werden, so muß deren Eingabe durch die Argumente T_c , P_c und Ω erfolgen. Bei der Belegung eines der Argumente 'string' oder val oder der Argumentengruppe T_c , P_c und Ω , können die jeweils anderen Argumente vernachlässigt werden. Die Kombination von eigenen Daten mit Daten aus der Datenbasis ist möglich. Diese Eingabemöglichkeiten werden im weiteren Verlauf der Arbeit als alternative Eingabemöglichkeit bezeichnet.
- Die restlichen Argumente (key , W_{ij} , swi_1 , swi_2 und R) sind optional und haben voreingestellte Werte.

Die erläuterten Verwendungsmöglichkeiten der Argumente werden noch in tabellarischer Form wiedergegeben, so wie es auch im weiteren Verlauf der Arbeit gehandhabt wird. Dabei sind alternative Eingabemöglichkeiten unterstrichen dargestellt. Optionale Argumente werden bei dieser Darstellung durch fette Schrift verdeutlicht.

<i>VOLCUBIC</i> :	Berechnung des molaren Volumens durch Lösen kubischer Gleichungen
Eingabe:	Temperatur, Druck, Vektor der Molenbrüche, <u>n Namen bzw. Summenformeln der beteiligten Spezies in Stringformat</u> , <u>Vektor der auf Data verweisenden Indizes</u> , Namen bzw. Summenformeln-Schalter , <u>krit. Temperatur</u> , <u>krit. Druck</u> , <u>krit. azentrischer Faktor</u> , binäre Interaktionsparameter , Phasen Schalter , Gleichungsschalter , Referenzdruck , Gaskonstante
Ausgabe:	Je nach Größe von T, P und x Matrix, Vektor oder Skalar des molaren Volumens

(**Fette Schrift** = optionale Argumente, unterstrichene Schrift = alternative Eingabemöglichkeit, siehe weitere Erläuterungen im Text und Kap. 1.2)

Bei der tabellarischen Darstellung wird nicht mehr besonders darauf hingewiesen, daß lediglich eine der Eingaben 'string', val oder T_c , P_c und Omega notwendig ist um die Berechnung durchzuführen.

Intern werden die Argumente durch ihre Position in der Argumenteliste identifiziert. Das kann bedeuten, daß für optionale Argumente leere Felder eingegeben werden müssen, damit die Position der nachfolgenden Argumente gewahrt bleibt. So berechnen die folgenden Funktionsaufrufe von *VOLCUBIC* beide das molare Volumen von Wasserstoff bei 300 K und 10 bar.

```
VOLCUBIC(300,10,1, [], [], [], 1, -1, 3, [], 33, 12.9, -0.2160)
```

```
VOLCUBIC(300,10,1, 'H2', [], 0)
```

Bei dem zuerst dargestellten Aufruf werden die zur Berechnung notwendigen Daten für T_c , P_c und Ω am Ende der Argumentenliste eingegeben. Bei dem zweiten Aufruf der Funktion werden die notwendigen Daten aus der Datenbasis entnommen.

Die Kombination von selbst eingegebenen Daten mit Daten aus der Datenbasis ist in dem folgenden Funktionsaufruf für `VOLCUBIC` gezeigt.

```
VOLCUBIC(300,10,1,'O2','#',[],0,1,-1,3,[],33,12.9,-0.2160)
```

Bei diesem Aufruf werden Daten für Sauerstoff aus der Datenbasis entnommen und Daten für Wasserstoff manuell eingegeben. Dabei signalisiert ein Platzhalter '#' in der Namensliste, daß mit den Argumenten T_c , P_c und Ω Daten für eine weitere Substanz eingegeben werden sollen. Eine detaillierte Beschreibung dieser Eingabestruktur findet sich in Kapitel 3.2.

2 Datenbasis

Viele der hier implementierten Abschätzungsverfahren benötigen Reinstoffparameter. Einige dieser Parameter sind in dem Datensatz *data* abgelegt. Die Zusammenstellung der gespeicherten Daten orientiert sich an den in Kapitel 1.2 erläuterten Gesichtspunkten, die der Themenauswahl der Programmsammlung zugrunde liegen.

2.1 Datenorganisation

Innerhalb des Datensatzes *data* sind die Substanzen nach der Anzahl der Kohlenstoffatome sortiert. Substanzen mit gleicher Kohlenstoffanzahl werden dann nach der Zahl der Wasserstoffatome und nachfolgend nach der Zahl der Heteroatome in alphabetischer Reihenfolge unterschieden.

Der Datensatz *data* ist in Form eines *structure array* organisiert und enthält alle in Tabelle 1 (s. S. 16) angegebenen Daten in eigenen, je nach dem Inhalt benannten Feldern. Zum Beispiel ist das Molgewicht als numerischer Wert in dem Feld mit dem Namen

MolWt gespeichert. Ein anderes Feld mit der Bezeichnung *name* enthält einen String, welcher den Namen der jeweiligen Substanz beinhaltet.

Des Weiteren besteht die Möglichkeit, mehrere dieser Felder zu einem eigenen Datenbereich zusammenzuführen. Dies ist durch Indizierung der betrachteten 'Ebene' des *structure arrays* möglich. Somit können auch uneinheitliche Daten organisiert und unter einem Oberbegriff gespeichert werden.

data enthält Daten von 721 Substanzen und ist in Form eines *MAT-Files* abgespeichert. Die so abgelegten Daten werden mit der Befehlszeile

```
» load data
```

geladen und stehen dann im Arbeitsspeicher zur Verfügung. Der Ladevorgang benötigt dabei nur wenige Sekunden. Durch den Aufruf eines indizierten Feldes sind dann alle Daten der betreffenden Substanz direkt zugänglich. Beispielsweise sind die Informationen für Methan unter der Indexnummer 116 abgelegt und können über die Feldbezeichnungen in Tabelle 1 (s. S. 16) abgerufen werden. So kann der Gefrierpunkt von Methan mit dem folgenden Befehl angezeigt werden: `data(116).Tfp`. Die Antwort von MATLAB lautet:

```
ans =
```

```
90.7000.
```

Im Verlauf der Arbeit werden ein Befehl und die dazu gehörige MATLAB-Antwort ohne weitere Erläuterung untereinander angegeben. Dabei wird ein eingegebener Befehl immer mit dem Zeichen '»' verdeutlicht. Davon abweichende Schreibweisen werden gesondert beschrieben.

Diese auf *data* verweisenden Referenz- oder Indexnummern werden im Folgenden zumeist nur noch als Indizes bezeichnet. Es sind

jedoch in dem Datensatz nicht für jede Substanz alle Daten verfügbar. Die entsprechenden Datenfelder sind leer. So beinhaltet *data* für Methan zum Beispiel keine Angaben über die Viskosität des Gases.

```
» data(116).VISCOSITY
```

```
ans =
```

```
[]
```

Neben dem Abruf von Informationen ist auch eine Erweiterung des Datensatzes um neue Substanzen oder neue Daten leicht möglich. Um dem Datensatz weitere Substanzen hinzuzufügen, muß ein neuer Datenbereich an *data* angehängt werden. Ausgehend von den 721 in *data* enthaltenen Substanzen fügt die Befehlszeile

```
» data(722).name = 'octanol';
```

einen neuen Datenbereich zu *data* hinzu. Das Datenfeld *name* beinhaltet unter der Indexnummer 722 den Namen einer neuen Substanz.

Alle anderen Datenfelder unter der Indexnummer 722 sind weiterhin leer.

```
» data(722).VISCOSITY
```

```
ans =
```

```
[]
```

Durch eine entsprechende Befehlszeile können weitere Daten der neuen Substanz gespeichert werden. Als Beispiel wird hier die Molmasse der betreffenden Substanz in *data* gespeichert:

```
» data(722).MolWt=130.23;.
```

Es ist auch möglich, Informationen in *data* zu speichern, die noch für keine andere Substanz enthalten sind. Mit folgender Befehlszeile wird ein weiterer Feldname definiert und belegt:

```
» data(722).Buchnummer=1163;
```

Das neu definierte Feld wird dabei an alle bereits bestehenden Datenbereiche als leeres Datenfeld angehängt. Das wird hier verdeutlicht am Beispiel des Aufrufs des neuen Datenfeldes für Methan.

```
» data(116).Buchnummer
```

```
ans =
```

```
    []
```

Permanent gespeichert werden die vorgenommenen Veränderungen an dem Datensatz *data* mit folgender Befehlszeile:

```
» save data data.
```

Damit wird der Inhalt von *data* in einem MAT-File mit dem Namen "data" abgespeichert. Der Name des zu erstellenden Files wird hier an erster Stelle nach dem Befehl *save* genannt. Daran schließt sich der Name der zu speichernden Variablen an. Mit *data* als Grundlage kann sich der Benutzer somit eine den Erfordernissen entsprechende Datenbank erstellen.

2.2 Inhalt des Datensatzes

Der verwendete Datensatz beinhaltet Daten von Reid, Prausnitz und Poling [1]. Dieser Datensatz wird *data* zugrunde gelegt, da er in kompakter Form Informationen enthält, die direkte reaktionstechnische Relevanz haben, so zum Beispiel Standardbildungsenthalpien oder freie Gibbssche Standardbildungsenthalpien, über die Gleichgewichtskonstanten berechnet werden können (s. Kap. 5.1). Weiterhin sind Daten beinhaltet, die indirekten reaktionstechnischen Bezug haben. Zum Beispiel werden kritische Größen zur Betrachtung von Zustandsgleichungen herangezogen (s. Kap. 4.1).

Die Spalte "Datenfeld" in Tabelle 1 gibt die Feldbezeichnungen des Datensatzes *data* wieder. Sie beinhalten die in der Spalte "Bedeutung" beschriebenen Informationen der betrachteten Substanz.

Tab 1: In *data* enthaltene stoffspezifische Größen

Datenfeld	Bedeutung	Einheit
MolWt	Molekulargewicht	g/mol
Tfp	Gefriertemperatur	K
Tb	Siedetemperatur bei 1 atm	K
Tc	Kritische Temperatur	K
Pc	Kritischer Druck	bar
Vc	Kritisches Volumen	cm ³ /mol
Zc	Kritischer Kompressibilitätskoeffizient	-
Omega	Azentrischer Faktor	-
Dipm	Dipolmoment	D
CPVAP_A bis CPVAP_D	Parameter zur Berechnung der isobaren Wärmekapazität des idealen Gases in J/(mol*K). Siehe Gleichung [1]	-
DELHF	Standardbildungsenthalpie bei 298.15 K und 1 atm	J/mol
DELGF	Freie Gibbssche Standardbildungsenthalpie bei 298.15 K und 1 atm	J/mol
Eq	Definition der Zugehörigkeit der folgenden vier Parameter zu einer Gleichung	-

Tab. 1: In *data* enthaltene stoffspezifische Größen (Fortsetzung)

VP_A, VP_B, VP_C, VP_D	Parameter zur Berechnung des Dampfdruckes in bar. Siehe Gleichung [2]-[4]	-
Tmin Tmax	Minimal- und Maximal-Temperatur für die Berechnung des Dampfdruckes	K
LDEN	Flüssigkeitsdichte	g/cm ³
TDEN	Temperatur für LDEN	K
name	Substanzname	-
sumform	Summenformel der Substanz	-
b0	Ausschlußvolumen ($b_0 = 2/3\pi N_0 \sigma^3$)	cm ³ /mol
Sigma	Moleküldurchmesser	Å
Epsilon	Quotient der charakteristischen Energie und der Boltzmann Konstanten	K
VIS_Eq	Definition der Zugehörigkeit der folgenden vier Parameter zu einer Gleichung	-
VIS_A, VIS_B, VIS_C, VIS_D	Parameter zur Berechnung der Flüssigkeitsviskosität in cP. Siehe Kapitel 7.2.3	-
VIS-Ta, VIS-Te	Gültigkeitsbereich der Gleichungen und Parameter zur Viskositätsberechnung	K
VISCOSITY, VIS_T	Viskosität bei der Temperatur VIS_T	cP, K

Die in Tabelle 1 enthaltenen Parameter finden in verschiedenen Gleichungen direkte Anwendung. Die Inhalte der Datenfelder *CPVAP_A*, *CPVAP_B*, *CPVAP_C* und *CPVAP_D* beziehen sich auf Gleichung [1]. Besagte Gleichung stellt ein Polynom der Temperatur zur Berechnung der isobaren Wärmekapazität dar. Hier werden aus Gründen der Übersichtlichkeit die Parameter mit A, B, C und D bezeichnet, also $A = CPVAP_A$, $B = CPVAP_B$, und so weiter:

$$C_p = A + B \cdot T + C \cdot T^2 + D \cdot T^3. \quad [1]$$

Gleichung [1] und die Inhalte obiger Datenfelder kommen direkt in der Funktion *CPIDEAL* zum Einsatz (s. Kap. 3.3).

Weiterhin sind Daten zur Bestimmung des Dampfdruckes in den Datensatz integriert. Insgesamt beinhalten die Datenfelder *VP_A*, *VP_B*, *VP_C* und *VP_D* Informationen für drei Dampfdruckgleichungen. Zwischen den drei Gleichungen wird mit dem Inhalt des Datenfeldes *Eq* unterschieden. Enthält *Eq* den Wert 1, so beinhalten die Datenfelder *VP_A* bis *VP_D* die Parameter für die Dampfdruckberechnung nach Wagner [1, 37]:

$$\ln(P_{vp}/P_c) = (1-x)^{-1} \cdot [A \cdot x + B \cdot x^{1.5} + C \cdot x^3 + D \cdot x^6]. [2]$$

$$x = (1 - T/T_c)$$

Auch hier werden aus Gründen der Übersichtlichkeit die Parameter mit A, B, C und D bezeichnet, also $A = VP_A$, $B = VP_B$, und so weiter.

Ist der Inhalt von *Eq* gleich 2, so gelten die Parameter *VP_A* bis *VP_D* für die Gleichung gemäß Frost, Kalkwarf und Thodos [1]:

$$\ln P_{vp} = A - B \cdot T^{-1} + C \cdot \ln T + D \cdot P_{vp} \cdot T^{-2}. [3]$$

Die Datenfelder *VP_A* bis *VP_C* beinhalten Parameter für die Antoine-Gleichung [4] [1, 35], wenn der Inhalt des Datenfeldes *data.Eq* gleich 3 ist:

$$\ln P_{vp} = A - \frac{B}{T + C}. [4]$$

Das Datenfeld *VP_D* bleibt in diesem Fall leer. Die Gleichungen [2] bis [4] sind in der Funktion *PVAP* realisiert. Diese berechnet den Dampfdruck entsprechend dem Inhalt des Datenfeldes *Eq*.

Tab 2: I/O-Charakteristik der Funktion *PVAP*

<i>PVAP</i> :	Berechnung des Dampfdruckes
Eingabe:	Temperatur, Indexnummer, Startschätzwert
Ausgabe:	Dampfdruck

(**Fette Schrift** = optionale Argumente)

Tabelle 2 gibt die Ein-/Ausgabe (In-/Output, kurz I/O-) Charakteristik des Files *PVAP* wieder. Dabei sind unter "Indexnummer" die auf das entsprechende Datenfeld verweisenden Indizes zu verstehen (s. Kap. 2.1). *PVAP* überprüft die in den Datenfeldern *Tmin* und *Tmax* gespeicherten Temperaturen. Diese stellen die obere und die untere Temperaturgrenze dar, innerhalb derer die Parameter *VP_A* bis *VP_D* anwendbar sind. Liegt die betrachtete Temperatur außerhalb des vorgesehenen Temperaturbereiches, so bricht *PVAP* die Verarbeitung mit einer Fehlermeldung ab. *PVAP* wendet zur Lösung von Gleichung [3] den Gleichungslöser *FSOLVE* an. Gleichung [3] ist dabei in dem M-File *PVAP2* abgelegt und wird aus *PVAP* heraus durch *FSOLVE* aufgerufen. Liefert die Funktion *FSOLVE* keinen Wert für den Dampfdruck, so kann der Benutzer mit einem weiteren Argument einen anderen Startschätzwert für den Gleichungslöser vorgeben. Die Eingabe dieses Argumentes ist freigestellt.

Des weiteren beinhaltet der Datensatz *data* Werte zur Abschätzung von Flüssigkeitsviskositäten. Die Funktion *VISLIQUID* verwendet die Inhalte der Datenfelder *VIS_A*, *VIS_B*, *VIS_C* und *VIS_D*. Diese Funktion schätzt die Viskosität von Flüssigkeiten unter Verwendung verschiedener Gleichungen ab. Im einzelnen wird diese Funktion in Kapitel 7.2.3 behandelt. Die Inhalte der Datenfelder *Ta* und *Te* geben den Temperaturbereich an, für den die in *VIS_A* bis *VIS_D* enthaltenen Parameter Gültigkeit besitzen.

In dem Datensatz *data* sind darüber hinaus Daten enthalten, die für viele Berechnungen relevant sind, aber nicht durch eine Gleichung berechnet werden müssen. So werden zum Beispiel die kritische Temperatur und der kritische Druck in den Datenfeldern *Tc* und *Pc* direkt für verschiedene Berechnungen eingesetzt. Weitere

wichtige Größen sind die Standardbildungsenthalpie in dem Datenfeld *DELHF* und die freie Gibbssche Standardbildungsenthalpie in dem Datenfeld *DELGF*. Die beiden Werte sind für den idealen Gaszustand bei 298.15 K gespeichert. Dabei sind die folgenden Referenzzustände für die Elemente benutzt worden:

- ideales Gas bei 1 atm: Ar, Cl₂, D₂, F₂, He, H₂, Kr, Ne, O₂, Rn, T₂, Xe,
- Feststoff: Al (krist.), As (krist.), B (krist.), C (Graphit), I₂ (krist.), P (rot), S (rhomboedrisch, krist.), Se (krist.), Si (krist.), Ti (α , krist.), U (krist.),
- Flüssigkeit: Br₂, Hg.

2.3 Daten aus der On-Line Datenbank DETHERM

Eine sehr viel größere Zahl an Informationen als der Datensatz *data* beinhaltet die Thermodynamische Datenbank der DECHEMA in Frankfurt, die DETHERM-Datenbank. Mit Hilfe dieser Datenbank bietet sich dem Benutzer die Möglichkeit, seine Datenbasis über die in *data* enthaltenen Daten hinaus zu erweitern.

Unter Verwendung des DETHERM Internet-Clients kann sich der Benutzer ein Datenfile der benötigten thermodynamischen Daten auf der Festplatte seines Rechners speichern. Die Daten werden in einem Textfile abgelegt. Um es dem Benutzer zu ermöglichen, die Daten dieses Files in eine MATLAB kompatible Form zu bringen, wurde das Programm *DTREAD* entwickelt.

Tab 3: I/O-Charakteristik der Funktion *DTREAD*

<i>DTREAD</i> :	Einlesen von DETHERM Datenfiles in MATLAB
Eingabe:	kompletter Filename, Angabe eines Pfades
Ausgabe:	<i>Structure array</i> aller gelesenen Daten.

(**Fette Schrift** = optionales Argument)

DTREAD liest alle in dem betrachteten File angegebenen Daten in ein *structure array* ein. Dabei werden alle Absatzbezeichnungen aus dem zu lesenden File als Bezeichnungen für Datenfelder in das

structure array übernommen. Dies ermöglicht eine eindeutige Zuordnung und Unterscheidung aller Informationen. Als Beispiel sei hier ein Ausschnitt aus einem DETHERM-Datenfile und dessen Entsprechung in einem *structure array* aufgeführt.

SYSTEM

MIXTURE-DATA

```

TABLE VLEMIX1
SOURCE-KEY 3.1-VLE-22.86
COMPONENTS COMP1 COMP2
STATE      VLE
PROPERTY   T          P          X          Y
UNIT       "K"         "Pa"       "mol/mol"  "mol/mol"
DATA      373.1      101325     0          0
&
DATA      370        101325     0.0108     0.1124
&
DATA      368.1      101325     0.022      0.186
&
DATA      367.1      101325     0.024      0.205

```

Multiple Einträge - wie die DATA Zeilen - werden als *cell array* in dem Datenfeld des erstellten *structure array* abgelegt. Daten, deren Feldbezeichnung nicht DATA ist, werden als Wort interpretiert und als Zeichenfolge gespeichert.

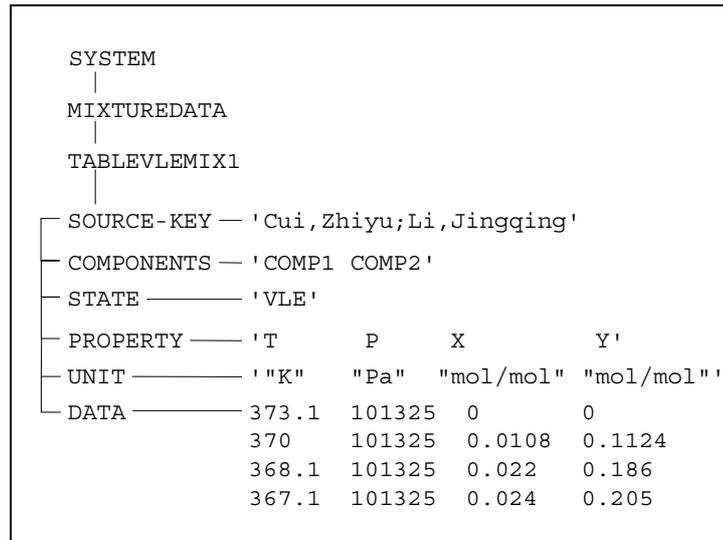


Abb. 2: Von dem Programm *DTREAD* erzeugtes *structure array* nach Einlesen des obigen Datenfiles

Etwaige numerische Werte sind damit nicht direkt als solche verwendbar, sondern müssen zuvor mit dem Befehl *str2num* umgewandelt werden.

```
» SYSTEM.SOURCE.KEY31VLE2286.VOLUME
```

```
ans =
```

```
'15'
```

```
» str2num(SYSTEM.SOURCE.KEY31VLE2286.VOLUME{1})
```

```
ans =
```

```
15
```

Die somit im MATLAB-Arbeitsspeicher verfügbaren Informationen können für eine Anwendung direkt benutzt werden. Die neuen Daten können jedoch auch zum Ausbau des Datensatzes *data* verwendet werden (s. Kap. 2.1).

Die zur Erweiterung von MATLAB verfügbare *Database Toolbox* ermöglicht die Kommunikation mit ODBC bzw. JDBC kompatiblen Datenbanken. ODBC ist die Abkürzung für Open Database Connectivity und stellt eine standardisierte Zugriffsmöglichkeit auf Datenbanken dar. JDBC ist die Abkürzung für Java Database Connectivity. JDBC ist eine Schnittstelle für Anwendungsprogramme, die es Java-Programmen ermöglicht, Datenbankabfragen in SQL, einer einheitlichen Datenbanksprache, auszuführen [76]. So können zum Beispiel von Oracle verwaltete Datenbanken mit der *Database Toolbox* angesprochen werden [73, 74]. Bei Oracle handelt es sich um ein Datenbank-Management-System, welches die einheitliche Verwaltung großer Datenmengen ermöglicht. Demgegenüber verwendet die DETHERM Datenbank eine eigene Zugriffssoftware. Sie unterstützt keine von der *Database Toolbox* gebotene Zugriffsmöglichkeit. Die aus der DETHERM-Datenbank erhaltenen Daten in einer eigenen Oracle Datenbank zu organisieren, um sie dann mittels der *Database Toolbox* abzurufen, ist möglich. Damit werden aber zwei separate Anwendungen verwendet, was unter Benutzung des Datensatzes *data* und des Files *DTREAD* in MATLAB umgangen werden kann.

Eine andere Möglichkeit, Daten aus einem ASCII-File in MATLAB einzulesen, besteht in der Verwendung von *DBAccess* [75]. Mit dieser MATLAB Erweiterung kann auf Inhalte von Textfiles zugegriffen werden, die eine Schlüsselwort basierende Struktur aufweisen.

```
SAMPLE DATA
MECH
  SPRING
    DATA
      1
    LOOP1
      0.5
    LOOP2
      3.0
```

In diesem Ausschnitt aus einem Beispielfile für die Verwendung von *DBAccess* sind *MECH*, *SPRING* und *DATA* Schlüsselworte. Die *DETERM*-Datenfiles besitzen eine ähnliche Struktur (s.Seite 22). Sie sind dennoch nicht mit *DBAccess* kompatibel. Diese Inkompatibilität resultiert aus der restriktiven Format-Auslegung von *DBAccess* [75].

Daher ist die Implementierung einer eigenen Softwarelösung im Rahmen der von *MATLAB* gebotenen File Ein-/Ausgabe-Funktionen gerechtfertigt.

2.4 Suchfunktionen

Bei einem Datensatz größeren Umfangs kann der Zugriff auf Daten einer bestimmten Substanz ein Suchproblem aufwerfen. Die Suche orientiert sich an dem Namen oder der Summenformel einer Substanz. Die gestellte Suchaufgabe ist somit genau definiert.

Da die *MATLAB* Funktion *FIND* nicht für die Verarbeitung von *structure arrays* vorgesehen ist, wurde die Funktion *DATASEARCH* geschrieben. Sie ermöglicht es, die auf *data* verweisende Indexnummer einer oder mehrerer Substanzen zu finden. Mit diesen Nummern ist es möglich, auf die Daten der gesuchten Substanzen zuzugreifen.

Tab 4: I/O-Charakteristik der Funktion *DATASEARCH*

<i>DATASEARCH</i> :	Suche von Indexnummern im Datensatz anhand von Substanznamen bzw. Summenformeln
Eingabe:	n <u>Substanznamen</u> bzw. n <u>Summenformeln</u> in <u>String-Format</u> oder als <u>cell array von Strings</u> , Such-Schalter
Ausgabe:	Je nach Anzahl der gesuchten Substanzen und der gefundenen Entsprechungen Matrix, Vektor oder Skalar der gefundenen Indexvektoren

(**Fette Schrift** = optionale Argumente, unterstrichene Schrift = alternative Eingabemöglichkeit, siehe weitere Erläuterungen im Text und Kap. 1.2)

Mit der Funktion *DATASEARCH* ist es möglich, nach dem Namen oder der Summenformel einer oder mehrerer Substanzen zu suchen. Diese Entscheidung kann über einen am Ende der Argumentenliste zu belegenden Schalter getroffen werden. Übergibt der Benutzer der Funktion eine Null für diesen Schalter, so sucht *DATASEARCH* nach der Summenformel der Substanz.

```
» datasearch('CH4',0)
```

```
ans =
```

```
116
```

Wird das Schalterargument mit dem Wert "eins" belegt, so sucht die Funktion nach dem Namen der Substanz.

```
» datasearch('methane',1)
```

```
ans =
```

```
116
```

Bei Nichtbelegung des Schalters wird die eingegebene Substanz als Summenformel interpretiert.

Die betrachteten Substanzen können in Form von einzelnen Strings oder als *cell array* einzelner Strings vor dem Such-Schalter übergeben werden.

DATASEARCH ermöglicht die Suche nach Anfangsilben bzw. Anfangsgruppen. Dazu muß der Benutzer an den zu suchenden String das Zeichen "*" anhängen. *DATASEARCH* wird daraufhin die Indexnummern aller gefundenen Entsprechungen ausgeben.

```
» datasearch('CH3*',0)
```

```
ans =
```

```
111 112 113 114 115
```

Mehr als eine Entsprechung kann *DATASEARCH* auch bei der Suche nach Summenformeln ausgeben, da diese nicht eindeutig sind.

```
» datasearch('C5H12',0)
```

```
ans =
```

```
311 312 313
```

Werden mehr als eine Summenformel definiert und mehrere Entsprechungen gefunden, gibt *DATASEARCH* eine Matrix der Indizes aus, wobei nicht belegten Matrixelementen der Wert Null zugewiesen wird.

```
» datasearch('CH4','C3H8','C5H12',0)
```

```
ans =
```

```
116 0 0  
206 0 0  
311 312 313
```

DATASEARCH ist bezüglich Abweichungen in der Groß-/Kleinschreibung nicht sensitiv. Falsch geschriebene oder falsch benannte Substanzen werden allerdings nicht gefunden. So kann der Benutzer vor dem Problem stehen, eine Substanz zu suchen, deren Benennung er nicht genau kennt. Die manuelle Suche in einer alphabetisch sortierten Liste aller in *DATA* gespeicherten Namen ermöglicht das Programm *SELGUI*. Bei diesem Programm handelt es sich um eine graphische Benutzerschnittstelle (*Graphical User Interface, GUI*).

Bei der Erstellung eines *GUI* können den Elementen der graphischen Oberfläche direkt Werte, Funktionen und auszuführende Programmschritte zugeordnet werden. Es ist jedoch auch möglich, den graphisch-interaktiven Elementen des *GUIs* ein oder mehrere

M-Files zu hinterlegen. Das GUI kann damit ausschließlich als graphischer Überbau über andere M-Files gestaltet werden. Dies kann unter Verwendung der GUI-Entwicklungsoberfläche *GUIDE* erfolgen. Dabei wird jedes graphische Element anhand von sogenannten *handles* identifiziert. Dies sind eindeutige Bezeichner der graphischen Elemente des GUIs [50].

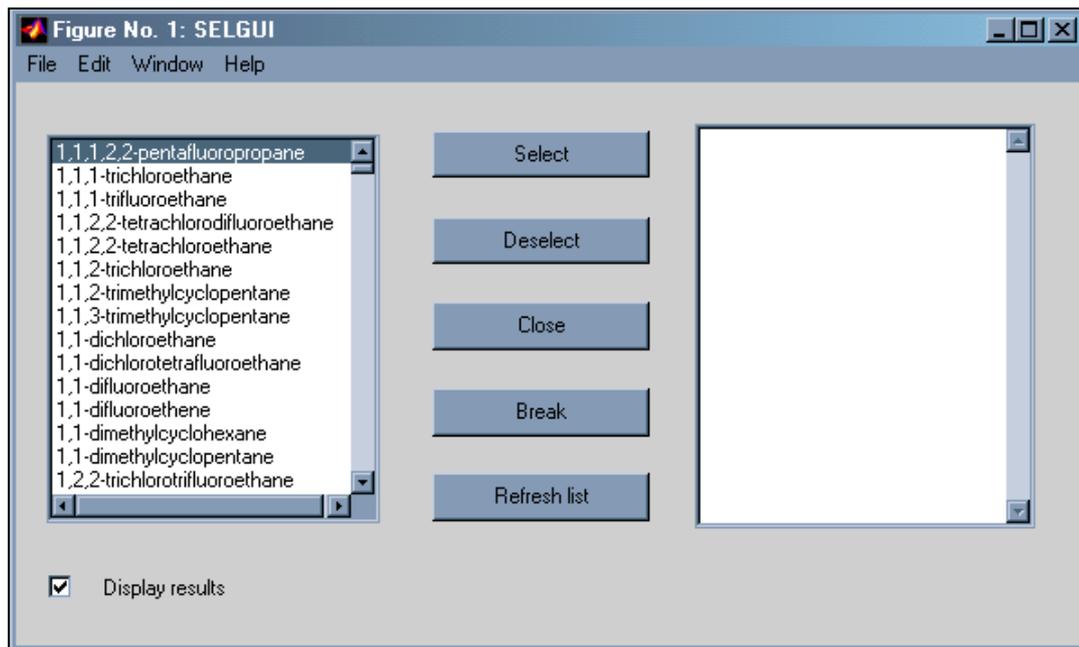


Abb. 3: Arbeitsfenster des GUI *SELGUI*

Das GUI *SELGUI* zeigt auf der linken Seite eine Namensliste aller in *data* enthaltenen Substanzen. Der Benutzer kann die Substanz fraglichen Namens in dieser Quell-Liste mit dem Mauszeiger markieren und mit dem ‚Select‘-Knopf in der Mitte auswählen. Die ausgewählte Substanz erscheint daraufhin in der Ziel-Liste auf der rechten Seite. Dies kann mit weiteren Substanzen wiederholt werden. Mögliche Fehler in der Auswahl können durch Markieren der fälschlicherweise ausgewählten Substanz in der Ziel-Liste und Drücken des ‚Deselect‘-Knopfes rückgängig gemacht werden. Mit dem ‚Close‘-Knopf wird das Programm beendet. Gleichzeitig werden die Indexnummern der ausgewählten Substanzen in der Variable *num* in dem MATLAB-Arbeitsspeicher gespeichert. Wird das Feld ‚Display results‘ markiert (s. Abb. 3), so werden die gefundenen Indexnummern in dem MATLAB-Fenster auch direkt angezeigt. Wird

dieses Feld nicht aktiviert, so werden die Indexnummern nur der Variablen *num* zugewiesen, aber nicht angezeigt. Durch Betätigen des ‚Break‘-Knopfes wird das Programm beendet, ohne daß der Variablen *num* ein Wert zugewiesen oder etwaige Indexnummern ausgegeben werden.

Der "Refresh list"-Knopf dient zur Aktualisierung der Quell-Liste nach einer Veränderung des Datensatzes *data* durch den Benutzer. Erst nach direkter Zuweisung des neuen, veränderten Datensatzes mit dem "Refresh list"-Knopf wirken sich die Veränderungen auch in dem *GUI* aus.

Bei der Implementierung von *SELGUI* wurde es notwendig, der Quell-Liste die Substanznamen aus *data* zuzuweisen. Problematisch ist, daß das *GUI* diese bei der Programmierung zugewiesenen Werte permanent speichert. Veränderungen an der der Quell-Liste zugrundeliegenden Variablen erscheinen nicht automatisch in dem *GUI*. Um die Flexibilität von *SELGUI* zu erhalten, wurde der "Refresh list"-Knopf erforderlich.

Es muß darauf hingewiesen werden, daß die Programmierung eines *GUI* trotz des neuen Entwicklungstools *GUIDE* und der *handle*-Technologie gewöhnungsbedürftig ist. Es gibt Anwendungen, bei denen ein *GUI* hilfreicher und eindeutiger eingesetzt werden kann, als nicht graphisch orientierte Programme. Jedoch muß bei dem Einsatz eines *GUI* stets die Relation zwischen Programmieraufwand und Erleichterung für den Benutzer berücksichtigt werden.

3 Thermodynamische Eigenschaften idealer Gase

Aus der historischen Entwicklung thermodynamischer Berechnungen ergibt sich die Einteilung in das ideale und das davon abweichende als real bezeichnete Verhalten von Gasen. Die zur Beschreibung idealen Gasverhaltens entwickelten Zusammenhänge sind mathematisch verhältnismäßig einfach. Weiterhin geben sie das Verhalten vieler Gase im direkt beobachtbaren Bereich mit akzeptabler Genauigkeit wieder.

Der Versuch, das Verhalten von Gasen bei höheren Drücken mittels der idealen Gasgleichung zu beschreiben, führt zu immer größeren Abweichungen. Hier kommen andere, speziell für diesen Bereich konzipierte Formelsysteme zur Anwendung.

In bezug auf reaktionstechnische Fragestellungen ist nicht allein die Veränderung einer thermodynamischen Größe mit der Temperatur, dem Druck und der Zusammensetzung von Interesse. Insbesondere sind auch die aus einer Reaktion resultierenden Veränderungen von großer Bedeutung. In diesem Kapitel werden daher die Reaktionsenthalpie $\Delta_r H$, die freie Gibbssche Reaktionsenthalpie $\Delta_r G$ und die Reaktionsentropie $\Delta_r S$ betrachtet. Weiterhin wird die Temperatur-abhängigkeit der Enthalpie, der Entropie und der freien Gibbsschen Enthalpie des idealen Gases behandelt.

Die Wärmekapazität eines idealen Gases und ihre Abhängigkeit von der Temperatur wird in Kapitel 3.3 thematisiert.

Bei der Implementierung wird die Bildung von Mischungsgrößen nicht berücksichtigt. Sie sind unter Anwendung der Vektorfunktionen von MATLAB auf die Mischungsregel

$$\Gamma_m = \sum_{i=1}^N x_i \cdot \Gamma_i \quad [5]$$

vom Benutzer leicht selbst durchführbar. In Gleichung [5] steht Γ für eine ideale thermodynamische Größe.

3.1 $\Delta_r H$, $\Delta_r G$ und $\Delta_r S$

Da bei chemischen Reaktionen oft eine beträchtliche Wärmetönung auftritt, wird in diesem Kapitel unter anderem die Reaktionsenthalpie $\Delta_r H$ und ihre Veränderung mit der Temperatur behandelt [2]. Reaktionsenthalpien können durch Linearkombinationen (s. Gleichung [6]) geeigneter anderer Reaktionen erhalten werden

$$\Delta_r H = \sum_{i=1}^N \nu_i \cdot \Delta_f H_i^0 . \quad [6]$$

Dazu können die tabellierten Standard-Bildungsenthalpien $\Delta_f H^\circ$ herangezogen werden. Die im Datensatz *data* enthaltenen Werte sind auf 298.15 K und 1 atm bezogen [1].

Die Temperaturabhängigkeit der Reaktionsenthalpie wird über das Kirchhoff'sche Gesetz berücksichtigt [12]:

$$\Delta_r H(T_2) = \Delta_r H(T_1) + \int_{T_1}^{T_2} \Delta_r C_p(T) dT . \quad [7]$$

In Gleichung [7] gibt $\Delta_r C_p$ die Summe der mit ihren stöchiometrischen Koeffizienten multiplizierten Wärmekapazitäten C_{pi} aller an der Reaktion beteiligter Substanzen wieder:

$$\Delta_r C_p = \sum_{i=1}^N \nu_i \cdot C_{pi} . \quad [8]$$

Die Temperaturabhängigkeit der Wärmekapazität C_{pi} bezieht sich, wie auch die Standard-Bildungsenthalpie $\Delta_f H_i^\circ$, auf einen Standardzustand.

Die freie Gibbssche Reaktionsenthalpie $\Delta_r G$ ist im Bezug auf reaktionstechnische Fragestellungen von besonderer Bedeutung, da mit ihr chemische Gleichgewichtskonstanten berechnet werden können [2]. Sie ergibt sich mittels einer Gleichung [6] entsprechenden Beziehung aus den Freien Gibbsschen Standardbildungsenthalpien.

Die Reaktionsentropie kann durch einen der Gleichung [7] entsprechenden Ausdruck wiedergegeben werden:

$$\Delta_r S(T) = \sum_{i=1}^N \nu_i \cdot S_i^\circ + \int_{298.15K}^{T_2} \Delta_r C_p(T) d\ln T . \quad [9]$$

Für die Bildung der ersten Summe in Gleichung [9] können die Standardentropien herangezogen werden:

$$\Delta_r S = \sum_{i=1}^N \nu_i \cdot S_i^0 . \quad [10]$$

Die S_i° -Werte der meisten Elemente und aller Verbindungen beziehen sich auf den idealen Gaszustand bei Standardbedingungen. Diese Entropien werden oft auch als absolute Entropien bezeichnet, da sie sich auf den dritten Hauptsatz der Thermodynamik beziehen [1].

Basierend auf dem Datensatz *data* werden die Berechnung von $\Delta_r H$, $\Delta_r G$ und $\Delta_r S$ mit den Programmen *HRx*, *GRx* und *SRx* realisiert. Diese ermöglichen die Berechnung der entsprechenden thermodynamischen Größe für n Reaktionen bei m Temperaturen. Dabei werden Phasenübergänge nicht berücksichtigt, das heißt, der Benutzer muß eventuell auftretende "Transformationswärmen" selbst in seine Betrachtung aufnehmen. Da die programmtechnische Realisierung der Berechnung von $\Delta_r H$, $\Delta_r S$ und $\Delta_r G$ sehr ähnlich ist, erfolgt die Erläuterung ausschließlich anhand der Berechnung der Reaktionsenthalpie.

Um Reaktionsenthalpien zu berechnen, muß der Benutzer zunächst die fraglichen Reaktionsgleichungen der Funktion übergeben. Bei der Implementierung der Funktion *HRx* wird die Eingabe von Reaktionsgleichungen von der eigentlichen Berechnung von $\Delta_r H$ getrennt. Die Eingabe der Reaktionsgleichungen erfolgt in dem separaten Programm *Rx2N*.

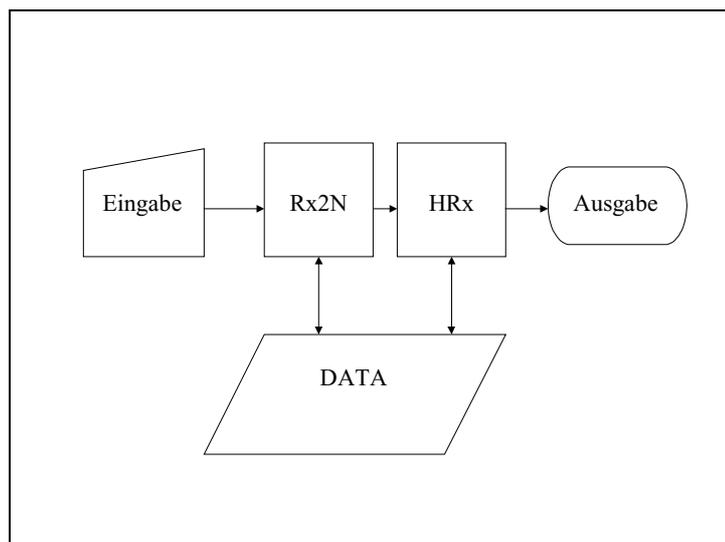


Abb. 4: Fließschema zum Programm *HRx*

Die Trennung ist sinnvoll, da es bei der Berechnung der Reaktionsentropie und der Freien Gibbschen Reaktionsenthalpie auch notwendig ist, Reaktionsgleichungen zu definieren. Weiterhin kann der Benutzer so die Analyse der eingegebenen Reaktionsgleichungen von der Berechnung der thermodynamischen Größe trennen und zeitkritische Routinen ohne die wiederholte Bearbeitung der Reaktionsgleichungen durchführen. Eingegeben werden die Reaktionen als einzelne Strings, nach dem Muster einer chemischen Reaktionsgleichung.

Tab 5: I/O-Charakteristik der Funktionen *Rx2N* und *Equ2Str*

<i>Rx2N</i> :	Bestimmung der Reaktionsmatrix und eines Indexvektors aus Reaktionsgleichungen
Eingabe:	n Reaktionsgleichungen in Stringformat
Ausgabe:	n Reaktionsmatrizen, n Indexvektoren
<i>Equ2Str</i> :	Bestimmung der stöchiometrischen Koeffizienten und der Summenformeln aus Reaktionsgleichungen
Eingabe:	Reaktionsgleichung in Stringformat
Ausgabe:	stöchiometrische Koeffizienten, Summenformeln

Beim Aufruf von *Rx2N* wird durch Verwendung des *cell arrays* mit Namen *varargin* eine Begrenzung der Anzahl an Reaktionen umgangen. *varargin* ermöglicht es, bei der Programmierung von Funktionen eine beliebige Zahl an Argumenten vorzusehen. Dazu wird als letztes Argument der Argumentenliste *varargin* angehängt. Die neue Funktion weist nun jedes "überzählige" Argument einem neuen Datenfeld in *varargin{i}* zu.

Die Funktion *equ2str* separiert aus dem variablen Argument *varargin* die stöchiometrischen Koeffizienten und die Summenformeln der eingegebenen Reaktionsgleichungen.

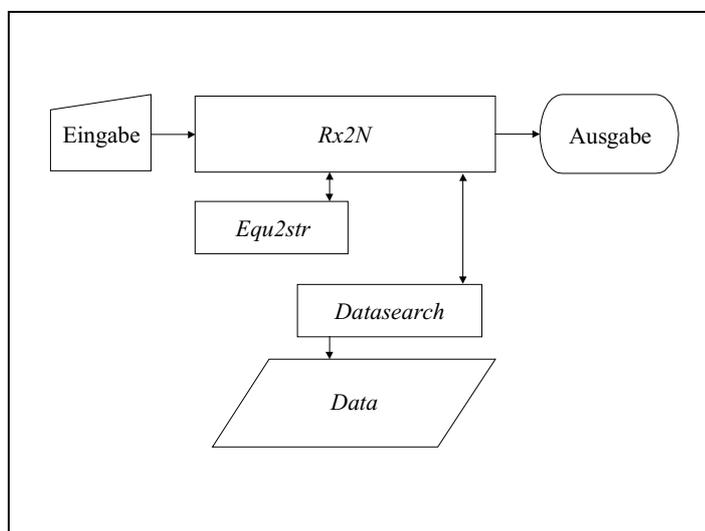


Abb. 5: Fließschema des Programmes *Rx2N*

Mit der Funktion *HRx* bieten sich dem Benutzer mehrere Möglichkeiten, die Datenbasis *data* zu nutzen. Wendet der Benutzer die Funktion *Rx2N* an, um die Eingabegrößen für *HRx* zu bestimmen, so überprüft *Rx2N* mit der Funktion *Datasearch* automatisch, ob die verwendeten Substanzen in dem Datensatz *data* enthalten sind. Dabei ergeben sich drei mögliche Suchergebnisse mit *Datasearch* im Datensatz *data*:

- es gibt eine
- es gibt keine oder

- es gibt mehr als eine zu der eingegebenen Substanz passende Entsprechung in *data*.

Findet *Rx2N* mit *Datasearch* mehr als eine Entsprechung in *data*, so wird diesbezüglich im MATLAB Fenster ein Textmenü aufgebaut.

```

» string1='H2 + 0.5 O2 -> H2O';
» string2='CH4 + 2 O2 -> CO2 + 2 H2O';
» [N,value]=RX2N(string1,string2)
There is more then one possible fit to the
used molecular formula. Please select the correct
one.
  1. hydrogen (equilibrium)
  2. hydrogen (normal)
  3. terminates calculation
Please enter a number  :1

N =

    -1.0    -0.5     1.0     0.0     0.0
     0.0    -2.0     2.0    -1.0     1.0

value =

    75     59     77    116     99

```

Mit diesem Textmenü wird es dem Benutzer ermöglicht, zwischen den aufgetretenen uneindeutigen Entsprechungen zu entscheiden. Ausgegeben werden eine Reaktionsmatrix und ein Zeilenvektor aller auf *data* verweisender Indizes. Nicht in *data* enthaltene Substanzen kennzeichnet *Rx2N* in dem Indexvektor *value* mit einer Null. Mit Reaktionsmatrix ist in diesem Zusammenhang die Matrix *N* der stöchiometrischen Koeffizienten aller eingegebenen Reaktionsgleichungen bezeichnet. Die Anwendung von Reaktionsmatrizen und Indizes zur Beschreibung von Reaktionen bieten den Vorteil der Eindeutigkeit. Dies ist zum Beispiel für Berechnungen, in denen $\Delta_r H$ sehr oft bestimmt werden muß, unerläßlich.

Tab 6: I/O-Charakteristik der Funktionen HR_x , SR_x und GR_x

HR_x :	Berechnung der temperaturabhängigen Reaktionsenthalpie
Eingabe:	Temperatur, <u>Vektor bzw. Matrix</u> der stöchiometrischen Koeffizienten, <u>Vektor der auf <i>data</i> verweisenden Indizes</u> , <u>Standardbildungsenthalpien, Parameter zur Berechnung der idealen Gaswärmekapazität</u>
Ausgabe:	Je nach Größe von T und N Matrix, Vektor oder Skalar der Reaktionsenthalpie bezogen auf den idealen Gaszustand
SR_x :	Berechnung der temperaturabhängigen Reaktionsentropie
Eingabe:	Temperatur, <u>Vektor bzw. Matrix</u> der stöchiometrischen Koeffizienten, <u>Vektor der auf <i>Data</i> verweisenden Indizes</u> , <u>Standardbildungsentropie, Parameter zur Berechnung der idealen Gaswärmekapazität</u>
Ausgabe:	Je nach Größe von T und N Matrix, Vektor oder Skalar der Reaktionsentropie bezogen auf den idealen Gaszustand
GR_x :	Berechnung der temperaturabhängigen Freien Gibbsschen Reaktionsenthalpie
Eingabe:	Temperatur, <u>Vektor bzw. Matrix</u> der stöchiometrischen Koeffizienten, <u>Vektor der auf <i>Data</i> verweisenden Indizes</u> , <u>Standardbildungsenthalpien, Standardbildungsentropie, Parameter zur Berechnung der idealen Gaswärmekapazität</u>
Ausgabe:	Je nach Größe von T und N Matrix, Vektor oder Skalar der freien Gibbsschen Reaktionsenthalpie bezogen auf den idealen Gaszustand

(**Fette Schrift** = optionale Argumente, unterstrichene Schrift = alternative Eingabemöglichkeit, siehe weitere Erläuterungen im Text und Kap. 1.2)

Sind alle in den Reaktionsgleichungen verwendeten Substanzen in dem Datensatz *data* enthalten, benötigt die Funktion HR_x zur Berechnung von $\Delta_r H$ die Reaktionsmatrix, den Indexvektor und einen Temperaturvektor. Alle relevanten Daten werden aus *data* entnommen. Sind nicht alle Substanzen in dem Datensatz enthalten, so ist es notwendig, die fehlenden Daten mittels optionaler Parameter einzugeben. Mit dieser Zweiteilung der Eingabemöglichkeit wird der einfache Zugriff auf die in *data* gespeicherten Daten kombiniert mit der Möglichkeit, nicht im Datensatz integrierte Daten

manuell eingeben zu können. Somit werden Berechnungen ausschließlich auf Basis des Datensatzes, teilweise mit oder gänzlich ohne diesen durchführbar.

Die in Tabelle 6 angegebenen Ausgabemöglichkeiten einer Matrix, eines Vektors oder einer skalaren Größe beziehen sich darauf, wieviele Ergebnisse errechnet werden müssen. Soll mit *HRx* die Reaktionsenthalpie einer Reaktion bei einer Temperatur bestimmt werden, so ist das Ergebnis eine skalare Größe. Werden hingegen zwei Reaktionen bei einer Temperatur betrachtet, so ergeben sich bereits zwei Ergebnisse. Die Ausgabe wird somit die Form eines Vektors haben. Werden die beiden Reaktionen nun bei zwei Temperaturen betrachtet, so resultieren vier Reaktionsenthalpien. Diese werden in Form einer Matrix ausgegeben. Wichtig ist in diesem Zusammenhang der Hinweis, daß in MATLAB ein skalarer Wert nichts grundsätzlich anderes ist als ein Vektor oder eine Matrix. Betrachtet man eine skalare Größe *X* in MATLAB mit der Funktion *size*, so erhält man Auskunft über die Zahl der Zeilen und Spalten von *X*.

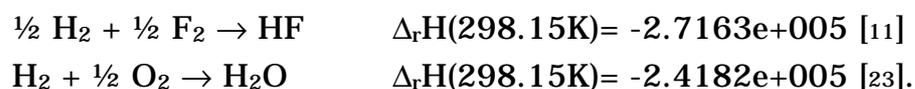
```
> X=1;  
> size(X)
```

```
ans =
```

```
1     1
```

Dennoch soll die Bezeichnung der Ausgaben (s. Tab. 6 und folgende) als Matrix, Vektor und Skalar beibehalten werden, da sie anschaulich deutlich macht, welche Dimension das zu erwartende Ergebnis des jeweiligen Funktionsaufrufes hat.

Als Anwendungsbeispiel werden hier die Reaktionsenthalpien folgender Reaktionen berechnet:



Dabei sollen die Daten für die Substanzen F₂ und HF zu Demonstrationszwecken manuell eingegeben werden.

```
» string1='0.5 H2 + 0.5 F2 -> HF';  
» string2='H2 + 0.5 O2 -> H2O';  
» [N,value]=RX2N(string1,string2)
```

Die Textmenüs zur Entscheidung über den molekularen Wasserstoff (s. Seite 37) in string1 und string2 sind hier nicht gezeigt.

N =

```
   -0.5000   -0.5000    1.0000         0         0  
   -1.0000         0         0   -0.5000    1.0000
```

value =

```
   75    32    72    59    77
```

Ob die benötigten Daten im Datensatz enthalten sind oder nicht, kann mit einem Aufruf der Function *EMPTY* überprüft werden.

```
» empty(value, 'DELHF')
```

ans =

```
   0    0    0    0    0
```

EMPTY überprüft die durch value und den Namen des Datenfeldes eindeutig bezeichneten Datenbereiche. Sind diese leer, so gibt die Funktion eine eins aus. Findet *EMPTY* nicht leere Datenbereiche, so gibt die Funktion dafür eine Null aus.

Tab 7: I/O-Charakteristik der Funktion *EMPTY*.

<i>EMPTY</i> :	Überprüft ob ein Datenfeld der fraglichen Substanzen Daten enthält
Eingabe:	n auf <i>data</i> verweisende Indexnummern, Bezeichnung des fraglichen Datenfeldes
Ausgabe:	Für jede überprüfte Substanz wird für ein leeres Datenfeld eine 1 und für ein Daten enthaltendes Datenfeld eine 0 ausgegeben.

Um die Werte für F_2 und HF beispielhaft manuell eingeben zu können, werden die entsprechenden Indexnummern in *value* gleich Null gesetzt.

```
» value(2:3) = [0,0]
value =

    75     0     0    59    77
```

Dies ist notwendig, da die neu eingegebenen Werte sonst nicht richtig zugeordnet werden können. Abschließend werden dann mit der Funktion *HRx* die Reaktionsenthalpien bei 298.15 Kelvin berechnet. Dabei sind die verwendeten Variablennamen (*dH*, *CPA*, *CPB*, *CPC*, *CPD*) willkürlich gewählt und nicht vorgeschriebene Bezeichner.

```
» dH = [0, -271300];
» CPA = [23.2200, 29.060];
» CPB = [0.0366, 0.0007];
» CPC = [-0.3613, -0.0203] * 1.0e-004;
» CPD = [0.1204, 0.0250] * 1.0e-007;
» HRx(298.15, N, value, dH, CPA, CPB, CPC, CPD)

ans =

1.0e+005 *
```

-2.7130 -2.4202

Beim Vergleich mit den Literaturwerten $\Delta_r H = -2.7163e+005$ J/mol [11] und $\Delta_r H = -2.4182e+005$ J/mol [23] zeigt sich die gute Übereinstimmung mit den berechneten Werten.

3.2 Temperatureinfluß auf H, G und S

Die Abhängigkeit der thermodynamischen Größen H, G und S von der Temperatur ist von grundlegendem Interesse. Bei reaktionstechnischen Betrachtungen werden oftmals nicht isotherme Bedingungen untersucht. Die genannten thermodynamischen Größen sind zum Beispiel wichtig für die Berechnung von Gleichgewichtskonstanten. Mit dem Kirchhoffschen Gesetz ergeben sich folgende Zusammenhänge [12]:

$$H_{T_2} = H_{T_1} + \int_{T_1}^{T_2} C_p(T) dT \quad [11]$$

$$S_{T_2} = S_{T_1} + \int_{T_1}^{T_2} C_p(T) d \ln T \quad [12]$$

Mit dem Differentialquotienten

$$\left(\frac{\partial G}{\partial T} \right)_p = -S \quad [13]$$

ergibt sich eine Möglichkeit, die Temperaturabhängigkeit der freien Gibbsschen Enthalpie zu berechnen [12]:

$$G_{T_2} = H_{T_1} - T_2 \cdot S_{T_1} - T_2 \cdot \int_{T_1}^{T_2} C_p(T) d \ln T + \int_{T_1}^{T_2} C_p(T) dT \quad [14]$$

Die Programme *Htemp*, *Stemp* und *Gtemp* ermöglichen durch die Anwendung der Gleichungen [11], [12] und [14] Aussagen über die Änderung der Enthalpie, der Entropie und der freien Gibbsschen Enthalpie mit der Temperatur. Da die programmtechnische Realisierung dieser Berechnungen sehr ähnlich ist, wird im Folgenden das Programm *Htemp* beispielhaft für die Programme *Htemp*, *Stemp* und *Gtemp* behandelt werden. *Stemp* und *Gtemp* unterscheiden sich von *Htemp* nur dadurch, daß nicht nur H_1 , sondern auch noch G_1 bei der Temperatur T_1 eingegeben werden kann.

Tab 8: I/O-Charakteristik der Funktionen *Htemp*, *Stemp* und *Gtemp*

<i>Htemp</i> :	Berechnung der Enthalpieänderung in Abhängigkeit von der Temperatur
Eingabe:	Temperatur, <u>Name/Summenformel der zu berechnenden Substanzen</u> , <u>Vektor der auf <i>Data</i> verweisenden Indizes</u> , Name/Summenformel Identifizierungsschalter , <u>Ausgangsenthalpie</u> , <u>Parameter zur Berechnung der idealen Gaswärmekapazität</u> , Starttemperatur
<i>Stemp</i> :	Berechnung der Entropieänderung in Abhängigkeit von der Temperatur
Eingabe:	Temperatur, <u>Name/Summenformel der zu berechnenden Substanzen</u> , <u>Vektor der auf <i>Data</i> verweisenden Indizes</u> , Name/Summenformel Identifizierungsschalter , <u>Ausgangsenthalpie</u> , <u>freie Gibbssche Ausgangsenthalpie</u> , <u>Parameter zur Berechnung der idealen Gaswärmekapazität</u> , Starttemperatur
<i>Gtemp</i> :	Berechnung der Änderung der freien Gibbsschen Enthalpie in Abhängigkeit von der Temperatur
Eingabe:	Temperatur, <u>Name/Summenformel der zu berechnenden Substanzen</u> , <u>Vektor der auf <i>Data</i> verweisenden Indizes</u> , Name/Summenformel Identifizierungsschalter , <u>Ausgangsenthalpie</u> , <u>freie Gibbssche Ausgangsenthalpie</u> , <u>Parameter zur Berechnung der idealen Gaswärmekapazität</u> , Starttemperatur

Tab 8: I/O-Charakteristik der Funktionen *Htemp*, *Stemp* und *Gtemp* (Fortsetzung)

Ausgabe:	Je nach Anzahl zu berechnender Substanzen und Temperatur Datenpunkte, Matrix, Vektor oder Skalar der auf den idealen Gaszustand bezogenen temperaturabhängigen - Enthalpie - Entropie - freien Gibbsschen Enthalpie
----------	--

(**Fette Schrift** = optionale Argumente, unterstrichene Schrift = alternative Eingabemöglichkeit, siehe weitere Erläuterungen im Text und Kap. 1.2)

Htemp berechnet, ausgehend von der Enthalpie H_1 , die Enthalpie H_2 der definierten Substanzen bei der Temperatur T_2 . Dabei ist es möglich, ausgehend von einer Starttemperatur die Enthalpien bei n Zieltemperaturen ermitteln.

Die Argumentverarbeitung in *Htemp* orientiert sich an der Eingabestruktur der Funktion *HRx*. Die Kombination von manuell eingegebenen Daten mit Daten aus dem Datensatz *data* ist möglich. Jedoch sind die Routinen zur Behandlung von *varargin* in *HRx* und *Htemp* unterschiedlich.

Bei *HRx* entfällt die Eingabe von chemischen Substanzen über Strings oder wird durch die Verwendung der Funktion *Rx2N* eindeutig. Bei der Anwendung von *Htemp* muß der Benutzer neben den Temperaturen T_1 und T_2 die gewünschten Substanzen eingeben. Dies kann in Form der Substanznamen, der entsprechenden Summenformeln oder der auf *data* verweisenden Indexnummern erfolgen. Die Unterscheidung zwischen Namen und Summenformeln wird wieder (s. Kap. 2.4) durch ein Schalterargument mit den Werten 1 und 0 geregelt. Die drei folgenden Aufrufe von *Htemp* liefern das gleiche Ergebnis, daher wird die Ausgabe des Ergebnisses hier nur einmal angegeben.

» `HTEMP(900, 'ethylbenzene', [], 1)`

```
» HTEMP(900, 'C8H10', [], 0)
```

```
» HTEMP(900, [], 447)
```

```
ans =
```

```
1.6728e+005
```

Der Aufruf bei dem die Summenformel zur Beschreibung der Substanz verwendet wird unterscheidet sich von den anderen durch seine Mehrdeutigkeit. Neben Ethylbenzol sind in *data* noch o-, m- und p-Xylol mit der Summenformel C_8H_{10} gespeichert. Das diesbezügliche Textmenü (s. Seite 37) zur Entscheidung zwischen den Substanzen ist hier nicht gezeigt. Werden zur Eingabe der Substanzen sowohl Strings als auch die Indexnummern benutzt, so werden die Indexnummern gegenüber den ausgeschriebenen Substanznamen bzw. Summenformeln bevorzugt verarbeitet. Die ausgeschriebenen Substanzbezeichnungen sind fehlerträchtiger als die Indexnummern.

Darüber hinaus ist die Kombination der Informationen aus *data* mit manuell eingegebenen Daten möglich. Aus dem Datensatz ausgelesen bzw. direkt eingegeben werden bei *Htemp* die Standardbildungsenthalpie ($data.DELHF \equiv H_0$) und Konstanten zur Berechnung der isobaren Wärmekapazität ($data.CPVAP_A \equiv CPA$ bis $data.CPVAP_D \equiv CPD$).

In diesem Beispiel werden die Bildungsenthalpien von Ethylbenzol und Styrol bei 900 K berechnet.

```
CPA = -43.1000;  
CPB = 0.7072;  
CPC = -4.8110e-004;  
CPD = 1.301e-007;  
H0 = 29810;  
» HTEMP(900, [], [440, 0], [], H0, CPA, CPB, CPC, CPD)
```

```
ans =
```

```
1.0e+005 *  
  
2.7446    1.6728
```

Styrol wird hier beschrieben durch die Indexnummer 440. Die verwendeten Variablennamen (H0, CPA, CPB, CPC und CPD) sind willkürlich gewählt.

In diesem Beispiel wird deutlich, wie manuell eingegebene Daten mit Daten aus *data* kombiniert werden können. Durch die Angabe einer Null in dem Vektor der Indexnummern wird dem Programm signalisiert, daß an dieser Stelle die Berechnung mit den manuell eingegebenen Daten fortgesetzt werden soll. Wird eine Stringeingabe für die aus *data* auszulesenden Daten gewählt, so gibt ein Nummernzeichen # die Stelle an, an der mit den manuell eingegebenen Daten gerechnet werden soll.

```
» HTEMP(900, 'styrene', '#', [], 1, H0, CPA, CPB, CPC, CPD)
```

```
ans =
```

```
1.0e+005 *  
  
2.7446    1.6728
```

Es ist somit nur möglich, die Daten für eine Substanz vollständig aus *data* zu entnehmen oder vollständig manuell einzugeben.

Zur Verarbeitung dieser Kombination von manuell eingegebenen Daten mit Werten aus *data* ist eine aufwendige Entscheidungsroutine erforderlich. Zu deren Verwirklichung werden Subfunktionen benutzt, die es ermöglichen in einer Funktion gekapselte Programmbereiche zu schaffen, die nur von der übergeordneten Funktion aus aufgerufen werden können.

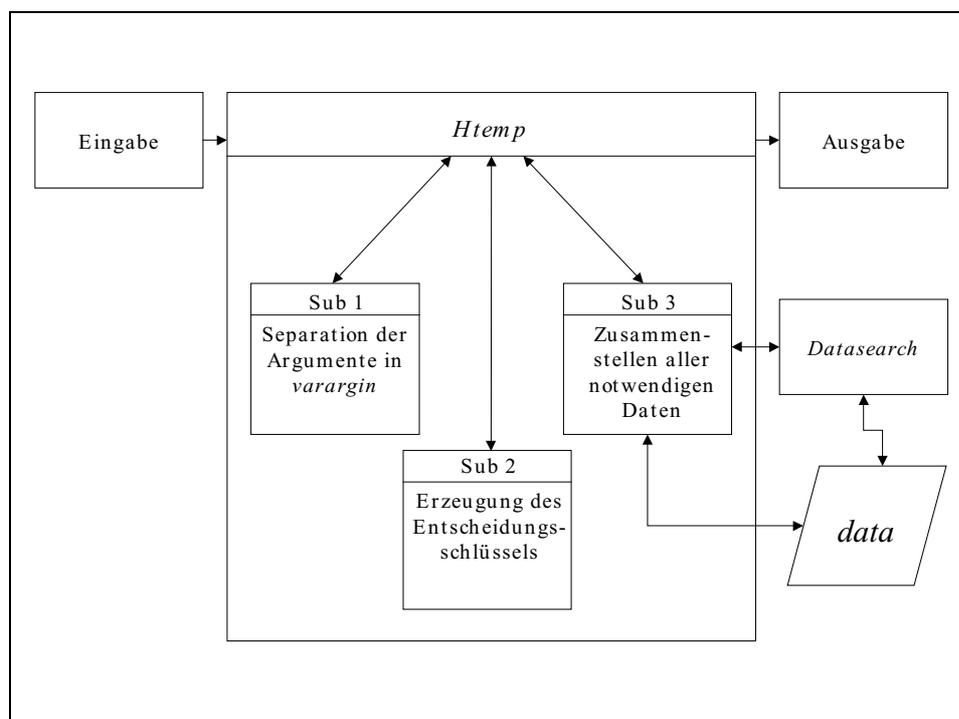


Abb. 6.: Fließschema des Programmes *Htemp*

In der ersten Subfunktion (Sub1) werden die mittels *varargin* übergebenen Daten separiert und einzelnen Variablen zugewiesen. Diese Subfunktion gibt Variablen für folgende Werte zurück:

- ausgeschriebene Substanzbezeichnungen (*string*),
- Indexnummern (*val*),
- Schalterargument (*key*) und
- manuell eingegebene Werte für die Ausgangsenthalpie H_1 und zur C_p Berechnung (H_0 , CPA , CPA , CPA und CPD).

Eine weitere Subfunktion (Sub2) analysiert, ob die Konstanten zur C_p Berechnung und die Ausgangsenthalpie H_1 aus dem Datensatz entnommen werden müssen oder ob sie direkt eingegeben wurden und erzeugt diesbezüglich einen Entscheidungsschlüssel. Werden zu wenig Informationen eingegeben, wird die Ausführung des Programmes mit einer entsprechenden Fehlermeldung abgebrochen.

Mit Hilfe dieses Entscheidungsschlüssels stellt eine dritte Subfunktion (Sub3) alle Konstanten zur Berechnung der Wärmekapazität und die Ausgangsenthalpien zusammen. Wurden die Substanzen durch ihre Namen bzw. Summenformeln identifiziert, ordnet Sub3 mit der Funktion *Datasearch* die entsprechenden Indexnummern zu. Anschließend entnimmt sie die notwendigen Daten dem Datensatz und kombiniert diese mit den direkt eingegebenen Werten. Bei der Suche der Indexnummern von Substanznamen durch die Funktion *Datasearch* können wieder (s. Kap. 3.1) Mehrfachentsprechungen gefunden werden. Durch eine Benutzerentscheidung werden mit einem Textmenü die eindeutigen Indexnummern ermittelt.

Mit *Htemp* kann der Benutzer die Änderung der Enthalpie mit der Temperatur zwischen beliebigen Temperaturen berechnen. Voreingestellt ist die Berechnung der Bildungsenthalpie bei einer Temperatur T . Dabei wird von der Standardbildungsenthalpie bei 298.15 Kelvin ausgegangen. Soll eine Enthalpie bei einer anderen Temperatur berechnet werden, so müssen mit den Argumenten H_0 und T_0 die entsprechende Ausgangsenthalpie und -temperatur eingegeben werden.

Gemische von Gasen werden in dem Programm *Htemp* nicht berücksichtigt. Der Benutzer kann die Berechnung der Mischungsgröße aus den Reinstoffgrößen leicht selbst durchführen (s. Gl. [5]). In diesem Beispiel werden die Bildungsenthalpien von Methan, Ethan und Propan bei einer Temperatur von 900 K berechnet.

» load data

```

» val=[116,168,206];
» Hi =HTEMP(900, [],val)

```

Hi =

```

1.0e+004 *

3.1350    5.2635    7.5967

```

Tabelle 9 enthält die H_i entsprechenden Literaturwerte.

Tab 9: Bildungsenthalpie von Methan, Ethan und Propan bei einer Temperatur von 900 K [11]

ΔH in J/mol	Methan	Ethan	Propan
Literaturwert	31427	52644	75743
Berechneter Wert	31350	52635	75967

Anschließend wird mittels Gleichung [5] die entsprechende Mischungsgröße bestimmt. x gibt hier die Molenbrüche der Substanzen in der Mischung wieder.

```

» x=[0.2,0.3,0.5];
» x*Hi '

```

ans =

```

6.0044e+004

```

3.3 Temperatureinfluß auf die Wärmekapazität idealer Gase

Die schon in Abschnitt 3.1 angesprochene Wärmekapazität C_p gibt die Änderung der Enthalpie mit Veränderung der Temperatur bei konstantem Druck wieder.

$$c_p = \left(\frac{\partial H}{\partial T} \right)_p \quad [15]$$

Die Wärmekapazität ist bezogen auf den idealen Gaszustand bei 298.15 K und einer Atmosphäre. Zur Berechnung der isobaren Wärmekapazität idealer Gase kommt ein Polynom dritter Ordnung der Temperatur zur Anwendung.

$$C_p^0 = A_{C_p} + B_{C_p} \cdot T + C_{C_p} \cdot T^2 + D_{C_p} \cdot T^3 \quad [16]$$

Die Parameterwerte des Polynoms sind in der Literatur tabelliert. Für die in *data* enthaltenen Substanzen sind sie in den Datenfeldern *CPVAP_A* bis *CPVAP_D* im Datensatz integriert. In Gleichung [16] sind die Inhalte der Datenfelder abgekürzt wiedergegeben, d.h. A_{C_p} entspricht *CPVAP_A*, B_{C_p} entspricht *CPVAP_B* und so weiter.

Für die Wärmekapazität eines Gemisches idealer Gase gilt das unter 3.2 Gesagte. Diese kann somit gemäß folgender Gleichung bestimmt werden.

$$C_{p,m}^0 = \sum_{i=1}^N y_i \cdot C_{p,i}^0 \quad [17]$$

In dem File *CPIDEAL* wird die Berechnung der Wärmekapazität des idealen Gases mit Gleichung [16] implementiert. Da die Berechnung auch mit Daten aus *data* erfolgen kann, wird auch hier eine Eingabeform gewählt, die es ermöglicht, direkt auf den Datensatz

data zuzugreifen. Die Analyse der Eingaben erfolgt ähnlich der Bearbeitung in der Funktion *Htemp* durch eine Entscheidungsroutine (s. Abschnitt 3.2). Diese ermöglicht es, die Berechnung mit dem Datensatz *data*, ohne diesen und mit einer Kombination aus Daten des Datensatzes, mit eingegebenen eigenen Daten durchzuführen. Der Unterschied liegt darin, daß hier nur auf die Datenfelder *CPVAP_A* bis *CPVAP_D* zugegriffen wird. Bei *Htemp* wird unter Umständen auch das Datenfeld *DELHF* abgefragt.

Tab 10: I/O-Charakteristik der Funktion *CPIDEAL*

<i>CPIDEAL</i> :	Berechnung der Wärmekapazität idealer Gase
Eingabe:	Temperatur, <u>Namen/Summen-formeln der beteiligten Substanzen</u> , <u>Indexnummern der Substanzen</u> , Namen/Summenformel Schalter , <u>nicht in <i>data</i> enthaltene Parameter gemäß [16]</u>
Ausgabe:	je nach Größe von T und P Matrix, Vektor oder Skalar der Wärmekapazität von idealen Gasen

(**Fette Schrift** = optionale Argumente, unterstrichene Schrift = alternative Eingabemöglichkeit, siehe weitere Erläuterungen im Text und Kap. 1.2)

Als Anwendungsbeispiel wird die Wärmekapazität von Kohlendioxid bei 300 °C berechnet. Der Literaturwert beträgt 37.196 J/(mol*K) [11].

```
» load data
» cpideal(300, [], 99)
```

```
ans =
```

```
37.2533
```

4 Thermodynamische Eigenschaften realer Gase

Die bisher betrachteten Zusammenhänge besitzen für den technischen Einsatz oftmals einen entscheidenden Nachteil. Sie beziehen sich zumeist auf den Bereich geringen Druckes. Da die ein solches System betreffenden Berechnungen auch als ideal bezeichnet werden, soll im Folgenden von realem Verhalten gesprochen werden.

Insbesondere Abweichungsfunktionen bzw. Exzeßgrößen geben den Unterschied zwischen idealem und realem Verhalten von Fluiden wieder. Sie müssen mit Hilfe von Zustandsgleichungen berechnet werden, die in der Lage sind, das Verhalten der betrachteten Substanzen im angegebenen Temperatur- und Druckbereich richtig wiederzugeben.

Zentrale Themen dieses Abschnittes sind Funktionen, die es dem Benutzer ermöglichen, Zustandsgleichungen zu lösen.

Neben der Bestimmung von Abweichungsfunktionen (Kap. 4.2) und deren Anwendung (Kap. 4.3) sind die Berechnung von Fugazitätskoeffizienten (Kap. 4.4) und Wärmekapazitäten für reale Gase (Kap. 4.5) Einsatzgebiete für Zustandsgleichungen.

4.1 Zustandsgleichungen

Viele Reaktionen werden häufig bei hohen Drücken durchgeführt. Es zeigt sich, daß man dann nicht mehr auf den einfachen p, V, T -Zusammenhang der idealen Gasgleichung zurückgreifen kann. Deshalb werden hier thermische Zustandsgleichungen für Gase vorgestellt [77]. So können mittels einer geeigneten Zustandsgleichung zum Beispiel Aussagen über folgende Größen erhalten werden:

- Dichte von Gasen und Flüssigkeiten
- Dampf-Flüssig-Gleichgewichtsdaten
- Exzeßenthalpien und Exzeßentropien.

Obwohl viele Zustandsgleichungen vorgeschlagen wurden, gibt es zur Zeit keine Gleichung, die für alle Substanzen und alle Temperatur- und Druckbereiche Gültigkeit besitzt. Walas [6] und Dohrn [78] bieten einen Überblick der historischen Entwicklung und der aktuellen Anwendung von Zustandsgleichungen. Betrachtet werden in dieser Arbeit zwei Klassen von Zustandsgleichungen, die vor allem auch in der Technik eine breitere Anwendung gefunden haben: Kubische Gleichungen vom Van der Waals Typ (s. Tab. 11 und Gl. [18]) und die von Benedict, Webb und Rubin (BWR) vorgeschlagene Zustandsgleichung (s. Gl. [20]):

$$P = \frac{R \cdot T}{V - b} - \frac{a}{V^2 + u \cdot b \cdot V + w \cdot b^2}. \quad [18]$$

Die Popularität der kubischen Gleichungen gründet sich auf ihre Fähigkeit, neben dem Verhalten der Gasphase auch das Verhalten der flüssigen Phase zu beschreiben. Des weiteren sind sie dabei recht problemlos zu handhaben. Weiterhin muß eine Gleichung, die die Kriterien [19] des kritischen Punktes sinnvoll erfüllen soll, mindestens kubisch bezüglich des Volumens sein:

$$\left(\frac{\partial P}{\partial V} \right)_{T_c} = \left(\frac{\partial^2 P}{\partial V^2} \right)_{T_c} = 0. \quad [19]$$

Die in Tabelle 11 (s. S. 52) angegebenen Ausdrücke für die Parameter a und b wurden mittels der Kriterien [19] erhalten. Unter Verwendung der kritischen Größen Temperatur und Druck wurde [19] auf die kubische Gleichung angewandt und somit ein Ausdruck für die Parameter a und b erhalten. Sowohl Soave [31] als auch Peng und Robinson [32] haben auf diesem Weg für a und b Funktionen der Temperatur und des azentrischen Faktors erhalten.

Die BWR-Gleichung eignet sich zur Berechnung der Gasphase bis zur kritischen Dichte mit guter Genauigkeit. Bei Drücken zwischen

200 kPa und dem kritischen Punkt sind die maximalen Fehler kleiner als 5 % [78]. Ähnlich den kubischen Gleichungen wurden auch diese Zustandsgleichungen erfolgreich auf den flüssigen Bereich ausgedehnt

[6, 18]. Sie sollen als Beispiel für komplexere Zustandsgleichungen betrachtet werden:

$$\begin{aligned}
 P &= \rho \cdot R \cdot T + u_1 \cdot \rho^2 + u_2 \cdot \rho^3 + \alpha_0 \cdot u_3 \cdot \rho^6 + u_4 \\
 u_1 &= \left(B_0 \cdot R \cdot T - A_0 - \frac{C_0}{T^2} + \frac{D_0}{T^3} - \frac{E_0}{T^4} \right) \\
 u_2 &= \left(b \cdot R \cdot T - a - \frac{d}{T} \right) \quad . [20] \\
 u_3 &= \left(a + \frac{d}{T} \right) \\
 u_4 &= \frac{c \cdot \rho^3}{T^2} \cdot (1 + \gamma_0 \cdot \rho^2) \cdot \exp(-\gamma_0 \cdot \rho^2)
 \end{aligned}$$

Gleichung [20] gibt hier die BWR-Gleichung mit einer Erweiterung nach Starling wieder, die sogenannte BWRS-Gleichung. Die BWR-Gleichung ergibt sich daraus mit den Vereinfachungen, daß D_0 , E_0 und d gleich Null sind. Die zum Einsatz der beiden Gleichungen notwendigen Parameter sind für weite Temperatur- und Druckbereiche über die Literatur zugänglich [18, 24].

Tab 11: Kubische Zustandsgleichungen [1]

$P = \frac{R \cdot T}{V - b} - \frac{a}{V^2 + u \cdot b \cdot V + w \cdot b^2} \quad [21]$				
Gleichung	u	w	b	a
Van der Waals	0	0	$\frac{R \cdot T_c}{8 \cdot P_c}$	$\frac{27 \cdot R^2 \cdot T_c^2}{64 \cdot P_c}$
Redlich-Kwong	1	0	$\frac{0.08664 \cdot R \cdot T_c}{P_c}$	$\frac{0.42748 \cdot R^2 \cdot T_c^{2.5}}{P_c \cdot T^{0.5}}$
Soave	1	0	$\frac{0.08664 \cdot R \cdot T_c}{P_c}$	$\frac{0.42748 \cdot R^2 \cdot T_c^2}{P_c} \cdot (1 + f\omega \cdot (1 - T_r^{0.5}))^2$ $f\omega = 0.48 + 1.574 \cdot \omega - 0.176 \cdot \omega^2$
Peng-Robinson	2	-1	$\frac{0.07780 \cdot R \cdot T_c}{P_c}$	$\frac{0.45724 \cdot R^2 \cdot T_c^2}{P_c} \cdot (1 + f\omega \cdot (1 - T_r^{0.5}))^2$ $f\omega = 0.37464 + 1.54226 \cdot \omega - 0.26992 \cdot \omega^2$

($T_r = T/T_c$)

Auch wenn obige Zustandsgleichungen recht erfolgreich eingesetzt wurden, so zeigen sie dennoch Schwächen. Werden ihre Parameter zum Beispiel an Dampfdruckdaten angepaßt, so ist damit eine Vorhersage dieser Drücke möglich. Andere Größen, wie zum Beispiel die Exzeßenthalpie, können mit diesen Parametern jedoch nicht gleichzeitig mit adäquater Genauigkeit vorhergesagt werden. Weiterhin sollten sie nicht außerhalb des angepaßten Parameterbereiches zur Extrapolation und Vorhersage benutzt werden. Auch sind die Gleichungen nur für unpolare bis wenig polare Substanzen einsetzbar. Sie sind nicht in der Lage, die Abweichungen stark polarer Substanzen, wie z.B. Essigsäure, die selbst im Gasraum zur Dimerisierung neigt, richtig wiederzugeben. Dazu sind teils recht aufwendige Theorien notwendig. Ein Ansatz ist die in [7] beschriebene chemische Theorie.

Je nach betrachtetem PVT-Bereich wird bei der Berechnung des molaren Volumens bzw. der molaren Dichte mit einer der verwendeten Zustandsgleichungen eine Entscheidung in bezug auf das Zweiphasengebiet erforderlich. Im zweiphasigen Bereich konden-

siert komprimiertes Gas bzw. verdampft Flüssigkeit. Aber sowohl kubische Gleichungen als auch die BWR-Gleichung beschreiben diesen Übergang nicht mit einem horizontalen Verlauf der PV-Isothermen. Vielmehr treten in dem betrachteten Bereich zwei oder mehr Extrema der PVT-Kurve auf (s. Abb. 7). Die Schnittpunkte der Isothermen mit der begrenzenden Kurve des Zweiphasengebietes geben das spezifische Volumen des Gases bei der Kondensation bzw. der Flüssigkeit beim Verdampfen wieder. Sie sind gleichzeitig die zwei physikalisch sinnvollen Lösungen der Zustandsgleichung für diesen Bereich.

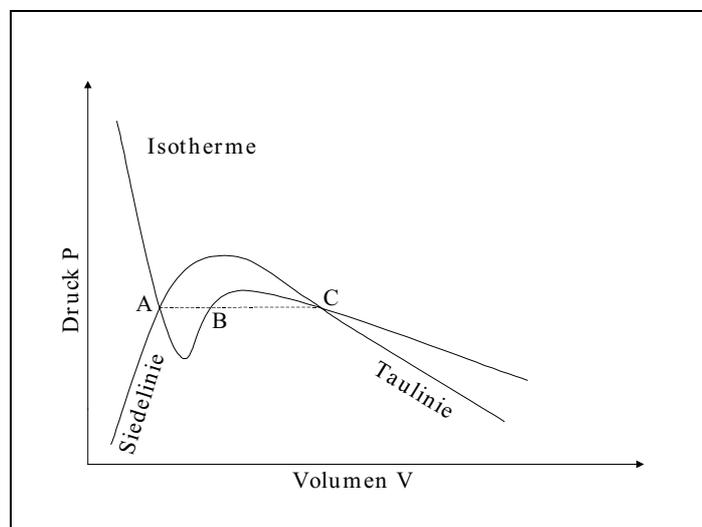


Abb. 7: Schematischer Verlauf der Isothermen im PV-Diagramm einer kubischen Gleichung im Zweiphasengebiet

Weist eine kubische Gleichung drei mögliche Lösungen (A, B, C) für ein definiertes Druck- und Temperatur-Verhältnis auf, so gibt die kleinste reale Lösung (A) das molare Volumen der flüssigen Phase wieder. Die größte reale Lösung (C) repräsentiert das molare Volumen der Gasphase. Eine mögliche dritte, intermediäre Lösung (B) hat keine physikalische Bedeutung. Ein ähnliches Vorgehen wie bei kubischen Gleichungen ist auch bei der Ermittlung des molaren Volumens bzw. der Dichte mittels der BWR-Gleichung möglich.

Durch Lösen der Zustandsgleichung wird die zur vollständigen Beschreibung des Systems notwendige dritte thermodynamische

Größe, das molare Volumen bzw. die molare Dichte, erhalten. Der Kompressibilitätskoeffizienten Z ist gemäß Gleichung [22] definiert [1]:

$$Z = \frac{P \cdot V}{R \cdot T} . \quad [22]$$

Der Kompressibilitätskoeffizient Z stellt somit ein Maß für die Abweichung von der Idealität der betrachteten Substanz dar. Für $Z = 1$ geht die Formel über in das ideale Gasgesetz.

Alle programmierten Gleichungssysteme bieten neben der Berechnung von Reinstoffen auch die Möglichkeit, Gemische zu beschreiben. Dazu ist es erforderlich, durch Mischungsregeln aus den Reinstoffparametern die Mischungsparameter zu bestimmen.

Die Mischungsregeln für die zweiparametrischen kubischen Zustandsgleichungen sind [1]:

$$a_m = \sum_{i=1}^N \sum_{j=1}^N y_i \cdot y_j \cdot (a_i \cdot a_j)^{0.5} \cdot (1 - k_{ij}) \quad [23]$$

$$b_m = \sum_{i=1}^N y_i \cdot b_i . \quad [24]$$

wobei a_i und b_i sich über die kritischen Größen Temperatur, Druck und den azentrischen Faktor berechnen lassen (s. Tab. 11). k_{ij} stellt hier einen binären Interaktionsparameter dar.[1]

Die ursprünglich für die Parameter der BWR-Gleichung vorgeschlagene Mischungsregel lautet [6]

$$M_0 = \left(\sum_{i=1}^N y_i \cdot M_i^{\frac{1}{r}} \right)^r . \quad [25]$$

M_0 beschreibt hier die Mischungsgröße des betrachteten Reinstoffparameters M_i . Dabei gelten die Zuordnungen in Tabelle 12.

Tab 12: Zuordnung der Reinstoffparameter der BWR-Gleichung für die Mischungsregel [25]

r	M_i	M_m
1	B_i	B_m
2	A_i, C_i, γ_i	A_m, C_m, γ_m
3	a_i, b_i, c_i, α_i	a_m, b_m, c_m, α_m

Für die Mischungsregeln der BWRS-Gleichung siehe Starling [18].

Die Bestimmung von Mischungsgrößen beim Einsatz der hier diskutierten Zustandsgleichungen ist aufwendiger und fehlerträchtiger als bei idealen Größen (s. Kap. 3.1). Daher ist deren Berechnung in allen Programmen integriert, die auf der Lösung obiger Zustandsgleichungen basieren.

Das Programm *VOLCUBIC* ermöglicht die Berechnung des molaren Volumens von Reinsubstanzen oder Gemischen bei n Temperaturen und m Drücken. Je nachdem wieviele Datenpunkte berechnet werden sollen, gibt *VOLCUBIC* eine Matrix, einen Vektor oder einen skalaren Wert aus. Da die Argumentverarbeitung entsprechend dem Vorgehen bei der Bestimmung der Temperaturabhängigkeit der Enthalpie organisiert ist, resultiert eine analoge Eingabestruktur (s. Kap. 3.2).

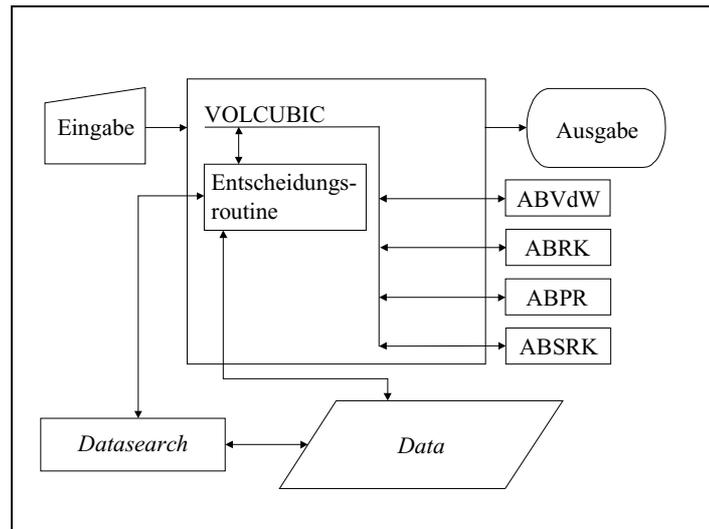


Abb. 8: Fließschema des Programmes *VOLCUBIC*

Mit dem Programm *VOLCUBIC* werden die in Tabelle 11 (s. S. 52) definierten kubischen Gleichungen abgedeckt. Mit einem Schalterargument kann der Benutzer zwischen den vier kubischen Gleichungen wählen. Die zum Lösen der Gleichungen erforderlichen Parameter a und b werden über die separaten Funktionen *ABVdW*, *ABRK*, *ABPR* und *ABSRK* berechnet. Diese Implementierung als eigenständige Funktionen ermöglicht es dem Benutzer, die Parameter auch unabhängig von der Lösung der kubischen Gleichungen zu berechnen und anderweitig einzusetzen.

Mit einem weiteren Schalterargument kann der Benutzer entscheiden, welches Ergebnis im zweiphasigen Gebiet berechnet werden soll. Wird für den Schalter eine Null eingegeben, so werden beide Ergebnisse in einem *cell array* ausgegeben. Wird jedoch nur eine Lösung gefunden, so gibt *VOLCUBIC* ausschließlich diese aus. Damit kann der Benutzer mittels geeigneter Kriterien selbst eine Entscheidung treffen, welche Lösung der Gleichung zu welcher Phase gehört [25]. Gibt der Benutzer für diesen Schalter eine Eins ein, so berechnet *VOLCUBIC* das molare Volumen der flüssigen Phase. Durch Eingabe des Wertes minus eins für diesen Schalter wird das molare Volumen der Gasphase bestimmt.

Tab 13: I/O-Charakteristik der Funktion *VOLCUB* und *VOLCUBIC*

<i>VOLCUBIC</i> :	Berechnung des molaren Volumens durch Lösen kubischer Gleichungen
Eingabe:	Temperatur, Druck, Vektor der Molenbrüche, <u>n Namen bzw. Summenformeln der beteiligten Spezies in Stringformat</u> , <u>Vektor der auf Data verweisenden Indizes</u> , Namen bzw. Summenformeln-Schalter , <u>krit. Temperatur</u> , <u>krit. Druck</u> , <u>krit. azentrischer Faktor</u> , binäre Interaktionsparameter , Phasen Schalter , Gleichungsschalter , Referenzdruck , Gaskonstante
Ausgabe:	Je nach Größe von T, P und x Matrix, Vektor oder Skalar des molaren Volumens
<i>VOLCUB</i> :	Berechnung des molaren Volumens durch Lösen kubischer Gleichungen
Eingabe:	Temperatur, Druck, Vektor der Molenbrüche, krit. Temperatur, krit. Druck, krit. azentrischer Faktor, binäre Interaktionsparameter, Phasen Schalter, Gleichungsschalter, Gaskonstante , Referenzdruck , kubische Parameter a und b
Ausgabe:	Je nach Größe von T, P und x Matrix, Vektor oder Skalar des molaren Volumens

(**Fette Schrift** = optionale Argumente, unterstrichene Schrift = alternative Eingabemöglichkeit, siehe weitere Erläuterungen im Text und Kap. 1.2)

Da das Volumen bei einigen Berechnungen sehr oft bestimmt werden muß, wurde das Programm *VOLCUB* implementiert. Es ermöglicht dieselben Berechnungen wie das Programm *VOLCUBIC*. Dabei wird nicht die aufwendige Entscheidungsroutine von *VOLCUBIC* zur Verfügung gestellt. *VOLCUB* greift auch nicht selbstständig auf *data* zu. Der Benutzer muß alle notwendigen Daten selbst eingeben.

In Tabelle 14 sind die mit Hilfe der Befehle *tic/toc* ermittelten durchschnittlichen Zeiten von tausend Programmläufen verglichen. Dabei zeigt sich, daß das Programm *VOLCUB* je nach verwendeten Argumenten bis zu 2.9 mal schneller ist als das Programm

VOLCUBIC. In Tabelle 14 sind die Zeiten der Aufrufe beider Funktionen unter Anwendung der kritischen Größen miteinander verglichen.

Tab 14: Über tausend Funktionsaufrufe gemittelte Zeit [s] für einen Programmaufruf.

Übergebene Argumente	<i>VOLCUBIC</i>	<i>VOLCUB</i>	Zeitvergleich
T_c, P_c, Ω_c	0.0219 s	0.0073 s	2.95

Weiterhin ermöglicht *VOLCUB* die Benutzung eigener nicht aus den kritischen Daten berechneter Parameter a und b (s. Tab. 11). Dies kann insbesondere bei der Anpassung der Parameter an spezielle Datensätze hilfreich sein. Zu diesem Zweck wurde die Funktion *ABMIX* programmiert. Sie wendet die Mischungsregeln kubischer Gleichungen (s. Gl. [23] und [24]) auf die manuell eingegebenen Reinstoffparameter a und b an. *ABMIX* wird - wie auch *ABVdW*, *ABRK*, *ABSRK* und *ABPR* - von *VOLCUB* als externe Funktion aufgerufen; sie ist damit auch unabhängig von *VOLCUB* und *VOLCUBIC* zugänglich.

Tab 15: I/O-Charakteristik der Funktion *ABVdW*, *ABRK*, *ABSRK*, *ABPR* und *ABMIX*

<i>ABVdW</i> , <i>ABRK</i> , <i>ABSRK</i> , <i>ABPR</i> :	Berechnung kubischer Konstanten für Reinstoff und Mischung entsprechend den Gleichungen nach Van-der-Waals, Redlich-Kwong, Soave-Redlich-Kwong und Peng-Robinson
Eingabe:	Temperatur, Vektor der Molenbrüche, krit. Temperatur, krit. Druck, krit. azentrischer Faktor, binäre Interaktionsparameter, Gaskonstante
Ausgabe:	Je nach Größe von T und x Matrix, Vektor oder Skalar der Reinstoff- und Mischungsparameter der spezifizierten kubischen Gleichung

Tab 15: I/O-Charakteristik der Funktion *ABVdW*, *ABRK*, *ABSRK*, *ABPR* und *ABMIX* (Fortsetzung)

<i>ABMIX</i> :	Berechnung kubischer Konstanten für Mischungen aus den Konstanten reiner Stoffe
Eingabe:	Vektor der Molenbrüche, Vektoren der Reinstoffparameter, binäre Interaktionsparameter
Ausgabe:	Mischungsparameter

(**Fette Schrift** = optionale Argumente)

Die Lösung der kubischen Gleichungen kann mit der von MATLAB zur Verfügung gestellten Funktion *ROOTS* einfach und komfortabel verwirklicht werden. Zur Anwendung der Funktion *ROOTS* ist es erforderlich, die zu lösende Gleichung in eine polynome Form zu bringen. Mit Gleichung [26] wird ein Polynom dritten Grades in Z erhalten. Mit der Funktion *ROOTS* werden die Wurzeln von Gleichung [26] erhalten

$$Z^3 - A \cdot Z^2 + B \cdot Z + C = 0$$

mit

$$A = 1 + B^* - u \cdot B^*$$

$$B = A^* + w \cdot B^{*2} - u \cdot B^* - u \cdot B^{*2} \quad [26]$$

$$C = -A^* B^* - w \cdot B^{*2} - w \cdot B^{*3}$$

$$\text{wobei } A^* = \frac{a \cdot P}{R^2 \cdot T^2} \quad B^* = \frac{b \cdot P}{R \cdot T}$$

Demgegenüber erweist sich das Lösen der BWR-Gleichung als aufwendiger. So beschränken sich z.B. Johnson und Colver [16] auf eine iterative Methode, die etwaig auftretende Extrema gänzlich vernachlässigt. Mills, Wills und Bhirud [26] verwenden Newton-Raphson-Methoden erster und zweiter Ordnung, um mit der BWRS-Gleichung die Dichte eines Gemisches zu bestimmen. Auch MATLAB bietet keine Möglichkeit, alle Wurzeln der BWR-Gleichung direkt zu bestimmen [34].

In dem File *VBWR* wird eine kombinierte Strategie verfolgt. Zunächst werden mit einer aus den Numerical Recipes [33] entnommenen Routine die auftretenden Extrema der Gleichung lokalisiert. Dazu wird ein Gitter zu untersuchender Punkte über den in Frage stehenden Druck-/Dichtebereich gelegt. Dabei kann der Benutzer sowohl die Bereichsgrenzen definieren als auch durch die Anzahl der Gitterpunkte die Gitterweite bestimmen. Die Routine sucht nun in den sich aus der Gitterweite ergebenden Intervallen nach einer Vorzeichenänderung des Terms $\partial P/\partial \rho$. Ein mögliches Extremum wird somit durch die es einschließenden Intervallgrenzen definiert.

Die damit grob lokalisierten Punkte werden dann unter Einsatz der MATLAB Funktion *CONSTR* genau ermittelt. Diese Funktion ermöglicht die Minimierung einer in einem M-File abgelegten Funktion, wobei *CONSTR* gleichzeitig benutzerdefinierte Grenzen vorsieht, welche das Optimierverfahren nicht über- bzw. unterschreitet. Des weiteren sieht *CONSTR* die Angabe von Nebenbedingungen vor. Dabei können diese als Gleichungen oder Ungleichungen in dem betrachteten M-File abgelegt werden.

Als Beispiel für die Anwendung von *CONSTR* sollen Werte für x gefunden werden, bei denen die Funktion $f(x) = -x_1 x_2 x_3$ ein Minimum aufweist [45]. Startwert der Minimierung sind die Werte $x = [10, 10, 10]$. Dabei sollen noch zwei Nebenbedingungen beachtet werden:

- $x_1 - 2x_2 - 2x_3 \leq 0$ und
- $x_1 + 2x_2 + 2x_3 \leq 72$ bzw. $x_1 + 2x_2 + 2x_3 - 72 \leq 0$.

Für diese Aufgabe werden die Gleichungen zunächst in Form eines M-Files abgelegt:

```
function [f,g] = fun(x)
f = -x(1)*x(2)*x(3);
g(1) = -x(1)-2*x(2)-2*x(3);
g(2) = x(1)+2*x(2)+2*x(3)-72; .
```

Das M-File wird im MATLAB Fenster mit der Optimieroutine *CONSTR* aufgerufen.

```
» x = constr('fun', [10,10,10])  
  
» x =  
  
    24.0000    12.0000    12.0000
```

Die Werte der sich ergebenden Nebenbedingungen lassen sich durch einen weiteren Aufruf von *fun* mit den gefundenen Werten für *x* erhalten.

```
» [f,g] = fun(x)  
f =  
    -3.4560e+03  
g =  
    -72     0
```

Der *CONSTR* zugrunde liegende Algorithmus ist eine Methode der sequentiellen quadratischen Programmierung [45]. Dabei handelt es sich um eine Weiterentwicklung der Methode des nicht-linearen Programmierens [68].

Mit Kenntnis der Lage möglicher Extrema kann nun entschieden werden, in welchem Druckbereich nach dem Schnittpunkt der Druck-/Dichte-Kurve der BWR-Gleichung mit der Isobaren des Systemdruckes gesucht werden soll. Je nach der Lage oder nach der Entscheidung des Benutzers werden dann alle gefundenen und signifikanten Schnittpunkte ausgegeben. In einem in Abbildung 9 dargestellten Verlauf kann sich der Benutzer den linken, den rechten oder beide Schnittpunkte der P/ρ -Kurve mit der Isobaren ausgeben lassen.

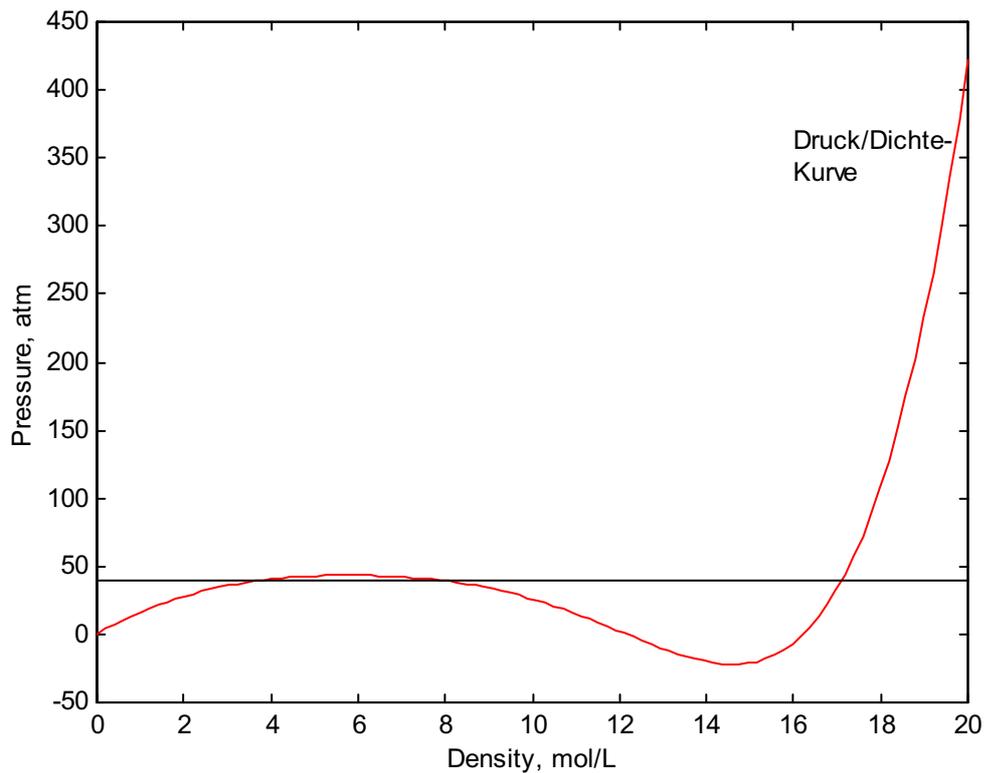


Abb. 9: Verlauf der P/ρ -Kurve für ein Gemisch aus Methan, Ethan und n-Dekan (230 K/40 bar), gemäß der BWRS-Gleichung

Das Anwendungsbeispiel einer gasförmigen Mischung aus 40 % Methan und 60 % Propan zeigt die gute Übereinstimmung der berechneten Werte mit den Literaturwerten.

Tab 16: Vergleich durch *VBWR* berechneter Werte mit Literaturwerten [16].

Temperatur [°C]	Druck [bar]	$V_{\text{calc.}}$ [cm^3/mol]	$V_{\text{Lit.}}$ [cm^3/mol]
70.11	27.56	865.23	865.27
70.11	41.35	510.96	511.0
236.78	27.56	1486.89	1486.90
236.78	41.35	975.92	975.93

Wie auch die Funktion *VOLCUB* berechnet die Funktion *VBWR* das molare Volumen von fluiden Reinstoffen oder fluiden Gemischen bei n Temperaturen und m Drücken. Dabei muß der Benutzer ne-

ben der Zusammensetzung, den Temperaturen und Drücken auch die Parameter der BWR-Gleichung eingeben. Ähnlich dem File *VOLCUB* muß der Benutzer bei *VBWR* jedoch alle Eingaben selbst durchführen.

Tab 17: I/O-Charakteristik der Funktion *VBWR*

<i>VBWR</i> :	Berechnung des molaren Volumens mittels der BWRS bzw. der BWR Gleichung mit möglicher graphischer Ausgabe der Ergebnisse
Eingabe:	Molenbrüche, Temperatur, Druck, Vektoren aller Reinstoffparameter, binäre Interaktionsparameter, Phasenschalter, Gaskonstante, Anzahl der zu untersuchenden Gitterpunkte, obere und untere Iterationsgrenzen, Vektor der Integrations Optionen, graphischer Ausgabeschalter
Ausgabe:	Je nach Größe von T, P und x Matrix, Vektor oder Skalar des molaren Volumens mit möglicher graphischer Ausgabe der Verläufe von P/ρ

(**Fette Schrift** = optionale Argumente)

VBWR ermöglicht dem Benutzer neben der Berechnung des molaren Volumens mittels der BWR-Gleichung auch die Anwendung der Gleichungserweiterung nach Starling. Allein die Belegung von drei zusätzlichen Argumenten beim Funktionsaufruf entscheidet über die zu berechnende Zustandsgleichung. Werden alle der in Gleichung [20] angegebenen Parameter beim *VBWR* Aufruf übergeben, so verwendet *VBWR* die BWRS-Gleichung für die Berechnung. Werden beim Funktionsaufruf die Argumente, die die Parameter D_0 , E_0 und d enthalten, nicht mit Werten belegt, so berechnet *VBWR* das molare Volumen mittels der BWR-Gleichung:

$$\begin{aligned}
P &= \rho \cdot R \cdot T + u1 \cdot \rho^2 + u2 \cdot \rho^3 + \alpha_0 \cdot u3 \cdot \rho^6 + u4 \\
u1 &= \left(B_0 \cdot R \cdot T - A_0 - \frac{C_0}{T^2} + \frac{D_0}{T^3} - \frac{E_0}{T^4} \right) \\
u2 &= \left(b \cdot R \cdot T - a - \frac{d}{T} \right) \quad . [20] \\
u3 &= \left(a + \frac{d}{T} \right) \\
u4 &= \frac{c \cdot \rho^3}{T^2} \cdot (1 + \gamma_0 \cdot \rho^2) \cdot \exp(-\gamma_0 \cdot \rho^2)
\end{aligned}$$

Zur Behandlung der BWR-Parameter stehen die Funktionen *ABBWR* und *ABBWRMIX* zur Verfügung. *ABBWR* ermöglicht die Berechnung von Reinstoff- oder Mischungsparametern der BWR-Gleichung mittels der Generalisierung nach Han-Starling (s. Gl. [27])

$$\begin{aligned}
\rho_{ci} \cdot B_{0i} &= A_1 + B_1 \cdot \varpi_i ; \quad \frac{\rho_{ci} \cdot A_{0i}}{R \cdot T_{ci}} = A_2 + B_2 \cdot \varpi_i \\
\frac{\rho_{ci} \cdot C_{0i}}{R \cdot T_{ci}^3} &= A_3 + B_3 \cdot \varpi_i ; \quad \rho_{ci}^2 \cdot \gamma_i = A_4 + B_4 \cdot \varpi_i \\
\rho_{ci}^2 \cdot b_i &= A_5 + B_5 \cdot \varpi_i ; \quad \frac{\rho_{ci}^2 \cdot a_i}{R \cdot T_{ci}} = A_6 + B_6 \cdot \varpi_i \\
\rho_{ci}^3 \cdot \alpha_i &= A_7 + B_7 \cdot \varpi_i ; \quad \frac{\rho_{ci}^2 \cdot c_i}{R \cdot T_{ci}^3} = A_8 + B_8 \cdot \varpi_i \\
\frac{\rho_{ci} \cdot D_{0i}}{R \cdot T_{ci}^4} &= A_9 + B_9 \cdot \varpi_i ; \quad \frac{\rho_{ci}^2 \cdot d_i}{R \cdot T_{ci}^2} = A_{10} + B_{10} \cdot \varpi_i \\
\frac{\rho_{ci} \cdot E_{0i}}{R \cdot T_{ci}^5} &= A_{11} + B_{11} \cdot \varpi_i \cdot \exp(-3.8 \cdot \varpi_i)
\end{aligned} \quad , \quad [27]$$

wobei für die Parameter A_{1-11}/B_{1-11} die Werte in Tabelle 18 gelten.

Tab 18: Parameter zur Berechnung der BWRS-Parameter nach Han-Starling [6].

Indexnummer j	Parameter A _j	Parameter B _j
1	0.443690	0.115449
2	1.28438	-0.920731
3	0.356306	1.70871
4	0.544979	-0.270896
5	0.528629	0.349261
6	0.484011	0.754130
7	0.0705233	-0.044448
8	0.504087	1.32245
9	0.0307452	0.179433
10	0.0732828	0.463492
11	0.006450	-0.022143

ABBWRMIX hingegen berechnet ausschließlich die BWR-Parameter einer Mischung aus den entsprechenden Parametern der reinen Substanzen.

Tab 19: I/O-Charakteristik der Funktionen *ABBWR* und *ABBWRMIX*

<i>ABBWR</i> :	Berechnung der Mischungs- oder Reinstoffkonstanten für BWRS- und BWR-Gleichungen
Eingabe:	Molenbrüche, krit. Temperatur, krit. Dichte, azentrische Faktoren, binäre Interaktionsparameter, Gaskonstante
Ausgabe:	BWR-Parameter einer Mischung bzw. reiner Substanzen
<i>ABBWRMIX</i> :	Berechnung der Mischungskonstanten für BWRS- und BWR- Gleichungen
Eingabe:	Vektor der Molenbrüche, Vektoren aller Reinstoffparameter, binäre Interaktionsparameter, Gaskonstante
Ausgabe:	Mischungsparameter

(**Fette Schrift** = optionale Argumente)

Im Folgenden werden verschiedene Anwendungen vorgestellt, die auf der Lösung von Zustandsgleichungen basieren.

4.2 Abweichungsfunktionen

Zur thermodynamischen Beschreibung des Verhaltens eines realen Systems werden Abweichungsfunktionen herangezogen. Diese geben den "Überschuß" der real ermittelten Zustandsfunktion über die ideal berechnete Zustandsfunktion an [77]. Somit stellen Abweichungsfunktionen ein wichtiges Bindeglied zwischen den Betrachtungen aus Kapitel 3 und dem realen Fluidverhalten dar.

Ist Γ eine thermodynamische Größe bei einem bestimmten Temperatur- und Druckverhältnis und ist weiterhin Γ° der Wert dieser Größe bei gleicher Temperatur, beim Referenzdruck P° und im idealen Gaszustand, dann ergibt sich eine Abweichungsfunktion mit $\Gamma - \Gamma^\circ$ oder $\Delta \Gamma$.

$$\Gamma = \Gamma^\circ + \Delta \Gamma \quad [28]$$

Für die Festlegung des Referenzzustandes bieten sich vier gängige Möglichkeiten an. Dabei werden P° bzw. V° als konstant angesehen oder $P^\circ = P$ bzw. $V^\circ = V$ gesetzt. P° und V° sind über die Gleichung für ein ideales Gas miteinander verknüpft [1].

Bei den in Gleichung [28] verwendeten thermodynamischen Größen handelt es sich um molare Größen.

Wie bereits ausgeführt, können Abweichungsfunktionen mittels Zustandsfunktionen (s. Kap. 4.1) ausgedrückt werden. Dazu werden die Abweichungsfunktionen für die Helmholtz Energie A und die Entropie S abgeleitet:

$$A - A^\circ = - \int_{\infty}^V \left(P - \frac{R \cdot T}{V} \right) dV - R \cdot T \cdot \ln \frac{V}{V^\circ} \quad [29]$$

$$S - S^0 = - \int_{\infty}^V \left(\left(\frac{\partial P}{\partial T} \right)_V - \frac{R}{V} \right) dV - R \cdot \ln \frac{V}{V^0}. \quad [30]$$

Über diese Abweichungsfunktionen und den Kompressibilitätsfaktor Z sind auch andere thermodynamische Größen zugänglich.

$$\begin{aligned} H - H^0 &= (A - A^0) + T \cdot (S - S^0) + R \cdot T \cdot (Z - 1) \\ U - U^0 &= (A - A^0) + T \cdot (S - S^0) \\ G - G^0 &= (A - A^0) + R \cdot T \cdot (Z - 1) \end{aligned} \quad [31]$$

In den Funktionen *DEPFUNCCUB* und *DEPFUNCBWR* werden die Gleichungen [29] und [30] mittels druckexpliziter Zustandsgleichungen (s. Abschnitt 4.1) gelöst. Neben den Abweichungsfunktionen der Entropie und der Helmholtz Energie wird auch der Kompressibilitätskoeffizient Z berechnet, um eine direkte Ableitung gemäß den Gleichungen [31] zu ermöglichen.

Mit der Funktion *DEPFUNCCUB* können die Berechnungen der genannten Abweichungsfunktionen durch vier unterschiedliche kubische Gleichungen (s. Tab. 10) erfolgen. Diese Einteilung resultiert aus dem Vorgehen bei der Implementierung der Berechnung von Zustandsgleichungen mit der Funktion *VOLCUB* in Kapitel 4.1. Analog der Programmierung dieser Funktion kann der Benutzer auch bei *DEPFUNCCUB* durch ein Schalterargument zwischen den vier Zustandsgleichungen wählen. Es können die Abweichungsfunktionen für ein Gemisch oder einen Reinstoff bei n Temperaturen und m Drücken bestimmt werden. Ähnlich verhält es sich mit der Funktion *DEPFUNCBWR*. Über sie sind die in *VBWR* implementierten Zustandsgleichungen für die Berechnung von Abweichungsfunktionen zugänglich.

Eigene kubische Parameter kann der Benutzer in *DEPFUNCCUB* analog dem Vorgehen bei *VOLCUB* eingeben.

In diesem Zusammenhang ist hervorzuheben, daß zur Berechnung der Abweichungsfunktion der Entropie (s. Gl. [30]) die partielle Ableitung der druckexpliziten Zustandsgleichung nach der Temperatur erforderlich ist. Somit muß auch die Abhängigkeit des kubischen Parameters a von der Temperatur bekannt sein.

Für die kubischen Gleichungen nach Soave und nach Peng-Robinson finden sich in der Literatur Ausdrücke zur Berechnung eines Differentialquotienten, $\partial a/\partial T$ [1]. Diese basieren auf der Kenntnis der kritischen Temperatur, des kritischen Druckes und des azentrischen Faktors. Die Kombination von manuell eingegebenen kubischen Parametern mit der Berechnung von $\partial a/\partial T$ über die kritischen Größen würde einem Mischansatz entsprechen.

Um dies zu vermeiden, kann der Benutzer ein M-File erstellen, welches die Berechnung des Differentialquotienten ausführt. Beim Aufruf von *DEPFUNCCUB* können der Name und etwaige zusätzliche Parameter dieses M-Files übergeben werden. Damit kann der Benutzer die Temperaturabhängigkeit des kubischen Parameters a über eigene mathematische Zusammenhänge wiedergeben. Wird kein M-Filename angegeben, so verwendet *DEPFUNCCUB* den entsprechenden Ausdruck gemäß Soave oder Peng-Robinson zur Berechnung.

Tab 20: I/O-Charakteristik der Funktionen *DEPFUNCCUB* und *DEPFUNCBWR*

<i>DEPFUNCCUB:</i>	Berechnung der Entropie- und Helmholtzenergie-Abweichungsfunktionen und des Kompressibilitätskoeffizienten Z mittels kubischer Zustandsgleichungen
Eingabe:	Temperatur, Druck, Molenbrüche, <u>krit. Temperatur</u> , <u>krit. Druck</u> , <u>krit. azentrischer Faktor</u> , binäre Interaktionsparameter, Phasenschalter, Gleichungsschalter, Gaskonstante , Referenzdruck , kubische Parameter a und b , Funktionsname zur Berechnung von $\partial a/\partial T$, notwendige Parameter zur Berechnung von $\partial a/\partial T$
Ausgabe:	Je nach Größe von T, P und x Matrix, Vektor oder Skalar des molaren Volumens
<i>DEPFUNCBWR:</i>	Berechnung der Entropie- und Helmholtzenergie-Abweichungsfunktionen und des Kompressibilitätskoeffizienten z mittels der BWRS- bzw. BWR-Zustandsgleichungen
Eingabe:	Temperatur, Druck, Molenbrüche, Reinstoffparameter, binäre Interaktionsparameter, Phasenschalter, Gaskonstante , Anzahl der zu untersuchenden Gitterpunkte , obere und untere Iterationsgrenzen , Vektor der Integrations Optionen , graphischer Ausgabeschalter
Ausgabe:	Je nach Größe von T, P und x Matrix, Vektor oder Skalar des molaren Volumens

(**Fette Schrift** = optionale Argumente, unterstrichene Schrift = alternative Eingabemöglichkeit, siehe weitere Erläuterungen im Text und Kap. 1.2)

DEPFUNCCUB und *DEPFUNCBWR* stellen die zentralen Funktionen zur Berechnung der Abweichung des Verhaltens eines Gases vom Verhalten des idealen Gases dar. Ein Vergleich experimenteller Werte mit Ergebnissen aus Berechnungen mit der Funktion *DEPFUNCCUB* wird hier indirekt durchgeführt. Verglichen werden Werte der unter Einsatz von Gleichung [31] erhaltenen Abweichungsfunktion der Enthalpie und die Abweichungsfunktion der Entropie.

Tab 21: Vergleich mit der Funktion *DEPFUNCCUB* berechneter Werte mit Literaturwerten [1].

	Literatur [1]	berechnet, SRK	berechnet, PR
H-H° [J/mol]	-1.0292*10 ⁴	-9.8936*10 ³	-9.9329*10 ³
S-S° [J/(mol*K)]	-59.64	-56.59	-56.22

Die gefundenen Fehler liegen mit Werten zwischen 3 und 5 % im Bereich der Ungenauigkeit der eingesetzten Zustandsgleichung.

4.3 Druckeinfluß auf H, G und S

Ein weiteres Anwendungsbeispiel für Zustandsgleichungen ist die Berechnung der Veränderung der Enthalpie, Entropie und der freien Gibbschen Enthalpie mit dem Druck. Vor allem bei höheren Drücken wird die Modellvorstellung des idealen Gasgesetzes den realen Gegebenheiten nicht mehr gerecht. Auch hier soll wieder die Berechnung der Enthalpieänderung beispielhaft für die Bestimmung des Druckeinflusses auf H, G und S behandelt werden.

Die Funktionen *Hpres/HP* und *HPresBWR* ermöglichen die Berechnung der Enthalpieänderung bei Variation des Druckes von Reinsubstanzen oder Gemischen. Dabei können sowohl der Druck als auch die Temperatur vektorieLL verarbeitet werden. Je nach der Zahl eingegebener Datenpunkte berechnen die Funktionen einen skalaren Wert, einen Vektor oder eine Matrix. Die Dimension der Matrix richtet sich nach der Zahl der veränderlichen Größen. Diese sind hier Temperatur, Druck und Zusammensetzung. Somit kann die Ergebnismatrix maximal dreidimensional werden.

Durch die in Abschnitt 4.2 diskutierten Abweichungsfunktionen wird auch die Argumentverarbeitung von *Hpres/HP* und *HPresBWR* bestimmt. *HP* bietet die zum Vorgehen bei *VOLCUB* analoge Möglichkeit der Eingabe von eigenen kubischen Parametern. Wie auch bei *DEPFUNCCUB* kann der Benutzer hier den Namen und die notwendigen Parameter eines Files zur Berechnung

der Temperaturabhängigkeit des kubischen Parameters a eingeben.

Tab 22: I/O-Charakteristik der Funktionen HP/H_{pres} und $HP_{presBWR}$

HP :	Berechnung der Exzeß-Enthalpie mittels kubischer Zustandsgleichungen
Eingabe:	Temperatur, Startdruck, Enddruck, Molenbrüche, krit. Temperatur, krit. Druck, krit. azentrischer Faktor, binäre Interaktionsparameter, Phasenschalter, Gleichungsschalter, Gaskonstante , Referenzdruck , kubische Parameter a und b , Funktionsname zur Berechnung von $\partial a/\partial T$, Parameter zur Berechnung von $\partial a/\partial T$
Ausgabe:	je nach Größe von T und P Matrix, Vektor oder Skalar der Exzeß-Enthalpie
H_{pres} :	Berechnung der Exzeß-Enthalpie mittels kubischer Zustandsgleichungen
Eingabe:	Temperatur, Startdruck, Enddruck, Molenbrüche, n Namen/Summenformeln der beteiligten Spezies in Stringformat, Vektor der auf <i>Data</i> verweisenden Indizes, Namen/Summenformeln - Schalter , binäre Interaktionsparameter , Phasenschalter , Gleichungsschalter , Referenzdruck , Gaskonstante , skalare/vektorierte krit. Temperatur , skalarer/vektorierte krit. Druck , skalarer/vektorierte krit. azentrischer Faktor , (jedoch ist eine der vorgenannten Eingaben zur Bestimmung zur berechnender Substanzen notwendig)
Ausgabe:	je nach Größe von T und P Matrix, Vektor oder Skalar der Exzeß-Enthalpie

(Fortsetzung auf nächster Seite)

Tab 22: I/O-Charakteristik der Funktionen *HP/Hpres* und *HPresBWR* (Fortsetzung)

<i>HpresBWR</i> :	Berechnung der Exzeß-Enthalpie mittels BWRs- bzw. BWR-Gleichung
Eingabe:	skalare/vektorierte Molenbrüche, skalare/vektorierte Temperatur, Startdruck, skalarer/vektorisierter Enddruck, skalare/vektorierte Reinstoffparameter, binäre Interaktionsparameter, <u>Gaskonstante</u>
Ausgabe:	je nach Größe von T und P Matrix, Vektor oder Skalar der Exzeß-Enthalpie
Die Funktionen <i>GP/GPres</i> , <i>GPresBWR</i> und <i>SP/SPres</i> , <i>SPresBWR</i> berechnen mit analogen Eingabemasken die entsprechenden thermodynamischen Exzeßgrößen.	

(**Fette Schrift** = optionale Argumente, unterstrichene Schrift = alternative Eingabemöglichkeit, siehe weitere Erläuterungen im Text und Kap. 1.2)

Hpres/HP und *HPresBWR* basieren auf der Berechnung der Abweichungsfunktionen in *DEPFUNCCUB* und *DEPFUNCBWR*. Durch den Aufruf dieser Funktionen werden die Werte für ΔS , ΔA und der zugehörige Wert der Kompressibilität Z erhalten. Mit den Gleichungen [31] sind dann die Abweichungsfunktionen der Enthalpie, der inneren Energie und der Gibbsschen Enthalpie zugänglich.

4.4 Fugazitätskoeffizienten

Aus reaktionstechnischer Sicht können Fugazitäten zum Beispiel bei der Betrachtung von Gasgleichgewichtsreaktionen relevant sein. Sind an den betrachteten Gleichgewichten Gase beteiligt, die stark vom Verhalten idealer Gase abweichen, so ist der Druck durch die Fugazität f zu ersetzen [79]. Der Fugazitätskoeffizient ϕ berücksichtigt die realen Wechselwirkungen der Gasmoleküle untereinander [77]. Daher soll die Berechnung von Fugazitätskoeffizienten als weiteres Einsatzgebiet von Zustandsgleichungen berücksichtigt werden. In der Literatur findet sich ein Ausdruck für den Fugazitätskoeffizienten ϕ_i in einer Gasmischung [1]

$$R \cdot T \cdot \ln \varphi_i = - \int_{\infty}^V \left[\left(\frac{\partial P}{\partial n_i} \right)_{T, V, n_{j \neq i}} - \frac{R \cdot T}{V} \right] dV - R \cdot T \cdot \ln Z. [32]$$

Gleichung [32] ermöglicht die Berechnung von Fugazitätskoeffizienten φ_i in einer Gasmischung. Dabei kommt eine druckexplizite Zustandsgleichung zur Lösung des Volumenintegrals in Gleichung [32] zur Anwendung [1]. Für diese Berechnung sind die Funktionen *FUGCUB/FUGCUBIC* und *FUGBWR* implementiert worden. Die zur Verwendung von Gleichung [32] notwendigen Lösungen von Zustandsgleichungen werden durch Aufrufe der Programme *VOLCUB/VOLCUBIC* und *VBWR* erhalten. Daher bestimmen diese Aufrufe die Eingabecharakteristiken der Funktionen zur Berechnung von Fugazitätskoeffizienten.

Tab 23: I/O-Charakteristik der Funktionen *FUGCUB* und *FUGCUBIC*

<i>FUGCUBIC</i> :	Berechnung der Fugazitätskoeffizienten durch kubische Gleichungen
Eingabe:	skalare/vektorierte Molenbrüche, skalare/vektorierte Temperatur, skalarer/vektorisierte Druck, n Namen/Summenformeln der beteiligten Spezies in String-format, skalare/vektorierte auf <i>Data</i> verweisenden Indizes , Namen-/Summenformeln – Schalter , (jedoch eine der vorgenannten Eingaben zur Bestimmung der zu berechnenden Substanzen ist notwendig), binäre Interaktionsparameter , Gleichungsschalter , Referenzdruck , Gaskonstante , krit. Temperatur , krit. Druck , krit. azentrischer Faktor
Ausgabe:	Je nach Größe von T, P und x Matrix, Vektor oder Skalar der Fugazitätskoeffizienten

Tab 23: I/O-Charakteristik der Funktionen *FUGCUB* und *FUGCUBIC* (Fortsetzung)

<i>FUGCUB</i> :	Berechnung der Fugazitätskoeffizienten durch kubische Gleichungen
Eingabe:	skalare/vektorierte Molenbrüche, skalare/vektorierte Temperatur, skalarer/vektorierter Druck, binäre Interaktionsparameter, Gleichungsschalter, <u>Gaskonstante</u> , <u>Referenzdruck</u> , <u>kubische Parameter a und b</u>
Ausgabe:	Je nach Größe von T, P und x Matrix, Vektor oder Skalar der Fugazitätskoeffizienten

(**Fette Schrift** = optionale Argumente, unterstrichene Schrift = alternative Eingabemöglichkeit, siehe weitere Erläuterungen im Text und Kap. 1.2)

Die Möglichkeit mit dem File *FUGCUB* eigene kubische Parameter für die Berechnung zu definieren, ist insbesondere bei den hier durchgeführten Berechnungen von großem Nutzen. Die Zustandsgleichung muß nicht nur in der Lage sein, die volumetrischen Eigenschaften der Gasmischung richtig wiederzugeben, sondern auch den Differentialquotienten aus Gleichung [32] [1].

Tab 24: I/O-Charakteristik der Funktion *FUGBWR*

<i>FUGBWR</i> :	Berechnung der Fugazitätskoeffizienten mittels der BWRS- bzw. BWR-Gleichung
Eingabe:	skalare/vektorierte Molenbrüche, skalare/vektorierte Temperatur, skalarer/vektorierter Druck, skalare/vektorierte Reinstoffparameter, binäre Interaktionsparameter, <u>Gaskonstante</u>
Ausgabe:	Je nach Größe von T, P und x Matrix, Vektor oder Skalar des molaren Volumens

(**Fette Schrift** = optionale Argumente, unterstrichene Schrift = alternative Eingabemöglichkeit, siehe weitere Erläuterungen im Text und Kap. 1.2)

Ein Vergleich der mit *FUGCUBIC* berechneten Fugazitätskoeffizienten mit Literaturdaten erfolgt an dem Beispiel einer gasförmigen Wasserstoff-Ammoniak-Mischung. Angewendet wird in diesem Beispiel die Zustandsgleichung nach Redlich-Kwong.

Tab 25: Vergleich berechneter Fugazitätskoeffizienten mit Literaturdaten [80] für H₂/NH₃

x(H ₂)	T in °C	P in bar	φ(H ₂ , Lit.)	φ(H ₂ , Calc.)
0.2	71.2	34.9019	1.1880	1.1424
0.356	71.4	35.2530	1.1250	1.0912
0.496	71.4	35.0960	1.0820	1.0594
0.642	71.4	35.2910	1.0430	1.0378
0.705	71.4	35.3090	1.0360	1.0311
0.841	71.4	34.8330	1.0220	1.0212

Der Fehler liegt mit bis zu 3 Prozent im Genauigkeitsbereich der verwendeten kubischen Zustandsgleichung.

4.5 Wärmekapazität realer Gase

Eine weitere oft als Exzeßgröße angegebene Eigenschaft realer Gase ist die Wärmekapazität. Gleichung [33] gibt den entsprechenden Zusammenhang wieder:

$$C_p = C_p^0 + \Delta C_p. \quad [33]$$

Die Exzeß-Wärmekapazität ΔC_p ergibt sich gemäß Gleichung [34]

$$C_p - C_p^0 = T \cdot \int_{\infty}^V \left(\frac{\partial^2 P}{\partial T^2} \right) dV - \frac{T(\partial P / \partial T)_V^2}{(\partial P / \partial V)_T} - R. \quad [34]$$

Durch Einsetzen der in Kapitel 4.1 diskutierten kubischen Zustandsgleichungen und anschließende Berechnung des Volumenintegrals werden Ausdrücke erhalten, die in den Funktionen *DEPARTCUB* und *DEPARTCUBIC* zur Berechnung der Exzeßwär-

mekapazität zur Anwendung kommen. Dabei sind die erhaltenen Ausdrücke für ΔC_p Funktionen des molaren Volumens, welches durch einen Aufruf der Programme *VOLCUB* bzw. *VOLCUBIC* bestimmt wird.

Für die Berechnung von ΔC_p mit der BWR-Gleichung findet sich bei Sledjeski [30] ein expliziter Ausdruck in Abhängigkeit von der Dichte und der Temperatur. Dieser Zusammenhang wird in dem File *DEPARTCPBWR* verwendet. Um die Berechnung mit besagtem Zusammenhang durchführen zu können, wird die Funktion *VBWR* zur Bestimmung des molaren Volumens aufgerufen.

Tab 26: I/O-Charakteristik der Funktionen *DEPARTPCUB* und *DEPARTPCUBIC*

<i>DEPARTPCUB</i> :	Berechnung der Exzeßwärmekapazität mittels kubischer Gleichungen
Eingabe:	Temperatur, Druck, Molenbrüche, krit. Temperatur, krit. Druck, azentrischer Faktor, Wahl der kubischen Gleichung, allg. Gaskonstante, kubische Konstanten a und b
<i>DEPARTPCUBIC</i> :	Berechnung der Exzeßwärmekapazität mittels kubischer Gleichungen
Eingabe:	Temperatur, Druck, <u>Namen- bzw. Summenformeln der beteiligten Substanzen</u> , <u>Indexnummern der Substanzen</u> , Namen-/Summenformel-Schalter, Molenbrüche, Gaskonstante, <u>krit. Temperatur</u>, <u>krit. Druck</u>, <u>azentrischer Faktor</u>
Ausgabe:	je nach Größe von T und P Matrix, Vektor oder Skalar der Exzeßwärmekapazität

(**Fette Schrift** = optionale Argumente, unterstrichene Schrift = alternative Eingabemöglichkeit, siehe weitere Erläuterungen im Text und Kap. 1.2)

Da in den Funktionen *DEPARTPCUB/DEPARTPCUBIC* und *DEPARTCPBWR* eine Lösung der entsprechenden Zustandsgleichungen erforderlich ist, wird die Eingabestruktur dieser Funktio-

nen durch die aufgerufenen Funktionen zur Volumenberechnung bestimmt (s. Kap. 4.1).

Bei der Verwendung von *DEPARTCPCUB* muß der Benutzer die zur Berechnung notwendigen kritischen Daten Temperatur, Druck und den azentrischen Faktor selbst eingeben. Darüber hinaus können mit *DEPARTCPCUB* eigene kubische Parameter für die Berechnung verwendet werden. *DEPARTCPCUBIC* hingegen kann direkt auf Daten aus dem Datensatz *data* zugreifen. Der Einsatz eigener kubischer Parameter ist hier jedoch nicht möglich.

Hervorzuheben ist in diesem Zusammenhang, daß sowohl *DEPARTCPCUB* als auch *DEPARTCPCUBIC* die realen Exzeßwärmekapazitäten von Reinsubstanzen berechnen. Möchte der Benutzer die Exzeßwärmekapazität einer Mischung berechnen, so bilden *DEPARTCPCUB* und *DEPARTCPCUBIC* zunächst die entsprechenden Werte der Reinsubstanzen. Daraus wird dann über eine Gleichung [5] entsprechende Mischungsregel

$$\Gamma_m = \sum_{i=1}^N x_i \cdot \Gamma_i \quad [5]$$

die Mischungsgröße bestimmt, das heißt, die mit den Funktionen *DEPARTCPCUB* und *DEPARTCPCUBIC* berechneten Exzeßwärmekapazitäten einer Mischung sind die Werte einer idealen Mischung realer Gase. Soll aus den realen Reinstoffgrößen $\Delta C_{p,i}$ mit einer anderen Mischungsregel eine Mischungsgröße bestimmt werden, so muß der Benutzer dies selbst durchführen.

Bei der Verwendung von *DEPARTCPBWR* muß der Benutzer alle Parameter der BWR- bzw. BWRS-Gleichung selbst eingeben. Eine Berechnung dieser Parameter ist über die Funktion *ABBWR* möglich.

Tab 27: I/O-Charakteristik der Funktion *DEPARTCPBWR*

<i>DEPARTCPBWR</i> :	Berechnung der Exzeßwärmekapazität mittels BWR-Gleichung
Eingabe:	Temperatur, Druck, Molenbrüche, Reinstoffparameter, binäre Interaktionsparameter, allg. Gaskonstante
Ausgabe:	je nach Größe von T und P Matrix, Vektor oder Skalar der Exzeß-Wärmekapazität

(**Fette Schrift** = optionale Argumente)

Betrachtet man als Anwendungsbeispiel die Exzeßwärmekapazität einer Mischung von 46.3 % Methan mit 53.7 % Propan bei einer Temperatur von 361.05 K, so zeigen sich sehr unterschiedliche Ergebnisse.

Tab 28: Vergleich berechneter Exzeßwärmekapazitäten ΔC_p [J/(mol*K)] mit Literaturwerten [16]

Druck [bar]	$\Delta C_{p,lit.}$	$\Delta C_{p,BWR}$	$\Delta C_{p,RK}$	$\Delta C_{p,SRK}$	$\Delta C_{p,PR}$
13.8	3.92	3.93	3.22	3.72	3.73
40.8	15.57	15.63	76.51	89.01	88.17
68.03	34.39	34.62	29.7	37.13	38.13

Die bei einem Druck von 40.8 bar auftretenden starken Abweichungen der Ergebnisse kubischer Gleichungen (Subskript RK für Redlich-Kwong, SRK für Soave-Redlich-Kwong, PR für Peng-Robinson) resultieren aus der Nähe zum kritischen Punkt. Die Mischung hat einen kritischen Druck von 44.1 bar. In der Region dieses Druckes zeigt sich die größere Vorhersagegenauigkeit der BWR bzw. BWRS Gleichung.

5 Chemische Gleichgewichte

Die Berechnung chemischer Gleichgewichte stellt eine wichtige Anwendung bei der Auslegung chemischer Reaktoren und dem Auffinden der ökonomisch sinnvollsten Reaktionsbedingungen dar [40]. Daher ist die Betrachtung chemischer Gleichgewichtsreaktionen in Abhängigkeit der Reaktionsparameter sinnvoll [81].

Anhand von Literaturbeispielen werden Vor- und Nachteile verschiedener Methoden zur Ermittlung von chemischen Gleichgewichtszusammensetzungen diskutiert und beurteilt. Darüber hinaus werden Programme vorgestellt, die Möglichkeiten von MATLAB für die Bestimmung chemischer Gleichgewichte nutzen und teilweise erweitern.

In diesem Zusammenhang spielen neben numerischen Methoden nicht zuletzt auch stöchiometrische Überlegungen eine Rolle. Ihre Anwendung wird daher in einem gesonderten Kapitel betrachtet.

Die Berechnung von Gleichgewichtskonstanten wird in einem weiteren Kapitel beschrieben.

5.1 Gleichgewichtskonstanten

Chemische Gleichgewichte sind in ihrer Lage durch die Angabe von Gleichgewichtskonstanten eindeutig beschrieben. Diese lassen sich aus der Freien Gibbschen Reaktionsenthalpie berechnen [12, 52]:

$$\ln K = -\frac{\Delta_r G}{R \cdot T} \quad [35]$$

Hier bezeichnet das griechische Delta - entsprechend Gleichung [8] - eine auf eine Reaktion bezogene Größe:

$$\Delta_r C_p = \sum_{i=1}^N \nu_i \cdot C_{pi} \quad [8]$$

Für die Temperaturabhängigkeit der Gleichgewichtskonstanten ergibt sich folgender Zusammenhang [12]:

$$\ln K = -\frac{H_{298}^0}{R \cdot T} + \frac{S_{298}^0}{R} + \frac{1}{R} \cdot \int_{298}^T \frac{C_p^0}{T} dT - \frac{1}{R \cdot T} \cdot \int_{298}^T C_p^0 dT. \quad [36]$$

Gleichung [36] kommt in der Funktion *KT* zur Berechnung von temperaturabhängigen Gleichgewichtskonstanten zur Anwendung. Die Temperaturabhängigkeit der Wärmekapazität C_p wird in diesem Zusammenhang durch Gleichung [16] berücksichtigt:

$$C_p^0 = A_{C_p} + B_{C_p} \cdot T + C_{C_p} \cdot T^2 + D_{C_p} \cdot T^3. \quad [16]$$

Daher fließen die in Gleichung [16] auftretenden Parameter auch als Argumente in die Funktion *KT* mit ein. Weitere Argumente der Funktion *KT* sind die Standardbildungs-Enthalpie und die Freie Gibbssche Standardbildungs-Enthalpie.

Tab 29: I/O-Charakteristik der Funktion *KT*

<i>KT</i> :	Berechnung der temperaturabhängigen Gleichgewichtskonstanten chemischer Reaktionen
Eingabe:	Temperatur, Vektor der stöchiometrischen Koeffizienten, Vektor der auf <i>Data</i> verweisenden Indizes, <u>Standardbildungsenthalpie</u> , <u>freie Gibbssche Standardbildungsenthalpie</u> , <u>Parameter zur Berechnung der idealen Gaswärmekapazität</u>
Ausgabe:	Je nach Größe von T und N Matrix, Vektor oder Skalar Gleichgewichtskonstanten der betrachteten chemischen Reaktionen

(**Fette Schrift** = optionale Argumente, unterstrichene Schrift = alternative Eingabemöglichkeit, siehe weitere Erläuterungen im Text und Kap. 1.2)

Die Eingabe der Argumente für *KT* kann auf verschiedene Weise erfolgen. Sie können vollständig aus dem Datensatz *data* entnom-

men werden. Dann sind keine weiteren manuellen Eingaben notwendig. Die zur Berechnung erforderlichen Daten kann der Benutzer auch manuell eingeben. Dabei muß die Reihenfolge der Angaben beachtet werden, da sonst die Funktion die Daten nicht den richtigen Substanzen zuordnen kann. Eine Kombination manuell eingegebener Daten mit der Benutzung des Datensatzes *data* ist auch möglich.

Weitere Argumente, die der Funktion *KT* übergeben werden müssen, sind die Temperatur, die Reaktionsmatrix und ein Vektor aller auf *data* verweisenden Indexnummern. Wie auch bei der Implementierung der Funktion *HRx* (s. Kap. 3.1) ist bei *KT* die Eingabe der fraglichen Reaktionen nicht als Zeichenfolge, sondern in Form einer Matrix der stöchiometrischen Koeffizienten vorgesehen. Über den Vektor der Indexnummern erfolgt dann die eindeutige Identifizierung der Substanzen. Sollen bei der Berechnung manuell eingegebene Daten verarbeitet werden, so muß der Benutzer die entsprechenden Substanzen in dem Vektor der Indexnummern mit einer Null kennzeichnen. Im Vorfeld kann wieder die Funktion *RX2N* zur Anwendung kommen (s. Kap.3.1).

Im folgenden Beispiel wird das Wassergasgleichgewicht bei 1000 K und 1 bar betrachtet.

```
» load data
» string='H2 + CO2 -> H2O + CO';
» [N,val]=RX2N(string);
```

Das nach dieser Eingabe erscheinende Textmenü zur Entscheidung zwischen zwei Entsprechungen in *data* für H₂ ist hier nicht gezeigt. Zur Demonstration sollen die Daten für CO₂ manuell eingegeben werden. Dazu wird die entsprechende Indexnummer in *N* gleich Null gesetzt.

```
» N(2)=0;
```

CO₂ ist durch die 2. Indexnummer in der Variablen `val` gekennzeichnet. Durch die folgenden Befehle werden die notwendigen Daten neu definierten Variablen zugewiesen:

```
» dH=data(val(2)).DELHF;dG=data(val(2)).DELGF;  
» a=data(val(2)).CPVAP_A;b=data(val(2)).CPVAP_B;  
» c=data(val(2)).CPVAP_C;d=data(val(2)).CPVAP_D;
```

Beim Aufruf von *KT* werden die Variablen `dH`, `dG` und `a` bis `d` dann direkt eingegeben.

```
» KT(1000,N,val,dH,dG,a,b,c,d)
```

```
ans =
```

```
0.7875
```

Dieser Wert liegt mit einem Fehler von 11 Prozent in Übereinstimmung mit dem Literaturwert von 0.7 [12].

5.2 Stöchiometrische Berechnungen

Für die Berechnung chemischer Gleichgewichte sind grundlegende stöchiometrische Vereinbarungen getroffen worden. Ausgehend von einem geschlossenen System werden der Aufbau und die Handhabung von Element-Spezies-Matrizen, von stöchiometrischen Matrizen und das Auffinden bzw. der Nutzen von Schlüsselkomponenten und Schlüsselreaktionen erläutert.

5.2.1 Schlüsselkomponenten

Betrachtet wird ein geschlossenes, chemisches System, in dem N Komponenten $S_1, \dots, S_j, \dots, S_N$ miteinander reagieren. Während dieser Reaktionen bleibt jedoch die Zahl der Atome jedes einzelnen Elementes konstant. Quantitativ läßt sich dieser Zusammenhang mittels einer linearen Gleichung ausdrücken [2, 4, 43]:

$$\sum_{i=1}^N a_{hi} \cdot n_i = b_h \quad h = 1, \dots, L \quad [37]$$

bzw. in Matrixschreibweise darstellen

$$\underline{A} \cdot \underline{n} = \underline{b}. \quad [38]$$

Hier ist L die Anzahl der Elemente in den N Komponenten S_i und somit h ein Zähler der Elemente. Weiterhin ist a_{hi} der Koeffizient des Elementes h in der Summenformel der Komponente S_i , \underline{A} die Matrix der entsprechenden Koeffizienten (s. Kap. 5.2.2) und n_i stellt die Stoffmenge der Spezies S_i in Mol dar. Für den Begriff der Komponente kann auch das Synonym Spezies verwendet werden. Somit ergibt sich b_h zu der konstanten Molzahl des Elementes h.

Zur Verdeutlichung sei das Beispiel eines Systems von NH_3 , O_2 , H_2O , NO und NO_2 angeführt [4]. Die Zusammensetzung wird durch den Spaltenvektor $\underline{b} = [4;12;14]$ bestimmt. Damit ergibt sich Gleichung [37] zu

$$\begin{aligned} 1 n_1 + 0 n_2 + 0 n_3 + 1 n_4 + 1 n_5 &= b_N = 4 \\ 3 n_1 + 0 n_2 + 2 n_3 + 0 n_4 + 0 n_5 &= b_H = 12 \\ 0 n_1 + 2 n_2 + 1 n_3 + 1 n_4 + 2 n_5 &= b_O = 14. \end{aligned}$$

Daraus resultiert Gleichung [38] in vektorisierter Form:

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 3 & 0 & 2 & 0 & 0 \\ 0 & 2 & 1 & 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} n_{\text{NH}_3} \\ n_{\text{O}_2} \\ n_{\text{H}_2\text{O}} \\ n_{\text{NO}} \\ n_{\text{NO}_2} \end{pmatrix} = \begin{pmatrix} 4 \\ 12 \\ 14 \end{pmatrix}.$$

Betrachtet man zwei unterschiedliche Zustände eines reagierenden Systems, so stellen Δn_i bzw. $\Delta \underline{n}$ die Veränderung der Stoffmenge der Komponente i bzw. aller Komponenten dar. Da die Gesamt-

menge der beteiligten Elemente jedoch konstant bleibt, resultieren aus den Gleichungen [37] und [38] folgende Zusammenhänge:

$$\sum_{i=1}^N a_{hi} \cdot \Delta n_i = 0 \quad h = 1, \dots, L \quad [39]$$

$$\underline{A} \cdot \underline{\Delta n} = 0. \quad [40]$$

Die Matrix \underline{A} hat die Dimension $L \times N$ und ihr Rang, R_A , ist meistens gleich L . Da in den meisten Fällen $N > L$ gilt, ist das Gleichungssystem unterbestimmt. Die Lösung eines unterbestimmten Gleichungssystems ist durch die Ermittlung von R_A gebundenen Unbekannten in Abhängigkeit von $N - R_A$ freien Unbekannten gegeben [44]. Die freien Unbekannten sind hier die sogenannten Schlüsselkomponenten. Diese sind stets so zu wählen, daß in den Nichtschlüsselkomponenten mindestens R_A Elemente enthalten sind. Dies entspricht mathematisch der Forderung, daß die zu den R_A Nichtschlüsselkomponenten gehörenden Spaltenvektoren der Element-Spezies-Matrix \underline{A} linear unabhängig sind [2].

Zur Ermittlung von Sätzen von Schlüsselkomponenten kann das Programm *KEYESM* verwandt werden.

Tab 30: I/O-Charakteristik von *KEYESM*

<i>KEYESM</i> :	Suche nach Kombinationen von Schlüsselkomponenten
Eingabe:	Element-Spezies-Matrix, \underline{A}
Ausgabe:	Position der Spezies in der Element-Spezies-Matrix ausgegeben in Form eines <i>cell array</i>

KEYESM ermittelt aus der sogenannten Element-Spezies-Matrix \underline{A} nach vorgenannter Definition mögliche Kombinationen von Schlüsselkomponenten. Diese werden in Form von Positionsangaben der Spezies in der Matrix \underline{A} ausgegeben. Wird *KEYESM* auf das Beispiel von NH_3 , O_2 , H_2O , NO und NO_2 angewendet, so ergeben sich folgende Kombinationen von Schlüsselkomponenten:

Tab 31: Ergebnis nach *KEYESM* Aufruf, (SK \equiv Schlüsselkomponenten, NSK \equiv Nichtschlüsselkomponenten)

Lösung	1	2	3	4	5	6	7	8	9
Ausgaben	(4, 5)	(3, 5)	(3, 4)	(2, 5)	(2, 4)	(2, 3)	(1, 5)	(1, 4)	(1, 2)
SK	NO, NO ₂	H ₂ O, NO ₂	H ₂ O, NO	O ₂ , NO ₂	O ₂ , NO	O ₂ , H ₂ O	NH ₃ , NO ₂	NH ₃ , NO	NH ₃ , O ₂
NSK	NH ₃ , O ₂ , H ₂ O	NH ₃ , O ₂ , NO	NH ₃ , O ₂ , NO ₂	NH ₃ , H ₂ O, NO	NH ₃ , H ₂ O, NO ₂	NH ₃ , NO, NO ₂	O ₂ , H ₂ O, NO	O ₂ , H ₂ O, NO ₂	H ₂ O, NO, NO ₂

5.2.2 Element-Spezies-Matrix

Zur Erstellung der Element-Spezies-Matrix A kann das Programm *ELSPEMA* verwendet werden. Als Eingabeargumente übergibt der Benutzer einen String aller auftretenden Elemente und einen String aller auftretenden Spezies. Dabei sind die Elemente bzw. Spezies durch ein Leerzeichen voneinander zu trennen.

Tab 32: I/O-Charakteristik von *ELSPEMA*

<i>ELSPEMA</i> :	Erstellen der Element-Spezies-Matrix
Eingabe:	Liste aller Elemente in String Format, Liste aller Spezies in String Format
Ausgabe:	Element-Spezies-Matrix

Soll die mit *ELSPEMA* erstellte Element-Spezies-Matrix für stöchiometrische Betrachtungen (s. Kap. 5.2.1 und Kap. 5.2.3) verwendet werden, so müssen die Schlüsselkomponenten beim Funktionsaufruf in der Speziesliste als letzte geschrieben werden.

Der *ELSPEMA* zugrunde liegende Algorithmus erkennt selbständig die Schreibweise der eingegebenen Summenformeln. Sollten geladene Spezies betrachtet werden, so gibt *ELSPEMA* die Ladungen als zusätzliche Zeile am Ende der Element-Spezies-Matrix aus.

» A=ELSPEMA('H N O Mn', 'NO H2O NO3- H+ MnO4- Mn++')

A =

0	2	0	1	0	0
1	0	1	0	0	0
1	1	3	0	4	0
0	0	0	0	1	1
0	0	-1	1	-1	2

Die letzte Zeile der Matrix A bezieht sich hier ausschließlich auf die Ladungen der betrachteten Spezies. Dabei wird allein für negative Ladungen ein Vorzeichen angegeben. Positive Ladungen werden als positive ganze Zahlen ausgegeben.

5.2.3 Schlüsselreaktionen

Nachdem in Kapitel 5.2.1 zur Ermittlung von Schlüsselkomponenten die Funktion *KEYESM* erläutert wurde, stellt sich jetzt die Frage mit welchen stöchiometrischen Beziehungen das reagierende System beschrieben werden kann. Betrachtet wird wieder ein geschlossenes, reagierendes chemisches System. Chemische Reaktionen können mit Hilfe der Element-Spezies-Matrix A dargestellt werden.

$$\sum_{i=1}^N a_{hi} \cdot v_{ih} = 0. \quad h = 1, \dots, L \quad [41]$$

Hier sind a_{hi} die Elemente der Matrix A und v_{ih} die stöchiometrischen Koeffizienten der Substanzen in den ablaufenden Reaktionen. Bei Gleichung [41] handelt es sich um ein lineares homogenes Gleichungssystem. Da das reagierende chemische System zumeist mehr Spezies als Elemente enthält (s. das Bsp. in Kap.5.2.1), ist das Gleichungssystem [41] unterbestimmt und hat unendlich viele Lösungen. Es können Lösungen erhalten werden, wenn man für die stöchiometrischen Koeffizienten v_{ih} der Schlüsselkomponenten willkürlich Werte annimmt und das resultierende Gleichungssys-

tem löst. Aus $N - R_A$ solcher Lösungen können durch Linearkombinationen beliebig viele andere Lösungen erhalten werden. Zur Verdeutlichung sei das Beispiel aus Kap. 5.2.1, ein System aus NH_3 , O_2 , H_2O , NO und NO_2 , angeführt [4]. Hier werden für die v_{ih} der Schlüsselkomponenten nach einander die Werte (1,0) und (0,1) angenommen und die verbleibenden Gleichungen gelöst. Damit ergeben sich folgende Vektoren \underline{v}_i der stöchiometrischen Koeffizienten:

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 3 & 0 & 2 & 0 & 0 \\ 0 & 2 & 1 & 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} -1 \\ -1.25 \\ 1.5 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 3 & 0 & 2 & 0 & 0 \\ 0 & 2 & 1 & 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} -1 \\ -1.75 \\ 1.5 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

Die Zusammenstellung der $N - R_A$ von einander linear unabhängigen Lösungsvektoren \underline{v}_i ergibt einen fundamentalen Lösungssatz. Dies ist hier die Matrix \underline{N} .

$$\underline{v}_1 = \begin{pmatrix} -1 \\ -1.75 \\ 1.5 \\ 0 \\ 1 \end{pmatrix}; \underline{v}_2 = \begin{pmatrix} -1 \\ -1.25 \\ 1.5 \\ 1 \\ 0 \end{pmatrix}; \underline{N} = \begin{pmatrix} -1 & -1 \\ -1.75 & -1.25 \\ 1.5 & 1.5 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}$$

Die Matrix \underline{N} ist mit Hilfe des Programmes *STOIMA* zugänglich.

Tab 33: I/O-Charakteristik der Funktion *STOIMA*

<i>STOIMA</i> :	Erstellen der stöchiometrischen Matrix, \underline{N}
Eingabe:	Element-Spezies-Matrix, \underline{A}
Ausgabe:	Stöchiometrische Matrix, \underline{N}

STOIMA bildet unter Anwendung eines Algorithmus nach Smith und Missen [4] aus der Element-Spezies-Matrix einen vollständigen Satz von Schlüsselreaktionen. Intern erzeugt die Funktion eine Matrix die nach dem Schema

$$\underline{N} = \begin{pmatrix} -\underline{Z} \\ \underline{I}_R \end{pmatrix} \quad [42]$$

aufgebaut ist. In Gleichung [42] sind $-\underline{Z}$ die stöchiometrischen Koeffizienten der Nichtschlüsselkomponenten in den Schlüsselreaktionen und \underline{I}_R eine $R_A \times R_A$ Einheitsmatrix.

Bei der Anwendung der Funktion *STOIMA* ist es notwendig, daß die Element-Spezies-Matrix \underline{A} entsprechend der Konvention aus Kapitel 5.2.2 aufgestellt worden ist. Das heißt, daß die Spalten der Matrix \underline{A} , die die Schlüsselkomponenten repräsentieren, an letzter Stelle in \underline{A} stehen.

Der in *STOIMA* verwendete Algorithmus kann ausschließlich Element-Spezies-Matrizen verarbeiten, die sich auf ungeladene Komponenten beziehen. Sind bei der Erstellung von \underline{A} durch die Funktion *ELSPEMA* geladene Spezies mit berücksichtigt worden, so enthält \underline{A} eine Zeile, die sich ausschließlich auf die Ladung der Komponenten bezieht (s. Kap. 5.2.2). Eine solche Element-Spezies-Matrix kann erst mit *STOIMA* verarbeitet werden, wenn die sich auf Ionenladungen beziehende Zeile entfernt wird.

Zur Veranschaulichung der stöchiometrischen Matrix \underline{N} und der Element-Spezies-Matrix dienen die Funktionen *ESM2EQU* und *N2EQU*. Sie erstellen Reaktionsgleichungen der Schlüsselreaktionen in Stringformat. Dabei geht *ESM2EQU* von der Element-

Spezies-Matrix \underline{A} und der Liste aller beteiligten Spezies aus. Daraus wird dann der String der Schlüsselreaktionen erstellt. *N2EQU* hingegen geht von der stöchiometrischen Matrix \underline{N} und der Liste aller beteiligten Spezies aus.

Tab 34: I/O-Charakteristiken von *ESM2EQU* und *N2EQU*

<i>ESM2EQU:</i>	Ausgabe von Schlüsselreaktionen in Stringformat
Eingabe:	Element-Spezies-Matrix, \underline{A} und Speziesliste in Stringformat, Elemente der Liste je durch Freizeichen getrennt
Ausgabe:	String der Schlüsselreaktionen
<i>N2EQU:</i>	Ausgabe von Schlüsselreaktionen in Stringformat
Eingabe:	Stöchiometrische Matrix, \underline{N} und Speziesliste in Stringformat, Elemente der Liste je durch Freizeichen getrennt
Ausgabe:	String der Schlüsselreaktionen

Oftmals ist jedoch ein umgekehrtes Vorgehen erforderlich. Aus einem Satz von Reaktionsgleichungen, der im allgemeinen sowohl linear abhängige als auch unabhängige Reaktionen enthält, wird ein Satz von Schlüsselreaktionen gesucht. Die Funktion *KEYREAC* sucht in den, durch eine stöchiometrische Matrix \underline{N} definierten, Reaktionen nach einem Satz von Schlüsselreaktionen. Voraussetzung für den Erfolg der Suche ist, daß in \underline{N} auch ein vollständiger Satz von Schlüsselreaktionen enthalten ist.

Tab 35: I/O-Charakteristik der Funktion *KEYREAC*

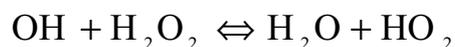
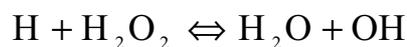
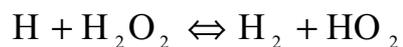
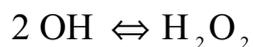
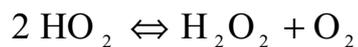
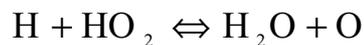
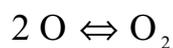
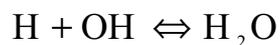
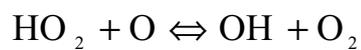
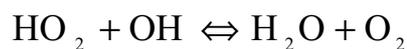
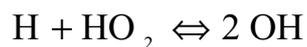
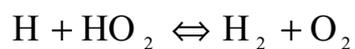
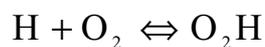
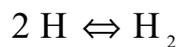
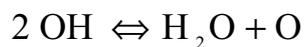
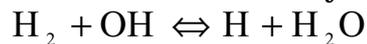
<i>KEYREAC:</i>	Bilden eines Satzes von Schlüsselreaktionen
Eingabe:	Stöchiometrische Matrix, \underline{N} , der vorgegebenen Reaktionen
Ausgabe:	Stöchiometrische Matrix, \underline{N} , der Schlüsselreaktionen

Ob in den eingegebenen Reaktionen ein vollständiger Satz Schlüsselreaktionen enthalten ist, kann der Benutzer an folgenden Punkten erkennen:

- Die von *KEYREAC* ausgegebene Matrix muß $N - R_A$ Zeilen enthalten.

- Die $N - R_A$ Schlüsselkomponenten kommen in jeder Zeile nur einmal vor.
- Die von *KEYREAC* ausgegebene Matrix muß eine zeilenreduzierte gestaffelte Form aufweisen.

Dies wird hier an einem Beispiel aus [42] verdeutlicht. Betrachtet wird das Reaktionssystem,



welches durch die folgende Matrix der stöchiometrischen Koeffizienten beschrieben wird:

RN =

1 1 -1 0 0 0 -1 0

-1	0	0	0	0	1	1	-1
1	0	-1	0	0	-1	1	0
0	1	0	0	0	1	-2	0
-2	0	1	0	0	0	0	0
-1	0	0	1	0	0	0	-1
-1	0	1	-1	0	0	0	1
-1	0	0	-1	0	0	2	0
0	1	0	-1	0	0	-1	1
0	0	0	-1	0	-1	1	1
-1	1	0	0	0	0	-1	0
0	0	0	0	0	-2	0	1
-1	1	0	-1	0	1	0	0
0	0	0	-2	1	0	0	1
0	0	0	0	1	0	-2	0
-1	0	1	1	-1	0	0	0
-1	1	0	0	-1	0	1	0
0	0	0	1	-1	-1	1	0
0	1	0	1	-1	0	-1	0

Durch Anwendung der Funktion *KEYREAC* auf RN resultieren folgende stöchiometrischen Gleichungen.

» $N = \text{KEYREAC}(\text{RN})$;

N =

1	0	0	0	0	0	-1	0.5
0	1	0	0	0	0	-2	0.5
0	0	1	0	0	0	-2	1
0	0	0	1	0	0	-1	-0.5
0	0	0	0	1	0	-2	0
0	0	0	0	0	1	0	-0.5

Bei Anwendung der Funktion *N2EQU* auf N wird deutlicher, daß die von *KEYREAC* ausgegebenen Gleichungen die gestellten Anforderungen erfüllen.

```
» N2EQU(N, 'H H2O H2 HO2 H2O2 O OH O2')
```

```
ans =
```

```
OH = H + 0.5 O2
```

```
2 OH = H2O + 0.5 O2
```

```
2 OH = H2 + O2
```

```
OH + 0.5 O2 = HO2
```

```
2 OH = H2O2
```

```
0.5 O2 = O
```

Die in Kapitel 5.2 beschriebenen Funktionen behandeln stöchiometrische Standardfragestellungen. Mit ihrem Einsatz kann der Benutzer die zur Beschreibung eines reagierenden Gemisches aufgestellten Reaktionen auf stöchiometrische Vollständigkeit überprüfen. Die Funktionen ermöglichen aber auch die stöchiometrische Beschreibung eines Systems reagierender Substanzen. Der realisierte Ansatz verbindet Flexibilität durch Einzelanwendung der Funktionen mit der Möglichkeit einer umfassenden stöchiometrischen Beschreibung des gesamten Systems durch Anwendung der ineinandergreifenden Funktionen.

5.3 Berechnung von Simultangleichgewichten

Bei technischen Prozessen ist oftmals nicht nur ein einzelnes Produkt von Interesse, sondern die Verfolgung von Parallel- und Folgereaktionen erforderlich. So treten - zum Beispiel bei der Dehydrierung von gesättigten Kohlenwasserstoffen - nach einem komplexen Reaktionsschema aus Folge- und Parallelreaktionen nebeneinander Olefine, Diolefine, Polyolefinen, Acetylenkohlenwasserstoffen und Aromaten auf [2]. Für die Berechnung der Gleichgewichtslage eines solchen Reaktionssystems müssen mehrere chemische Gleichgewichte gleichzeitig berücksichtigt werden. Zur Lösung solcher Simultangleichgewichte sind mehrere Methoden in der Literatur diskutiert worden. In dem vorliegenden Kapitel wird auf Methoden zur Lösung dieser nicht-linearen Gleichungen eingegangen.

Auf einen weiteren wichtigen Aspekt, der Verfolgung des Reaktionsfortschrittes bei simultanen Reaktionen, wird in einem gesonderten Kapitel eingegangen.

5.3.1 Reaktionsfortschritt

Der Fortschritt einer chemischen Reaktion kann über den Umsatzgrad X verfolgt werden. Abhängige Variablen können so explizit durch X ausgedrückt werden. Dieses Vorgehen war bei analytischen Berechnungen notwendig. Der Reaktionsfortschritt simultan ablaufender Reaktionen kann jedoch nicht über den Umsatz verfolgt werden. Mit der Einführung einer Reaktionslaufzahl ξ kann auch der Fortgang simultaner Reaktionen berücksichtigt werden.

Der Fortschritt von Reaktionen kann jedoch auch gänzlich ohne die Anwendung einer 'zusätzlichen' Größe betrachtet werden. Dies leistet das Konzept der Schlüsselkomponenten (Index k) [2, 42]. Um eine leichtere Verständlichkeit zu gewährleisten, sollen hier zunächst einzeln ablaufende Reaktionen erläutert werden. Im Anschluß daran wird das Konzept dann auf simultane Gleichgewichte erweitert.

Die Molenbrüche aller an der Reaktion beteiligten Spezies (Index i) in Abhängigkeit der x_k ergeben sich aus der stöchiometrischen Gleichung $\nu_i A_i = 0$. Diese besagt:

$$\frac{1}{\nu_i} (n_i - n_i^0) = \frac{1}{\nu_k} (n_k - n_k^0). \quad [43]$$

Die Arbeitsgleichung ergibt sich durch Auflösen nach $x_i = n_i/n$:

$$x_i = \left[\frac{\nu_i}{\nu_k} (n \cdot x_k - n^0 \cdot x_k^0) + n^0 \cdot x_i^0 \right] \cdot \frac{1}{n}. \quad [44]$$

Dabei sind die Eingangsgrößen mit einer hochgestellten 0 gekennzeichneten. Aus Gleichung [43] kann durch Summieren über alle i ein Ausdruck für n erhalten werden:

$$n = n^0 \cdot \frac{\left(1 - \frac{v}{v_k} x_k^0 \right)}{\left(1 - \frac{v}{v_k} x_k \right)}, \quad [45]$$

wobei $v = v_i$ ist. Mit dem vorgestellten Konzept ist es problemlos möglich, auch simultan ablaufende Reaktionen zu betrachten. Dazu kann die mit Hilfe der Programme *ELSPEMA* und *STOIMA* (s. Kapitel 5.2.2 und 5.2.3) erstellte stöchiometrische Matrix \underline{N} verwendet werden. Durch Aufteilen in eine Schlüsselkomponenten- und eine Nichtschlüsselkomponenten-Matrix ergeben sich die Arbeitsmatrizen. Ausgehend von dem Format der mit obigen Programmen erhaltenen stöchiometrischen Matrix \underline{N} finden sich diese wie folgt.

$$\underline{N}' = [\underline{NK}; \underline{NNK}] = \begin{bmatrix} \underline{NK} \\ \underline{NNK} \end{bmatrix} \quad [46]$$

In Gleichung [46] wird die MATLAB-Schreibweise verwendet. Dabei bezeichnet \underline{N}' die transponierte Matrix von \underline{N} . In dem darauffolgenden ersten Klammersausdruck gibt das Semikolon einen Zeilenumbruch an. Dies verdeutlicht der zweite Klammersausdruck. Die Indizes in Gleichung [46] bringen die Einteilung der Matrix \underline{N} zum Ausdruck. Hier steht \underline{NK} für die stöchiometrische Matrix der Schlüsselkomponenten und \underline{NNK} für die stöchiometrische Matrix der Nichtschlüsselkomponenten. Unter Verwendung dieser partitionierten Matrizen läßt sich eine Gleichung [44] entsprechende Beziehung für simultane Reaktionen erhalten:

$$\underline{x}_i = \left[\underline{N} \underline{N} \underline{K} \cdot \underline{N} \underline{K}^{-1} (n \cdot \underline{x}_k - n^0 \cdot \underline{x}_k^0) + n^0 \cdot \underline{x}_i^0 \right] \cdot \frac{1}{n}$$

[47]

Hier sind \underline{x} Vektoren der Molenbrüche. Werden die stöchiometrischen Gleichungen so erstellt, daß $\underline{N} \underline{K}$ die Einheitsmatrix ist, so kann sie auch weggelassen werden. Mit dieser Vereinfachung ergibt sich ein Ausdruck für n zu:

$$n = n^0 \cdot \frac{\left(1 - \underline{s}^T \underline{x}_k^0 \right)}{\left(1 - \underline{s}^T \underline{x}_k \right)}, \quad [48]$$

wobei $\underline{s}_j = \underline{N} \underline{N} \underline{K}_{ij}$ ist. Soll hingegen eine stöchiometrische Matrix \underline{N} verwendet werden, in der die Matrix der Schlüsselkomponenten $\underline{N} \underline{K}$ nicht die Einheitsmatrix ist, so muß für \underline{s}_j ein anderer Ausdruck eingesetzt werden.

$$\underline{s}_j = \sum \left(\frac{\underline{N} \underline{N} \underline{K}}{\underline{I}} \frac{\underline{N} \underline{K}^{-1}}{ij} \right) \quad [49]$$

5.3.2 Lösen von nicht linearen algebraischen Gleichungen

Chemische Gleichgewichte werden durch das Massenwirkungsgesetz beschrieben. Dabei handelt es sich um algebraische Gleichungen. Gleichung [50] gibt zum Beispiel die Gleichgewichtsbeziehung einer Reaktion idealer Gase wieder:

$$K_{pj} = \prod_{i=1}^N p_i^{\nu_{ij}} \quad j = 1, \dots, R_A. \quad [50]$$

Zur Lösung algebraischer Gleichungen bietet MATLAB verschiedene Optimierverfahren an. Für die Lösung des chemischen Gleichgewichtes in Form algebraischer Gleichungen reichen die durch Gleichung [50] gegebenen Zusammenhänge allein nicht aus. Zur Lösung eines Systems mit n Unbekannten werden mindestens n

Gleichungen benötigt. Um insgesamt n Gleichungen zusammenzustellen, werden die Elementarbilanzen

$$\sum_{i=1}^N A_{hi} \cdot n_i = b_h \quad h = 1, \dots, L \quad [37]$$

und eine Schließbedingung

$$\sum_{i=1}^N n_i = n \quad [51]$$

herangezogen. Wendet man hier das Konzept der Schlüsselkomponenten (s. Kap. 5.3.1) an, so können zwanglos allein die Schlüsselreaktionen betrachtet werden. Damit muß der Gleichungslöser ($N - R_A$) Gleichungen mit ($N - R_A$) Unbekannten lösen.

Die von MATLAB zur Verfügung gestellten Funktionen unterscheiden sich durch die verwendeten Optimierungsalgorithmen. Die im Rahmen dieser Arbeit verwendete *Optimization Toolbox* beinhaltet neben Funktionen, die explizit zur Lösung nicht-linearer Gleichungen vorgesehen sind, auch Programme zur Parameteroptimierung und zur Datenanpassung [45]. Der Begriff der Parameteroptimierung bedeutet das Auffinden von optimalen Werten frei wählbarer Variablen für einen angestrebten Zustand einer betrachteten Funktion [82]. Auch wenn andere Funktionen der *Optimization Toolbox* zur Lösung nicht-linearer Gleichungen herangezogen werden können, so soll hier vor allem die Funktion *FSOLVE* betrachtet werden, da sie explizit als Lösungsverfahren für diesen Fall ausgezeichnet ist.

Um ein chemisches Gleichgewicht mit dieser Funktion berechnen zu können, muß ein separates M-File erstellt werden. In diesem File müssen die angesprochenen Ausdrücke abgelegt werden. Dieses M-File wird dann von dem Gleichungslöser *FSOLVE* aufgerufen. Das File stellt somit den funktionalen Zusammenhang dar, für den *FSOLVE* einen Satz freier Variablen finden muß, mit denen dieser

Zusammenhang minimal wird. Dabei kommt entweder ein Algorithmus nach Gauss-Newton oder ein Algorithmus nach Levenberg-Marquardt zur Anwendung [45]. Um dem Benutzer den Umgang mit diesem Werkzeug bei der Berechnung chemischer Simultangleichgewichte zu erleichtern, wurde das File *EENL* (Equilibrium Estimate Non Linear equations) programmiert.

Tab 36: I/O-Charakteristik der Funktion *EENL*

<i>EENL</i> :	Berechnung chemischer Simultangleichgewichte
Eingabe:	Vektor der Startwerte der Molenbrüche der Schlüsselkomponenten, Stöchiometrische Matrix der Reaktionen, Startstoffmenge, Vektor der Startmolenbrüche der Schlüsselkomponenten, Vektor der Startmolenbrüche der Nicht-Schlüsselkomponenten, Vektor der Gleichgewichtskonstanten, Methodenschalter für Aktivitätskoeffizientenberechnung, Reaktionstemperatur, zwei Argumente zur γ Berechnung
Ausgabe:	Auf die Schlüsselkomponenten bezogener Funktionsvektor entsprechend Gleichung [52]

(**Fette Schrift** = optionale Argumente)

Das File *EENL* gibt dem aufrufenden Gleichungslöser *FSOLVE* einen Antwortvektor zurück. Dabei werden unter Berücksichtigung des Konzeptes der Schlüsselkomponenten allein die Schlüsselreaktionen und die Schlüsselkomponenten betrachtet. Eingeben muß der Benutzer die Matrix der stöchiometrischen Koeffizienten der Reaktionen, die Gleichgewichtskonstanten, die Zusammensetzung des Ausgangsgemisches und die Stoffmenge bei Reaktionsbeginn.

Weiterhin bietet *EENL* die Möglichkeit, in die Berechnung des Gleichgewichtes Aktivitätskoeffizienten mit einzubeziehen. Diese können über mehrere Methoden berechnet werden: Die Methode nach Wilson, die UNIQUAC-Methode und die NRTL-Methode (s.

Kap. 6). Dabei werden die Aktivitätskoeffizienten γ_i mit in Gleichung [52] einbezogen [6]:

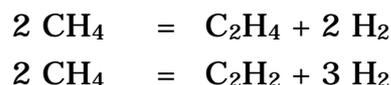
$$F_j(x_i, K_E) = K_{Ej} - \prod_{i=1}^N (x_i \cdot \gamma_i)^{v_{ij}} = 0. \quad [52]$$

Zur Berechnung der γ_i sind weitere Argumente notwendig. Diese werden am Schluß der Argumenteliste des *FSOLVE* Aufrufes übergeben. Mit einem Schalter kann der Benutzer entscheiden, welche der Gleichungen zur Anwendung kommen soll. Es besteht auch die Möglichkeit, daß ein konstanter Faktor an der Stelle der γ_i in die Berechnung mit einfließt. Dieser kann durch das erste optionale Argument für die γ_i Berechnung beim *EENL* Aufruf durch *FSOLVE* eingegeben werden.

Es sind mit *EENL* verschiedene Berechnungen möglich. Bei der Verarbeitung der Argumente gelten folgende Belegungen:

- $sw_i = 0$ bedingt die Vernachlässigung von γ_i
- $sw_i = 1$ bedingt die γ_i Berechnung nach Wilson
- $sw_i = 2$ bedingt die γ_i Berechnung nach UNIQUAC
- $sw_i = 3$ bedingt die γ_i Berechnung nach NRTL
- $sw_i = -1$ bedingt die Verwendung des eingegebenen zusätzlichen Parameters als konstanten Faktor.

EENL muß in einer bestimmten Weise mit *FSOLVE* aufgerufen werden. Im folgenden Beispiel wird die thermische Dehydrierung von Methan betrachtet. Dabei wird bei 1300 K und einem Gesamtdruck von 1 bar von reinem Methan ausgegangen [47].



In dem betrachteten Beispiel wird die γ_i Berechnung nicht berücksichtigt. Zunächst werden den notwendigen Argumenten Werte zugewiesen. Die Variablennamen sind wie bereits in früheren

Beispielen auch, willkürlich gewählt und nicht zwingend vorgeschrieben für die Benutzung von *EENL*

```
» NNS=[-2 2 1 0;-2 3 0 1]';
```

NNS gibt die Matrix der stöchiometrischen Koeffizienten der betrachteten chemischen Reaktionen wieder.

```
» x0S=[0;0];x0N_S=[1;0];
```

x0S und *x0N_S* stehen für die Molenbrüche der Schlüsselkomponenten (*x0S*) und der Nicht-Schlüsselkomponenten (*x0N_S*) des Ausgangsgemisches.

```
» n0=1;
```

Mit *n0* wird hier die Stoffmenge des Eduktgemisches angegeben.

```
» K=[0.03;0.01];
```

K beinhaltet hier die Gleichgewichtskonstanten der betrachteten Reaktionen. *swi* ist ein Schalterargument, welches *EENL* signalisiert, daß die Berechnung ohne Verwendung von Aktivitätskoeffizienten erfolgen soll.

```
» swi=0;
```

Dann werden Start-Schätzwerte für die Berechnung bestimmt.

```
» xSchaetz=[0.1;0.1];
```

Abschließend wird der Gleichungslöser aufgerufen.

```
» fsolve('EENL',xSchaetz,[],[],NNS,n0,x0S,x0N_S,K,swi)
```

```
ans =
```

0.0695

0.0677

Als Ergebnis liefert *FSOLVE* mit *EENL* einen Vektor der Molenbrüche der Schlüsselkomponenten, die die chemischen Gleichgewichte erfüllen. Schlüsselkomponenten sind hier Ethen C_2H_4 und Ethin C_2H_2 . Um den Umgang mit der Funktion *EENL* zu erleichtern, wurde das M-File *EENLMASK* erstellt. Es demonstriert an einem Beispiel die Handhabung von *EENL*. Da die Anwendung der Funktion *EENL* jedoch bereits ausführlich in diesem Kapitel behandelt wird, soll das File *EENLMASK* nicht tiefergehend erläutert werden.

Prinzipielles Problem bei der Verwendung von Verfahren zur Lösung nicht linearer algebraischer Gleichungen ist, daß die Startwerte nah bei den eigentlichen Zielwerten liegen müssen. Sogenannte "gute" Startwerte sind notwendig, da sonst Konvergenzprobleme auftreten können, das heißt, es wird entweder ein lokales Minimum in Form eines falschen oder nur vordergründig richtigen Ergebnisses gefunden, oder es kommt zu gar keiner Konvergenz. Besonders problematisch ist der Fall, daß ein Ergebnis nicht offenkundig falsch, etwa physikalisch unsinnig ist, und der Benutzer nicht sofort erkennt, ob es sich um ein richtiges oder falsches Ergebnis handelt [47, 65].

Läßt man *FSOLVE* bei der betrachteten Veresterung von zwei geringfügig anderen Startwerten ausgehen, so ergeben sich bereits gänzlich andere Gleichgewichtszusammensetzungen.

Tab 37: Vergleich zweier *EENL*-Aufrufe durch *FSOLVE* mit veränderten Startwerten mit Literaturwerten [47]

Literaturwerte	<i>FSOLVE</i> -Aufruf		<i>FSOLVE</i> -Aufruf	
	Startwerte	Ergebnisse	Startwerte	Ergebnisse
0.06949	0.1	0.0695	0.2	0.9381
0.06771	0.1	0.0677	0.2	0.0310

Die erhaltenen Ergebnisse können jedoch mit den Gleichgewichtsbedingungen, der Schließbedingung und den Elementarbilanzen überprüft werden. Erfüllen die Ergebnisse zwar die Elementarbilanzen und die Schließbedingung, aber das Gleichgewicht nicht, so kann ein erneutes Einsetzen der erhaltenen Ergebnisse als Startwerte für eine neue Iteration zu der gesuchten Gleichgewichtszusammensetzung führen.

Wird jedoch die Schließbedingung der Molenbrüche nicht erfüllt, so ist ein physikalisch unsinniges Ergebnis gefunden worden. Dies ist bei dem in Tabelle 36 gezeigten *FSOLVE* Aufruf mit den Startwerten [0.2,0.2]' der Fall. Hier muß der Benutzer *FSOLVE* mit anderen Startwerten neu aufrufen.

Eine dritte Möglichkeit, den Erfolg eines *FSOLVE* Aufrufes zu beeinflussen, ist mit der Veränderung der Abbruchtoleranzen gegeben. Beim Aufruf des Gleichungslösers *FSOLVE* kann der Benutzer mit achtzehn verschiedenen Optionen den Suchverlauf beeinflussen. Diese sind in einem Feld zusammengefaßt. Ihre voreingestellten Werte können mit dem Befehl *FOPTIONS* abgerufen werden. Mit den Optionen zwei und drei können die Abbruchtoleranzen für die unabhängigen Variablen und die Funktionswerte eingestellt werden. Voreingestellt sind hier jeweils die Werte 10^{-4} . Verringerte Toleranzen können zum Beispiel eine höhere Genauigkeit beim endgültigen Ergebnis erzwingen.

Der von MATLAB zur Verfügung gestellte Gleichungslöser *FSOLVE* ist ein für die Berechnung chemischer Simultangleichgewichte anwendbares Werkzeug.

5.3.3 Minimierung der Gibbschen Enthalpie

Eine weitere Möglichkeit, simultan ablaufende chemische Gleichgewichte zu berechnen, ist die Minimierung der freien Gibbschen Reaktionsenthalpie:

$$\Delta G = \sum_i^N n_i \cdot g_i \quad [53]$$

Für die partielle molare Freie Gibbssche Enthalpie g_i gilt folgender Zusammenhang:

$$g_i = g_i^0 + R \cdot T \cdot \ln a_i = g_i^0 + R \cdot T \cdot \ln(f_i / f_i^0). \quad [54]$$

Werden nur Systeme mit idealen Gasen berücksichtigt, so ergibt sich mit den Gleichungen [53] und [54] folgender Ausdruck:

$$\Delta G = \sum_i^N n_i \cdot \left[g_i^0 + R \cdot T \cdot \ln \left(\frac{n_i \cdot P}{n} \right) \right]. \quad [55]$$

Neben Gleichung [55] müssen noch die Elementbilanzen

$$\sum_{i=1}^N A_{hi} \cdot n_i = b_h \quad h = 1, \dots, L \quad [37]$$

und die Schließbedingung

$$\sum_{i=1}^N n_i = n \quad [51]$$

als Nebenbedingung berücksichtigt werden. Somit ergibt sich das Problem, ein Minimum mit Nebenbedingungen zu finden. Diese spezielle Fragestellung erlaubt die Anwendung von Lagrange Multiplikatoren [44]. Damit bietet sich die Möglichkeit, das System aus $N + L + 1$ Gleichungen auf ein System mit L Gleichungen zu reduzieren [6], das heißt, es müssen weit weniger Gleichungen simultan gelöst werden. Dieser Vorteil macht sich vor allem bei umfangreichen chemischen Systemen bemerkbar.

Die Notwendigkeit, "gute Startwerte" für die Lösung von nicht-linearen algebraischen Gleichungen zur Bestimmung chemischer

Gleichgewichtszusammensetzungen zu finden, entfällt laut Walas [6] bei der G-Minimierung.

Die Methode der G-Minimierung wird in der Literatur ausführlich diskutiert [6, 4]. Walas [6] beurteilt sie aus obengenannten Gründen als, der direkten Lösung der durch die Gleichgewichtsbeziehungen gegebenen algebraischen Gleichungen, überlegen.

Obwohl die Zahl der zu betrachtenden Gleichungen mit der Einführung der Lagrangen Multiplikatoren stark reduziert werden kann, so handelt es sich dennoch weiterhin um ein Minimierungsproblem. Dabei kommen dieselben iterativen Verfahren zur Anwendung, die auch bei der in Kapitel 5.3.2 angesprochenen direkten Lösung der Gleichgewichtsbeziehungen Verwendung finden.

5.3.4 Berechnung mit Massenwirkungskinetik

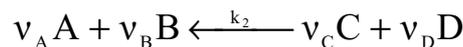
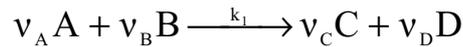
Eine weitere Möglichkeit, chemische Simultangleichgewichte zu berechnen, ist die Relaxationsmethode [2]. Hier werden bei n Reaktionen gedanklich n Reaktoren hintereinander geschaltet. In jedem Reaktor läuft nun eine Reaktion bis zum Gleichgewicht. Gleichzeitig wird die aus allen übrigen Reaktionen resultierende Veränderung der Konzentration vernachlässigt. Diese gedankliche Reaktorkaskade wird so lange durchlaufen, bis sich keine Änderung der Zusammensetzung mehr ergibt. Mit diesem Vorgehen wird das Problem der simultanen Gleichgewichtsreaktionen auf die wiederholte Lösung eines einzelnen Gleichgewichtes reduziert.

Greift man die Vorstellung der bis zum chemischen Gleichgewicht laufenden Reaktoren auf, so kann man zu einem anderen Konzept kommen. In einem Reaktor, der alle reagierenden Komponenten enthält, sollte sich nach entsprechend langer Reaktionszeit ein chemisches Gleichgewicht einstellen. Ein dynamisches Gleichgewicht kann weiterhin aus zwei gegenläufigen Reaktionen zusammengesetzt verstanden werden. Eine Gleichgewichtsreaktion kann somit durch zwei Reaktionsgleichungen ausgedrückt werden. Die-

ser Ansatz ist als Massenwirkungskinetik bekannt. Die Gleichgewichtsreaktion



wird durch die beiden Reaktionen



dargestellt. Weiterhin kann man zur Betrachtung der zeitlichen Änderung der Konzentrationen Differentialgleichungen (kurz DGI) heranziehen. Für das angeführte Beispiel ergeben sich folgende Bilanzgleichungen, wobei der Übersichtlichkeit halber auf die Anwendung des Konzeptes der Schlüsselreaktionen verzichtet wird:

$$\begin{aligned} \frac{\partial c_A}{\partial t} &= k_2 \cdot c_C^{v_C} \cdot c_D^{v_D} - k_1 \cdot c_A^{v_A} \cdot c_B^{v_B} \\ \frac{\partial c_B}{\partial t} &= k_2 \cdot c_C^{v_C} \cdot c_D^{v_D} - k_1 \cdot c_A^{v_A} \cdot c_B^{v_B} \\ \frac{\partial c_C}{\partial t} &= k_1 \cdot c_A^{v_A} \cdot c_B^{v_B} - k_2 \cdot c_C^{v_C} \cdot c_D^{v_D} \\ \frac{\partial c_D}{\partial t} &= k_1 \cdot c_A^{v_A} \cdot c_B^{v_B} - k_2 \cdot c_C^{v_C} \cdot c_D^{v_D} \end{aligned} \quad [56]$$

Führt man beide Überlegungen zusammen, so ergibt sich eine weitere Möglichkeit, chemische Gleichgewichte iterativ zu berechnen. Sie verbindet die Vorstellung eines Reaktors, in dem die Reaktionen bis zum Gleichgewicht ablaufen, mit der Behandlung einer Massenwirkungskinetik der betrachteten Gleichgewichte durch Differentialgleichungen. Dieser Ansatz liegt der Funktion *EE* (Equilibrium Estimate) zugrunde. *EE* muß ähnlich *EENL* als Funktionsfile für einen Gleichungslöser verwendet werden. Im Unterschied zu *EENL* wird *EE* mit einem numerischen DGI-Löser aufgerufen. Diese sind im Gegensatz zu numerischen Optimieral-

gorithmen weit stabiler und zuverlässiger. So ist das Konvergenzverhalten des Beispiels der thermischen Dehydrierung von Methan weitgehend unabhängig von der Wahl der notwendigerweise physikalisch sinnvollen Startzusammensetzung.

Die Funktion *EE* erstellt selbstständig die zur Beschreibung des Gleichgewichtes erforderlichen Gleichungen. Dazu sind neben der stöchiometrischen Matrix \underline{N} der betrachteten Reaktionen die Hilfsparameter `pos` und `neg` erforderlich. Sie sind mit Hilfe der Funktion *N2RXGLMAT* zugänglich.

Tab 38: I/O-Charakteristik der Funktion *N2RXGLMAT*

<i>N2RXGLMAT</i>	nach Vorzeichen sortierte Ausgabe von stöchiometrischen Koeffizienten einer stöchiometrischer Matrix \underline{N}
Eingabe:	Stöchiometrische Matrix der Reaktionen
Ausgabe:	Positionsangaben der positiven und negativen stöchiometrischen Koeffizienten

`pos` und `neg` enthalten Positionsangaben der negativen und positiven stöchiometrischen Koeffizienten in der Matrix \underline{N} . Damit entfällt das, in *N2RXGLMAT*, durch zwei verschachtelte *FOR/NEXT*-Schleifen realisierte Sortieren im ODE-File. Dies ist gerade bei größeren Reaktionssystemen zeitlich von Vorteil.

Tab 39: I/O-Charakteristik der Funktion *EE*

<i>EE</i> :	Berechnung chemischer Simultangleichgewichte
Eingabe:	unabhängige Variable <i>t</i> , Vektor der Konzentrationen der Schlüsselkomponenten, Integrationsoption <i>FLAG</i> , Hilfsparameter <i>pos</i> und <i>neg</i> , Stöchiometrische Matrix der Reaktionen, Startstoffmenge, Vektor der Startkonzentration der Schlüsselkomponenten, Vektor der Startkonzentration der Nicht-Schlüsselkomponenten, Vektor der Gleichgewichtskonstanten, Methodenschalter für Aktivitätskoeffizientenberechnung, Gas-/Flüssig-Phasen Reaktionsschalter, Reaktionstemperatur, zwei Argumente zur γ Berechnung
Ausgabe:	Auf die Schlüsselreaktionen bezogener Funktionsvektor entsprechend den Gleichungen [56]

(**Fette Schrift** = optionale Argumente)

Tabelle 40 zeigt als weiteres Beispiel die gute Übereinstimmung berechneter Gleichgewichtsmolenbrüche der Verbrennung von Propan mit Literaturwerten [36]. Die recht großen Unterschiede der relativen Abweichungen resultieren aus der Größe der absoluten Werte (s. Werte für N_2 und CO).

Tab 40: Vergleich der Berechnung der Verbrennung von Propan bei 2200 K mit Literaturwerten [36]

Molenbrüche der Substanzen	mit <i>EE</i> berechnet	Literaturwerte	relative Abweichung
CO_2	0.1077415	0.1077414	$5.89 \cdot 10^{-7}$
H_2O	0.1463639	0.1463641	$1.27 \cdot 10^{-6}$
N_2	0.7385305	0.7385306	$1.39 \cdot 10^{-7}$
CO	0.0031140	0.0031141	$2.14 \cdot 10^{-5}$
H_2	0.0008166	0.0008164	$2.49 \cdot 10^{-4}$
H	0.0000267	0.0000267	$1.18 \cdot 10^{-5}$
OH	0.0012268	0.0012268	$1.03 \cdot 10^{-5}$
O	0.0000155	0.0000155	$2.21 \cdot 10^{-5}$
NO	0.0010131	0.0010130	$2.22 \cdot 10^{-5}$
O_2	0.0011509	0.0011509	$4.67 \cdot 10^{-5}$

Je nachdem, ob der Benutzer eine Gasphasen- oder eine Flüssigphasenreaktion betrachten möchte, werden in *EE* ein idealer Rohrreaktor mit konstantem Druck oder ein absatzweise arbeitender idealer Rührkessel mit konstantem Volumen simuliert. Mittels eines Schalterargumentes kann der Benutzer zwischen den beiden Möglichkeiten entscheiden. Beim Aufruf des DGI-Lösers muß eine Zeitspanne festgelegt werden, über die die Berechnung ausgeführt werden soll. Eine graphische Auftragung der Berechnungsergebnisse gibt schnell Auskunft darüber, ob das chemische Gleichgewicht bereits erreicht wurde. Das heißt die Konzentrationen der betrachteten Substanzen ändern sich zeitlich nicht mehr. Ist dies nicht der Fall, so muß die Integration mit größerer Laufzeit erneut gestartet werden. Die bei dem ersten Programmlauf erhaltenen Konzentrationen können dabei als Startwerte für die neue Berechnung verwendet werden. Die Integrationszeit kann jedoch problemlos groß genug gewählt werden, da das Ergebnis dadurch nicht verändert wird. Weiterhin bedingt eine Verlängerung des Integrationsintervalls nicht zwangsläufig eine Erhöhung der notwendigen Rechenzeit. Die aufzuwendende Rechenzeit ist stark von der Wahl des geeigneten DGI-Lösers abhängig.

Eine weitere Option beim Einsatz von *EE* ist die Verwendung von Aktivitätskoeffizienten bei der Bestimmung der Gleichgewichtszusammensetzung von Reaktionen in flüssiger Phase. Die dafür notwendigen Parameter der g^E -Methoden Wilson, NRTL oder UNIQUAC (s. Kap. 6) können als optionale Argumente am Ende der Argumentenliste eingegeben werden.

Zur Integration von DGI-Systemen stellt *MATLAB* dem Benutzer verschiedene Gleichungslöser zur Verfügung. Je nach der Beschaffenheit des Gleichungssystems kann der Benutzer unter verschiedenen ODE-Solvern (Ordinary Differentiel Equation) wählen. Es wird zwischen Lösungsverfahren für steife und nicht steife DGIs unterschieden. Unter einem steifen DGI-System versteht man, daß sich, bezogen auf die Zeitspanne des Integrationsintervalls, der Wert einer DGI schnell ändern kann, aber der Wert einer weiteren DGI ü-

ber eine sehr viel längere Zeitspanne variiert [46]. Nicht für steife Systeme ausgelegte Lösungsverfahren sind hier wenig effektiv, da sie für die sich langsam ändernden Bereiche Zeitschritte benutzen, die klein genug sind, auch die schnelle Änderung zeitlich aufzulösen. Für steife DGIs kommen die Gleichungslöser *ODE15s* und *ODE23s* zur Anwendung, bei nicht steifen Systemen sind dies *ODE45*, *ODE23* und *ODE113*.

Als Beispiel sei die Bilanzierung von zwei Folgereaktionen angeführt, wobei k_2 um den Faktor 10^3 größer ist als k_1 :



Es ergibt sich ein System aus drei DGIs, die unterschiedlich schnell einem konstanten Wert zulaufen.

Mit dem folgendem einfachen M-File läßt sich der zeitliche Verlauf der Konzentrationsänderungen von A, B und C verfolgen.

```
function re = odetestfile(t,y)
k1=1;
k2=1000;
re(1)=-k1*y(1);
re(2)=k1*y(1)-k2*y(2);
re(3)=k2*y(2);
re=re';
% end function
```

Die Funktion *ODETESTFILE* muß nun von einem ODE-Solver aufgerufen werden. Daher tragen Funktionen wie diese auch den Namen ODE-File. Wird das ODE-File mit dem ODE-Solver *ODE45* aufgerufen, so benötigt die Berechnung ca. 13 Sekunden.

```

» y0=[1,0,0];
» tic,[t,y]=ode45('odetestfile',[0 6],y0);toc

elapsed_time =

    13.9600

```

Bei dem Aufruf des ODE-Solvers *ODE45* werden das Odefile *ODETESTFILE* als Zeichenkette, das Integrationsintervall [0 6] und die Startwerte *y0* angegeben. Verwendet man hingegen den ODE-Solver *ODE15S*, der auf die Behandlung steifer Systeme spezialisiert ist, so läuft die Berechnung innerhalb von einer halben Sekunde ab. Die für die Berechnung benötigte Zeit wurde dabei unter Verwendung der Funktionen *tic* und *toc* ermittelt und ausgegeben. Dieser zunächst vielleicht gering erscheinende Unterschied macht sich bei umfangreicheren DGI-Systemen oder weit längeren Integrationsintervallen jedoch viel stärker bemerkbar. Wichtig sind diese speziellen Verfahren im Hinblick auf unterschiedlich schnell ablaufende chemische Gleichgewichte und auf die Lösung von differential algebraischen Gleichungen (s. Seite 113).

Aufgrund der größeren Stabilität der DGI-Lösungsverfahren bietet die Berechnung der Gleichgewichtszusammensetzung unter Anwendung einer Massenwirkungskinetik eine gute Alternative zu den auf Optimierverfahren basierenden Methoden. Die bei der direkten Lösung algebraischer Gleichungen (s. Kap. 5.3.2) auftretende Notwendigkeit guter Startwerte wird bei der Behandlung simultaner Gleichgewichte mit DGIs gänzlich umgangen. Es resultiert für den Benutzer somit eine weit größere Zuverlässigkeit der Simulationsergebnisse.

Bei der Betrachtung von simultanen Gleichgewichtsreaktionen ergibt sich die Frage nach der Behandlung von nicht idealem Verhalten des Reaktionsgemisches. Dabei sind zwei Fälle zu unterscheiden.

Für Reaktionen in flüssiger Phase werden die Aktivitäten a_i durch folgende Zusammenhänge wiedergegeben [6]:

$$a_i = \frac{\gamma_i \cdot x_i \cdot f_i}{f_i^0} = \gamma_i \cdot x_i \cdot \exp\left(V_{i,L} \cdot \frac{P-1}{R \cdot T}\right), \quad [57]$$

wobei die reinen Flüssigkeiten bei 1 atm als Standardzustände betrachtet werden. Der Exponentialausdruck in Gleichung [57] wird als Poynting Faktor bezeichnet und berücksichtigt den Einfluß einer Druckveränderung auf die Fugazität einer Flüssigkeit. Bei geringen Drücken kann dieser Korrekturterm vernachlässigt werden. Damit ergibt sich die Gleichgewichtskonstante zu

$$K_E = K_a = K_\gamma \cdot K_x = \prod_{i=1}^N (x_i \cdot \gamma_i)^{\nu_i}. \quad [58]$$

Die Aktivitätskoeffizienten γ_i sind durch g^E -Methoden (s. Kap. 6) zugänglich, für die Reinstoff- und Gemischparameter in großem Umfang tabelliert sind [24].

Die Gleichungen für die Aktivitätskoeffizienten sind explizit. Somit können die γ_i bei jedem Integrationsschritt neben den aus der Massenwirkungskinetik resultierenden DGLs berechnet werden. Dieses Vorgehen wurde bei den Programmen *EE* und *EENL* als optionale Möglichkeit realisiert.

Um das Verhalten einer realen gasförmigen Reaktionsmischung zu berücksichtigen, wird ein anderer Weg beschritten. In diesem Fall werden allein die Fugazitäten betrachtet. Damit ergibt sich die Gleichgewichtskonstante zu

$$K_E = K_f = K_\phi \cdot K_p = K_\phi \cdot K_x \cdot P^{\Delta\nu} = \prod_{i=1}^N (x_i \cdot \phi_i \cdot P)^{\nu_i}. \quad [59]$$

Die Fugazitätskoeffizienten sind über Gleichung [32] (s. Kap. 4.4) zugänglich:

$$R \cdot T \cdot \ln \varphi_i = - \int_{\infty}^V \left[\left(\frac{\partial P}{\partial n_i} \right)_{T, V, n_{j \neq i}} - \frac{R \cdot T}{V} \right] dV - R \cdot T \cdot \ln Z. \quad [32]$$

Zur Behandlung des Differentialquotienten $\partial P / \partial n_i$ in Gleichung [32] muß eine Zustandsgleichung herangezogen werden. Für diese Anwendung wurden Gleichungen des kubischen Typs (s. Kap. 4.1 und 4.4) verwendet. Dies bedeutet, daß bei jedem Zeitschritt des DGI-Lösers neben den DGIs auch algebraische Gleichungen gelöst werden müssen. Es muß ein System Differential-Algebraischer-Gleichungen, ein DAGI-System, gelöst werden.

Diese Klasse von Gleichungen stellt ein besonderes Problem bei der numerischen Behandlung von DGI dar. Beim Einsatz normaler DGI-Löser bieten sich mehrere mögliche Lösungen. Prinzipiell können bei jedem Schritt des Lösungsalgorithmus die algebraischen Gleichungen gelöst werden, etwa mit *FSOLVE*. Dieses Vorgehen ist aber wegen des damit verbundenen großen Zeitaufwandes nicht praktikabel.

Ein anderer möglicher Weg ist die Umwandlung des DAGI-Systems in ein reines DGI-System. Dabei werden die DAGIs durch folgenden Zusammenhang wiedergegeben:

$$\underline{f}(\underline{y}', \underline{y}, \underline{x}, t) = 0. \quad [60]$$

Mit \underline{y}' ist die Ableitung der Variablen \underline{y} nach der Zeit t bezeichnet. Der Vektor \underline{x} repräsentiert dabei die zeitlich indifferenten Variablen. So sind die algebraischen Gleichungen charakterisiert durch

$$\underline{g}(\underline{y}, \underline{x}, t) = 0. \quad [61]$$

Differenziert man nun die algebraischen Gleichungen nach t ,

$$-\frac{\partial \underline{g}}{\partial t} = \frac{\partial \underline{g}}{\partial \underline{x}} \cdot \frac{d\underline{x}}{dt} + \frac{\partial \underline{g}}{\partial \underline{y}} \cdot \frac{d\underline{y}}{dt}, \quad [62]$$

und ist die Jacobi-Matrix $\partial \underline{g} / \partial \underline{x}$ nicht-singulär, so findet sich ein Ausdruck für die zeitliche Änderung der algebraischen Variablen \underline{x} :

$$\frac{d\underline{x}}{dt} = -\left(\frac{\partial \underline{g}}{\partial \underline{x}}\right)^{-1} \cdot \frac{\partial \underline{g}}{\partial t} - \left(\frac{\partial \underline{g}}{\partial \underline{x}}\right)^{-1} \cdot \frac{\partial \underline{g}}{\partial \underline{y}} \cdot \frac{d\underline{y}}{dt}. \quad [63]$$

Das ursprüngliche DAGI-System hat in diesem Fall den Index eins. Ist hingegen die Matrix $\partial \underline{g} / \partial \underline{x}$ singulär, so kann man versuchen Gleichung [62] durch geeignete algebraische Manipulationen wieder in die allgemeine Form eines DAGI-Systems zu bringen, um den algebraischen Teil erneut zu differenzieren. Ist die damit entstehende Jacobi-Matrix nicht-singulär, so hat das ursprüngliche DAGI-System den Index zwei. Ist auch diese Jacobi-Matrix singulär, so muß man mit diesem Vorgehen weiter fortfahren, bis sich $d\underline{x}/dt$ als kontinuierliche Funktion von \underline{x} und t ergibt. Die Anzahl der dazu notwendigen Differentiationen ist gleich dem Index des ursprünglichen DAGI-Systems. Dieses Verfahren wird jedoch für die Lösung von DAGI-Systemen mit höherem Index nicht empfohlen [42]. Daher beschränken sich die Betrachtungen in diesem Kapitel ausschließlich auf DAGI-Systeme mit Index eins.

Die derart erhaltenen 'neuen' DGIs können dann zusammen mit den bereits bekannten DGIs mit einem DGI-Löser behandelt werden. Um mit diesem Vorgehen zum Erfolg zu gelangen, ist ein konsistenter Satz von Anfangsbedingungen erforderlich. Durch die Umwandlung der DAGIs ist zwar ein reines DGI-System entstanden, die physikalischen Zusammenhänge haben sich jedoch nicht verändert.

Da in MATLAB 5.2 noch kein wirklicher DAGI-Löser zur Verfügung stand, wurde obiges Vorgehen unter Verwendung der *Symbolic Math Toolbox* realisiert [48]. Dabei werden umfangreiche Routinen implementiert, die es dem Benutzer ermöglichen, Systeme mit einer beliebigen Zahl von Reaktanden flexibel symbolisch zu behandeln. Bei der Behandlung großer Systeme mit der *Symbolic Math Toolbox* zeigt sich jedoch ein Problem. Bei der Bearbeitung großer Formelsysteme - gemäß Gleichung [63] - entstehen sehr große symbolische Ausdrücke. Diese müssen bei der Behandlung des neu erhaltenen DGI-Systems mit einem DGI-Solver durch den Befehl *EVAL* im ODE-File berechnet werden, damit der aufrufenden Funktion numerische Werte zurückgegeben werden können. Als problematisch erweisen sich dabei neben der recht langen Rechenzeit für die vor der Integration durchgeführten symbolischen Berechnungen vor allem geringfügige Abweichungen zwischen analytischer und numerischer Lösung. Diese Diskrepanzen liegen im Bereich der Rechengenauigkeit von 32 Nachkommastellen. Durch die Verarbeitung sehr großer Formelsysteme kumulieren diese kleinen Abweichungen und führen zu einem Abbruch der Berechnung aufgrund einer Segmentverletzung. Auch eine Erhöhung der Rechengenauigkeit bringt hier keine Verbesserung.

Dieser Lösungsansatz wird nicht weiter verfolgt, da MATLAB 5.3 einen DAGI-Löser zur Verfügung stellt. Alle bis dahin unter MATLAB angebotenen ODE-Solver sind reine DGI-Lösungsverfahren und können nicht oder nur unter größerem Aufwand des Benutzers DAGI-Systeme berechnen. So bietet sich neben der Umwandlung von DAGIs in DGIs beispielsweise noch die Möglichkeit, externe DAGI-Löser als sogenannte MEX-Files in ein SIMULINK-Modell einzubinden [49].

MATLAB 5.3 bietet mit Weiterentwicklungen der DGI-Löser *ODE23s* und *ODE15s* die Möglichkeit DAGIs mit Index eins zu berechnen. Dazu muß eine sogenannte "mass matrix" M definiert werden:

$$M(t) \cdot y' = F(t, y). \quad [64]$$

M enthält Informationen, die es ermöglichen, auch ein DAGL-System zu beschreiben. Dazu müssen die algebraischen Gleichungen entsprechend Gleichung [61] arrangiert

$$\underline{g}(\underline{y}, \underline{x}, t) = 0 \quad [61]$$

und im aufzurufenden ODE-File neben die DGls gestellt werden. Dann kann durch eine geeignete "mass matrix" M(t) ein DAGL-System beschrieben und zum Beispiel mit *ODE15s* berechnet werden.

Für ein System aus zwei algebraischen und vier Differentialgleichungen ergibt sich das folgende Schema für M(t):

$$M(t) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad [65]$$

Dabei wird davon ausgegangen, daß die algebraischen Gleichungen den DGls vorangestellt werden. Damit ergeben sich an den Positionen eins und zwei reine Null-Zeilen für die Matrix M (s. Gl. [65]).

Realisiert wurde eine Gleichung [64] entsprechende Berechnung in dem File *EEM*. Es ermöglicht die Berechnung chemischer Gleichgewichtsreaktionen in der Gasphase unter Berücksichtigung von Fugazitäten. Diese werden mittels kubischer Gleichungen und den entsprechenden thermodynamischen Zusammenhängen (Gleichung [32], s. Kapitel 4.4) bestimmt. Hier kann der Benutzer wieder zwischen den verschiedenen kubischen Gleichungen nach Van-der-Waals, nach Redlich-Kwong, nach Soave-Redlich-Kwong und nach Peng-Robinson wählen. Diese bei jedem Schritt des Integrati-

onsprogrammes zu lösende Gleichung stellt somit die algebraische Komponente des DAGI-Systems dar.

Tab 41: I/O-Charakteristik der Funktion *EEM*

<i>EEM</i> :	Berechnung chemischer Gasphasen-Simultangleichgewichte unter Berücksichtigung von Fugazitäten
Eingabe:	unabhängige Variable <i>t</i> , Vektor der Molenbrüche der Schlüsselkomponenten, Integrationsoption <i>FLAG</i> , Methodenschalter für Fugazitätskoeffizientenberechnung, Hilfsparameter <i>pos</i> und <i>neg</i> , Stöchiometrische Matrix der Reaktionen, Startstoffmenge, Vektor der Startmolenbrüche der Schlüsselkomponenten, Vektor der Startmolenbrüche der Nicht-Schlüsselkomponenten, Gesamtdruck, Vektor der Gleichgewichtskonstanten, Reaktionstemperatur, <u>kritische Temperatur, kritischer Druck, kritischer azentrischer Faktor, binäre Interaktionsparameter</u>
Ausgabe:	Auf die Schlüsselreaktionen bezogener Funktionsvektor entsprechender Gleichungen [63]

(**Fette Schrift** = optionale Argumente, unterstrichene Schrift = alternative Eingabemöglichkeit, siehe weitere Erläuterungen im Text und Kap. 1.2)

Als Anwendungsbeispiel für *EEM* soll die Synthese von Ammoniak bei 450°C und 300 bar betrachtet werden (s. Tab. 42). Ausgegangen wird dabei von einer Mischung aus Wasserstoff und Stickstoff mit $x = [0.25, 0.75]$.

Tab 42: Gleichgewichtszusammensetzungen der Synthese von Ammoniak bei 450°C und 300 bar, berechnete Werte und Literaturwerte [83]

	Literaturwerte	berechnete Werte	Fehler in Prozent
N ₂	0.1613	0.1608	0.310
H ₂	0.4837	0.4825	0.248
NH ₃	0.3550	0.3567	0.479

Zur Anwendung kommt bei obiger Berechnung die kubische Gleichung nach Soave-Redlich-Kwong.

6 Dampf-Flüssig-Gleichgewichte

Die Betrachtung von Dampf-Flüssig-Gleichgewichten wird im allgemeinen nicht als Teilgebiet der Reaktionstechnik erachtet. Sie wird häufig bei der Trennung von Flüssigkeitsgemischen in die reinen Komponenten - zum Beispiel durch Rektifikation - notwendig. Dennoch sollen die zugrundeliegenden Formelsysteme hier behandelt werden, da die Übergänge zwischen der Reaktionstechnik und verfahrenstechnischen Operationen nicht immer eindeutig sind. Reaktionskolonnen stellen ein solches Übergangsgebiet dar [51, 84, 85]. Für die Simulation eines solchen Reaktionsapparates werden die Aktivitätskoeffizienten der betrachteten Substanzen benötigt. Für deren Berechnung können entweder Zustandsgleichungen oder eine auf dem Konzept der molaren Gibbsschen Exzeß Enthalpie (g^E -Konzept) basierende Gleichung verwendet werden. Für ein Gemisch aus n Substanzen liefert die Gibbs-Duhem-Gleichung folgenden Zusammenhang [1]:

$$R \cdot T \cdot \ln \gamma_i = \left(\frac{\partial g^E}{\partial n_i} \right)_{T, P, n_{j \neq i}} . \quad [66]$$

Mit einem geeigneten mathematischen Ausdruck für g^E ermöglicht Gleichung [66] die Interpolation und Extrapolation begrenzter Dampf-Flüssig-Daten als Funktion der Zusammensetzung. Zu den am weitesten verbreiteten g^E -Gleichungen gehören die NRTL-Gleichung (Non Random Two Liquid), die UNIQUAC-Gleichung (Universal Quasi Chemical) und die Gleichung nach Wilson. Weiterhin werden hier noch die beiden Gleichungen nach Margules und nach van Laar berücksichtigt.

Tab 43: Tabelle der implementierten Gleichungen zur Berechnung von Aktivitätskoeffizienten [24].

Name	Parameter	$\ln \gamma_i =$
Margules	A_{12} A_{21}	$\ln \gamma_1 = [A_{12} + 2 \cdot (A_{21} - A_{12}) \cdot x_1] \cdot x_2^2$ $\ln \gamma_2 = [A_{21} + 2 \cdot (A_{12} - A_{21}) \cdot x_2] \cdot x_1^2$
van Laar	A_{12} A_{21}	$\ln \gamma_1 = A_{12} \cdot \left(\frac{A_{21} \cdot x_2}{A_{12} \cdot x_1 + A_{21} \cdot x_2} \right)^2$ $\ln \gamma_2 = A_{21} \cdot \left(\frac{A_{12} \cdot x_1}{A_{12} \cdot x_1 + A_{21} \cdot x_2} \right)^2$
Wilson	$\lambda_{ij} - \lambda_{ii}$ $\lambda_{ji} - \lambda_{jj}$ V_i^L bzw. Λ_{ij}	$\Lambda_{ij} = \frac{V_j^L}{V_i^L} \cdot \exp\left(-\frac{\lambda_{ij} - \lambda_{ii}}{R \cdot T}\right)$ $\Lambda_{ii} = \Lambda_{jj} = 1$ $\ln \gamma_i = -\ln\left(\sum_{j=1}^m x_j \cdot \Lambda_{ij}\right) + 1 - \sum_{k=1}^m \frac{x_k \cdot \Lambda_{ki}}{\sum_{j=1}^m x_j \cdot \Lambda_{kj}}$
NRTL	$g_{ji} - g_{ii}$ $g_{ij} - g_{jj}$ α_{ij} wobei $g_{ii} - g_{ii} = 0$ und $g_{jj} - g_{jj} = 0$	$\tau_{ij} = \frac{g_{ij} - g_{jj}}{R \cdot T}, \quad \ln G_{ij} = -\alpha_{ij} \cdot \tau_{ij}$ $A = \frac{\sum_{j=1}^m \tau_{ji} \cdot G_{ji} \cdot x_j}{\sum_{l=1}^m G_{li} \cdot x_l}$ $\ln \gamma_i = A + \sum_{j=1}^m \frac{G_{ij} \cdot x_j}{\sum_{l=1}^m G_{lj} \cdot x_l} \cdot \left(\tau_{ij} - \frac{\sum_{n=1}^m x_n \cdot \tau_{nj} \cdot G_{nj}}{\sum_{l=1}^m G_{lj} \cdot x_l} \right)$

Tab. 43: Tabelle der implementierten Gleichungen zur Berechnung von Aktivitätskoeffizienten [24]. (Fortsetzung)

UNIQUAC	$u_{ij}-u_{jj}$	$\ln \gamma_i^C = \ln \frac{\theta_i}{x_i} + \frac{z}{2} \cdot q_i \cdot \ln \frac{\vartheta_i}{\theta_i} + l_i - \frac{\theta_i}{x_i} \cdot \sum_{j=1}^m x_j \cdot l_j$
	wobei	
	$\tau_{ii} = 1$	
	$\tau_{jj} = 1$	
	q_i	
r_i	$\theta_i = \frac{r_i \cdot x_i}{\sum_{i=1}^N r_i \cdot x_i} \quad \vartheta_i = \frac{q_i \cdot x_i}{\sum_{i=1}^N q_i \cdot x_i} \quad \ln \tau_{ij} = \frac{u_{ij} - u_{jj}}{R \cdot T}$	
	$\ln \gamma_i^R = q_i \cdot \left[1 - \ln \left(\sum_{j=1}^m \vartheta_j \cdot \tau_{ji} \right) - \sum_{j=1}^m \frac{\vartheta_j \cdot \tau_{ij}}{\sum_{k=1}^m \vartheta_k \cdot \tau_{kj}} \right]$	
	$\ln \gamma_i = \ln \gamma_i^C + \ln \gamma_i^R \quad \text{mit}$	
	$l_i = \frac{z}{2} \cdot (r_i - q_i) - (r_i - 1) \quad z = 10$	

Die Namen der erstellten MATLAB Funktionen zur Berechnung obiger Formelsysteme lehnen sich an die Bezeichnungen der betrachteten Gleichung an. Sie ermöglichen die Berechnung von Aktivitätskoeffizienten in Flüssigkeitsgemischen. Dabei sind die Methoden nach Wilson, die NRTL-Methode und die UNIQUAC-Methode auch für Gemische mit mehr als zwei Substanzen einsetzbar. Vertiefende Informationen zur Theorie der betrachteten Methoden finden sich in [1, 6, 24].

Als Anwendungsbeispiel soll hier ein Gemisch aus Cyclohexan und Anilin bei 50.2°C betrachtet werden. Die Aktivitätskoeffizienten werden mit der Funktion *WILSON* berechnet.

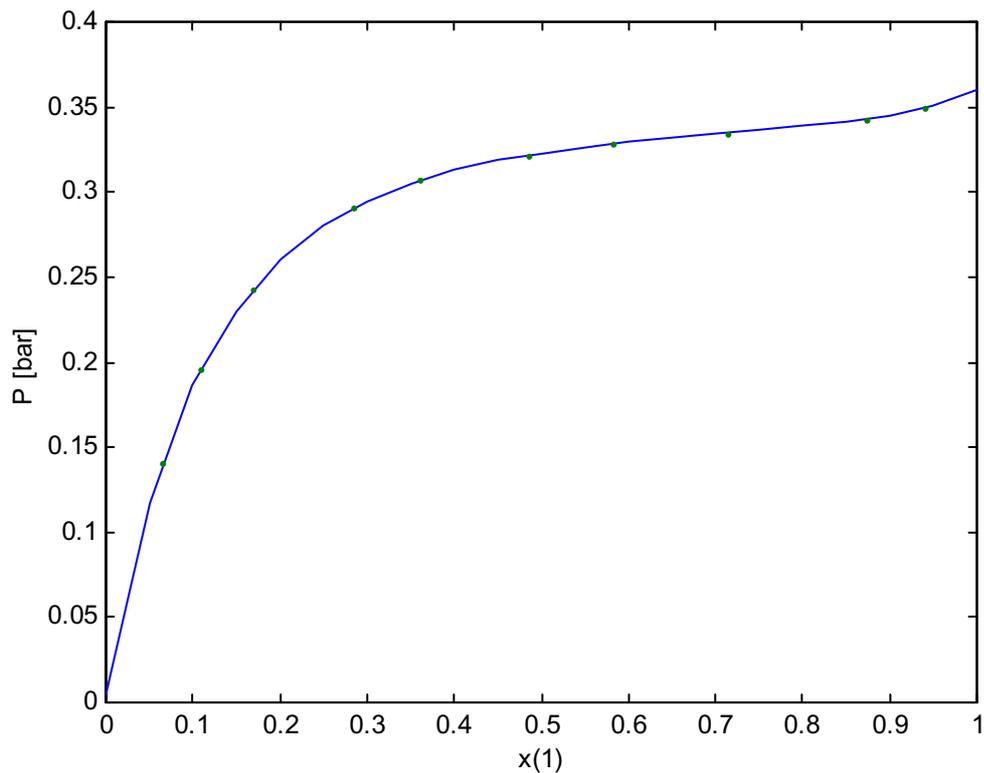


Abb. 10: Vergleich unter Anwendung von *WILSON* berechneter und experimenteller (Punkte) Dampfdrücke des Systems Cyclohexan/Anilin bei 50.2°C

In Abbildung 10 ist die gute Übereinstimmung experimenteller Werte [1] mit den Ergebnissen der Funktion *WILSON* ersichtlich. Der mittlere Fehler der Abbildung 10 zugrundeliegenden Berechnung liegt unter 1 %. Es muß daran erinnert werden, daß die implementierten Verfahren Formelsysteme darstellen, die an experimentelle Dampf-Flüssig-Gleichgewichtsdaten angepaßt wurden. Somit können ihre Ergebnisse nur so gut sein wie die aus der Anpassung resultierenden Parameter.

Tab 44: I/O-Charakteristik der Funktionen *MARGULES*, *VANLAAR*, *UNIQUAC*, *NRTL* und *WILSON*.

<i>MARGULES</i> :	Berechnung von Aktivitätskoeffizienten nach Margules
Eingabe:	Molenbruchvektor der betrachteten Lösung, Vektor/ Matrix der binären Interaktionsparameter
<i>VANLAAR</i> :	Berechnung von Aktivitätskoeffizienten nach Van Laar
Eingabe:	Molenbruchvektor der betrachteten Lösung, Vektor/ Matrix der binären Interaktionsparameter
<i>UNIQUAC</i> :	Berechnung von Aktivitätskoeffizienten nach der UNIQUAC-Gleichung
Eingabe:	Molenbruchvektor der betrachteten Lösung, Temperatur, Vektor/ Matrix der binären UNIQUAC Interaktionsparameter, Flächenparameter der betrachteten Substanzen, Volumenparameter der betrachteten Substanzen
<i>WILSON</i> :	Berechnung von Aktivitätskoeffizienten nach der Wilson-Gleichung
Eingabe:	Molenbruchvektor der betrachteten Lösung, Temperatur, Vektor/ Matrix der binären Wilson Interaktionsenergie, molare Volumen der reinen flüssigen Substanzen
<i>NRTL</i> :	Berechnung von Aktivitätskoeffizienten nach der NRTL-Gleichung
Eingabe:	Molenbruchvektor der betrachteten Lösung, Temperatur, Vektor/ Matrix der binären NRTL Interaktionsparameter, Bestimmtheitsparameter
Ausgabe:	Vektor der Aktivitätskoeffizienten

(**Fette Schrift** = optionale Argumente)

Die Unterscheidung zwischen Vektoren und Matrizen in der Argumenteliste obiger Funktionen bezieht sich auf die unterschiedlichen Eingabemöglichkeiten für die binären Interaktionsparameter. So können z. B. die neun UNIQUAC-Parameter eines Dreistoffgemisches entweder als vollständige Matrix [67]

$$\mathbf{u}_{ij} = \begin{bmatrix} \mathbf{u}_{11} & \mathbf{u}_{12} & \mathbf{u}_{13} \\ \mathbf{u}_{21} & \mathbf{u}_{22} & \mathbf{u}_{23} \\ \mathbf{u}_{31} & \mathbf{u}_{32} & \mathbf{u}_{33} \end{bmatrix} \quad [67]$$

oder in einer fortlaufenden Reihenfolge eingegeben werden

$$\mathbf{u}_{ij} = [\mathbf{u}_{12}, \mathbf{u}_{13}, \mathbf{u}_{21}, \mathbf{u}_{23}, \mathbf{u}_{31}, \mathbf{u}_{32}]. \quad [68]$$

Die Eingabe binärer Parameter kann schnell sehr unübersichtlich werden. Um diese Eingabe zu erleichtern, wurde die Funktion *INMATRIX* programmiert. Sie fragt nacheinander alle Positionen der fraglichen Parametermatrix ab und trägt diese auf den entsprechenden Positionen ein.

Tab 45: I/O-Charakteristik der Funktion *INMATRIX*

<i>INMATRIX</i> :	Gesteuerte Erstellung einer $n * n$ Matrix aus benutzerdefinierten Eingaben
Eingabe:	Zeilen-/Spaltenzahl n , konstanter Wert der Hauptdiagonalen
Ausgabe:	$n * n$ Matrix der benutzerdefinierten Eingaben

(**Fette Schrift** = optionale Argumente)

INMATRIX ist auch für die Definition anderer symmetrischer Matrizen - zum Beispiel bei der Eingabe binärer Wechselwirkungsparametern von Zustandsgleichungen - anwendbar (s. Kap. 4.1).

7 Molekulare Transportvorgänge

Tab 46: Zusammenfassung der Transportgrößen bei Gasen [12]

Effekt	transportierte Größe	Gesetzte	Gleichung
Diffusion	Masse, hier Teilchenzahl	$J = -D \cdot \frac{dN_i}{dz}$	[69]
Viskosität	Impuls	$J = -\eta \cdot \frac{du_x}{dz}$	[70]
Wärmeleitung	Energie	$J = -\lambda \cdot \frac{dT}{dz}$	[71]

7.1 Thermische Leitfähigkeit

Eine detaillierte Diskussion verschiedener Abschätzungsverfahren findet sich bei Reid, Prausnitz und Poling [1]. Implementiert wurden die Gleichungssysteme nach Chung und nach Ely-Hanley, da sie über einen weiten Temperatur- und Dichtebereich anwendbar sind. Dennoch sind die vorgeschlagenen Methoden nicht für polare Substanzen geeignet; auch einige unpolare Stoffe zeigen größere Abweichungen von den gemessenen Werten. Ähnliche Ungenauigkeiten ergeben sich bei allen breiter anwendbaren Abschätzungsverfahren. Des weiteren sind Methoden zur Abschätzung der thermischen Leitfähigkeiten von Flüssigkeiten implementiert worden.

7.1.1 Methode nach Ely-Hanley

Das Verfahren [38, 39, 56] geht davon aus, daß die kennzeichnenden Eigenschaften einer einphasigen, fluiden Mischung zu denen einer hypothetischen reinen Referenzsubstanz in Beziehung gesetzt werden können. Die Eigenschaften des Referenzfluids Methan bei dem entsprechenden Temperatur- und Dichteverhältnis werden dabei durch die Methode der übereinstimmenden Zustände beschrieben. Die mathematische Beschreibung der Methode nach Ely-Hanley ist in Anhang A enthalten. Das File *TCELYHANLEY* verwendet diese Methode zur Abschätzung von thermischen Leitfähigkeiten.

Tab 47: I/O-Charakteristik der Funktion *TCELYHANLEY*

<i>TCELYHANLEY</i> :	Abschätzung thermischer Leitfähigkeiten mittels der Methode nach Ely und Hanley
Eingabe:	Molenbrüche, Temperatur, molares Volumen, kritisches Volumen, azentrischer Faktor, kritischer Kompressibilitätskoeffizient, Molekulargewicht, isobare Wärmekapazität, Korrekturterm-Schalter
Ausgabe:	Thermische Leitfähigkeit der Mischung bzw. reinen Substanz

(**Fette Schrift** = optionale Argumente)

Das File *TCELYHANLEY* ermöglicht die Berechnung thermischer Leitfähigkeiten unter Verwendung oder unter Vernachlässigung des von Ely und Hanley vorgeschlagenen Korrekturterms [39]. Um den Term anzuwenden, muß der Benutzer am Schluß der Argumenteliste einen Schalter eingeben, in dessen Abhängigkeit der Korrekturterm verwendet wird.

In der Literatur [1] findet sich für gasförmiges Propylen bei 200°C und 150 bar ein Wert von $6.64 \cdot 10^{-2} \text{ W/(m} \cdot \text{K)}$. Die Funktion *TCELYHANLEY* berechnet einen um 3 % zu niedrigen Wert von $6.45 \cdot 10^{-2} \text{ W/(m} \cdot \text{K)}$.

7.1.2 Methode nach Chung

Eine weitere Abschätzungsmethode ist das Verfahren nach Chung [1]. Es ermöglicht - analog der Ely-Hanley Methode - die Abschätzung von thermischen Leitfähigkeiten über die Methode der übereinstimmenden Zustände. Jedoch werden für dieses Formelsystem weitere Parameter benötigt. Anhang B enthält weitere Einzelheiten bezüglich der Methode nach Chung.

Tab 48: I/O-Charakteristik der Funktion *TCCHUNG*

<i>TCCHUNG</i> :	Abschätzung thermischer Leitfähigkeiten mittels der Methode nach Chung
Eingabe:	Temperatur, Molenbrüche, Molekulargewicht, kritische Temperatur, isobare Wärmekapazität, azentrischer Faktor, Dipolmoment, Molvolumen, Kollisionsintegral der kinetischen Gastheorie, zwei binäre Interaktionsparameter, Korrekturterm für polare Substanzen
Ausgabe:	Thermische Leitfähigkeit der Mischung bzw. reinen Substanz

(**Fette Schrift** = optionale Argumente)

Sowohl bei der Ely-Hanley Methode als auch bei dem Verfahren nach Chung wird zunächst die Viskosität des betrachteten Systems bestimmt. Für dieser Abschätzung werden bei der Methode nach Chung binäre Interaktionsparameter, das molare Volumen des Systems und für stark polare Substanzen wie Alkohole eine substanzspezifischer Korrekturterm benötigt.

Die thermische Leitfähigkeit von gasförmigem Isopentan bei einem bar und 100°C ist in der Literatur [1] mit $2.2 * 10^{-2} \text{ W}/(\text{m} * \text{K})$ angegeben. Unter Anwendung der Funktion *TCCHUNG* berechnet sich ein Wert von $2.31 * 10^{-2} \text{ W}/(\text{m} * \text{K})$. Das entspricht einem Fehler von 5 Prozent.

Die Abschätzungsverfahren nach Chung und nach Ely-Hanley sind bei der Anwendung auf unpolare Gase mit durchschnittlichen Fehlern von 5 bis 7 % recht zuverlässig [1]. Es zeigt sich in diesem Zusammenhang keine Überlegenheit einer der beiden Methoden.

7.1.3 Abschätzung der thermischen Leitfähigkeit von Flüssigkeiten

Für die Abschätzung thermischer Leitfähigkeiten reiner Flüssigkeiten werden eine Vielzahl von Verfahren diskutiert. Latini schlägt eine Methode zur Abschätzung von λ unterhalb des normalen Siedepunktes vor:

$$\lambda_L = \frac{A \cdot (1 - T_r)^{0.38}}{T_r^{1/6}}. \quad [72]$$

T_r ist hier die reduzierte Temperatur $T_r = T/T_c$ und A ist ein substanzspezifischer Parameter. Diese Methode ist in der Funktion *TCLATINI* implementiert.

Tab 49: I/O-Charakteristik der Funktion *TCLATINI*

<i>TCLATINI</i> :	Abschätzung thermischer Leitfähigkeiten reiner Flüssigkeiten mittels der Methode nach Latini
Eingabe:	Temperatur, kritische Temperatur, normale Siedetemperatur, Molekulargewicht, Abschätzungsparameter A , alpha , beta , gamma
Ausgabe:	Thermische Leitfähigkeit von reinen Flüssigkeiten

(**Fette Schrift** = optionale Argumente)

Für den charakteristische Parameter A wird eine Näherung angegeben:

$$A = \frac{A^* \cdot T_b^\alpha}{M^\beta \cdot T_c^\gamma}. \quad [73]$$

Bei diesem Vorgehen werden die betrachteten Substanzen in Familien eingeteilt, für die dann die Parameter A^* , α , β und γ angegeben sind. Dabei ergeben sich jedoch Ungenauigkeiten, da viele Substanzen nicht behandelt werden oder nicht eindeutig einer Familie zugeordnet werden können. So kann zum Beispiel m-Cresol als Aromat oder Alkohol betrachtet werden. Die Abschätzung der thermischen Leitfähigkeit ergibt in beiden Fällen signifikante Fehler. Jedoch kann der Parameter für viele Substanzen auch direkt aus der Literatur entnommen werden [57].

Ein weiteres Abschätzungsverfahren wird von Sato und Riedel vorgeschlagen [1]. Für Berechnungen am normalen Siedepunkt T_b gilt laut Sato

$$\lambda(T_b) = \frac{1.11}{\sqrt{M}}, \quad [74]$$

wobei M das Molekulargewicht in g/mol wiedergibt. Um die thermische Leitfähigkeit auch bei anderen Temperaturen abschätzen zu können, schlägt Riedel eine weitere Gleichung vor. In Kombination mit Gleichung [74] ergibt sich die in der Funktion *TCSATORIEDEL* implementierte Gleichung:

$$\lambda_L = \frac{(1.11 / M^{1/2}) \cdot [3 + 20 \cdot (1 - T_r)^{2/3}]}{3 + 20 \cdot (1 - T_{br})^{2/3}}. \quad [75]$$

Tab 50: I/O-Charakteristik der Funktion *TCSATORIEDEL*

<i>TCSATORIEDEL</i> :	Abschätzung thermischer Leitfähigkeiten reiner Flüssigkeiten mittels der Methode nach Sato-Riedel
Eingabe:	Temperatur, kritische Temperatur, normale Siedetemperatur, Molekulargewicht
Ausgabe:	Thermische Leitfähigkeit reiner Flüssigkeiten

Die Sato-Riedel Methode zeigt große Abweichungen für Kohlenwasserstoffe mit geringer Molmasse und für verzweigte Kohlenwasserstoffe. Dabei liegen die Schätzungen höher als die experimentellen Werte. Bessere Ergebnisse werden bei Nicht-Kohlenwasserstoffen erhalten [1].

Auch für die Abschätzung der thermischen Leitfähigkeit von Flüssigkeitsmischungen werden Gleichungen vorgeschlagen. Für diese sind neben anderen Parametern zumeist auch die Leitfähigkeiten der reinen Substanzen erforderlich. Li entwickelte eine Gleichung,

in die neben den λ_i noch die molaren Volumen aller Komponenten einfließen:

$$\lambda_m = \sum_{i=1}^n \sum_{j=1}^n A_i \cdot A_j \cdot \lambda_{ij}, \quad [76]$$

$$\lambda_{ij} = 2 \cdot (\lambda_i^{-1} + \lambda_j^{-1})^{-1}, \quad [77]$$

$$A_i = \frac{x_i \cdot V_i}{\sum_{j=1}^n x_j \cdot V_j}. \quad [78]$$

Dieses Abschätzungsverfahren wird mit dem File *TCLI* dem Benutzer als Funktion zur Verfügung gestellt.

Tab 51: I/O-Charakteristik der Funktion *TCLI*

<i>TCLI</i> :	Abschätzung thermischer Leitfähigkeiten von Flüssigkeitsmischungen mittels der Methode nach Li
Eingabe:	Vektor Molenbrüche, Vektor der molaren Volumen, Vektor der thermischen Leitfähigkeiten der reinen Stoffe
Ausgabe:	Thermische Leitfähigkeit von Flüssigkeitsmischungen

Werden nicht wäßrige Flüssigkeitssysteme betrachtet, so können die molaren Volumen V_i auch durch die kritischen Volumen V_{ci} ersetzt werden.

Eine weitere Methode ist das Verfahren nach Rowley [1, 54]. Dabei kommt ein Modell lokaler Zusammensetzungen entsprechend dem NRTL-Modell zum Einsatz. Das Verfahren verwendet die durch die Literatur zugänglichen NRTL-Parameter g_{ij} um eine thermische Exzeß-Leitfähigkeit zu bestimmen. Durch Kombination mit den thermischen Leitfähigkeiten der reinen flüssigen Komponenten erhält Rowley folgende Zusammenhänge:

$$\lambda_m = \sum_{i=1}^N w_i \cdot \lambda_i + \sum_{i=1}^N w_i \cdot \frac{\sum_{j=1}^N w_j \cdot \lambda_{ji} \cdot (\lambda_{ji} - \lambda_i)}{\sum_{k=1}^N w_k \cdot \lambda_{ki}}, \quad [79]$$

$$\lambda_{ij} = \lambda_{ji} = \frac{w_{ij}^* \cdot w_{ij}^* \cdot \lambda_i + w_{ji}^* \cdot w_{ji}^* \cdot \lambda_j}{w_{ij}^* \cdot w_{ij}^* + w_{ji}^* \cdot w_{ji}^*}, \quad [80]$$

$$w_{ij}^* = \frac{w_{ij}^*}{\sum_{k=1}^N w_{ki}^* \cdot g_{ki}}, \quad [81]$$

$$w_{ij}^* = \frac{M_i \cdot \sqrt{g_{ji}}}{M_i \cdot \sqrt{g_{ji}} + M_j \cdot \sqrt{g_{ij}}}. \quad [82]$$

Um eine höhere Symmetrie der thermischen Exzeß-Leitfähigkeit zu erreichen, verwendet Rowley hier Massenanteile w_i an Stelle von Molanteilen. In die Funktion *TCROWLEY* muß der Benutzer aus Gründen der Konsistenz dennoch den Vektor aller Molenbrüche x_i eingeben. Dieser wird intern in Massenanteile umgerechnet.

Tab 52: I/O-Charakteristik der Funktion *TCROWLEY*

<i>TCROWLEY</i> :	Abschätzung thermischer Leitfähigkeiten von Flüssigkeitsmischungen mittels der Methode nach Rowley
Eingabe:	Vektor der Molenbrüche, Vektor der NRTL-Parameter, Vektor der Molmassen, Vektor der thermischen Leitfähigkeiten der reinen Stoffe
Ausgabe:	Thermische Leitfähigkeit von Flüssigkeitsmischungen

Als Anwendungsbeispiel soll die thermische Leitfähigkeit einer flüssigen Mischung von 48.7 % Aceton und 51.3 % Chloroform bei einer Temperatur von 25°C betrachtet werden [1]. Dazu werden die Ergebnisse der Abschätzungen nach Li und nach Rowley experimentellen Werten gegenübergestellt.

	Literaturwert	nach Li		nach Rowley	
		λ_m	Fehler	λ_m	Fehler
λ_m [W/(m*K)]	0.143	0.1379	3.6 %	0.1400	2.1 %

Neben den implementierten Methoden zur Berechnung der thermischen Leitfähigkeit von Flüssigkeitsmischungen werden noch weitere Verfahren vorgeschlagen. Die Methoden nach Li und nach Rowley werden programmtechnisch realisiert, da sie auch flüssige Gemische mit mehr als zwei Komponenten behandeln können. Beide Methoden schätzen die thermische Leitfähigkeit des Flüssigkeitsgemisches durch eine Interpolation der λ_i der reinen Komponenten ab.

Reid, Prausnitz und Poling [1] empfehlen zur Abschätzung der thermischen Leitfähigkeit organischer Flüssigkeiten unter ihrem normalen Siedepunkt die Methoden nach Li und nach Sato-Riedel. Die Berechnung der Leitfähigkeit bei reduzierten Temperaturen oberhalb von $T_r > 0.65$ ist aufgrund der geringen Datendichte in diesem Bereich nicht empfohlen. Ist die betrachtete Substanz unpolar, so können hier die Methoden zur Abschätzung der thermischen Leitfähigkeit von Gasen nach Chung und nach Ely-Hanley angewendet werden. Keines der implementierten Verfahren gibt den starken Anstieg von λ_i im Bereich des kritischen Punktes richtig wieder.

7.2 Viskosität

In diesem Kapitel werden Funktionen behandelt, die eine Abschätzung der Viskosität von Gasen und Flüssigkeiten ermöglichen. Aus der kinetischen Gastheorie resultiert ein Zusammenhang zwischen

der Viskosität, der Temperatur, dem Druck und der Molekülgröße [1]. Legt man für ein Gas das Modell sich nicht beeinflussender Kugeln zugrunde, so ergibt sich folgende Beziehung für die Viskosität:

$$\eta = 26.69 \cdot \frac{(M \cdot T)^{1/2}}{\sigma^2}, \quad [83]$$

wobei η die Viskosität in μP , M die molare Masse in g/mol , T die Temperatur in K und σ der gedachte Kugeldurchmesser in \AA ist.

7.2.1 Methode nach Lucas

Lucas [1] wendet eine Methode der übereinstimmenden Zustände an. Ausgehend von Gleichung [83] kann eine dimensionslose Viskosität η_r erhalten werden, wenn σ^3 mit dem kritischen Volumen gleichgesetzt wird. Darüber ist dann mit

$$V_c = \frac{R \cdot T_c}{P_c} \quad [84]$$

folgender Ausdruck zugänglich:

$$\eta_r = \xi \cdot \eta = f(T_r), \quad [85]$$

$$\xi = \left[\frac{R \cdot T \cdot N_0^2}{M^3 \cdot P_c^4} \right]^{1/6}. \quad [86]$$

Es sind hier N_0 die Avogadrosche Konstante in kmol^{-1} und M die Molmasse in kg/kmol .

Somit ergibt sich aus Gleichung [86] der folgende Ausdruck:

$$\xi = 0.176 \cdot \left(\frac{T_c}{M^3 \cdot P_c^4} \right)^{1/6}.$$

[87]

Lucas gibt für die in Gleichung [85] eingeführte Funktion der reduzierten Temperatur einen Ausdruck an. Dieses sehr umfangreiche Formelsystem wird in Anhang C ausführlich erläutert.

Tab 53: I/O-Charakteristik der Funktion *VILUCAS*

<i>VILUCAS</i> :	Abschätzung der Viskosität von Gasen mittels der Methode nach Lucas
Eingabe:	Molenbrüche, Temperatur, Druck, kritische Temperatur, kritischer Druck, Dampfdruck der reinen Komponenten, kritisches Volumen, kritischer Kompressibilitätsfaktor, Molmasse, Diplomoment, Schalterargument bezogen auf He, H ₂ , D ₂
Ausgabe:	Viskosität von Gasen

Die Funktion *VILUCAS* ermöglicht die Abschätzung der Viskosität von reinen Gasen und Gasgemischen im Bereich niedriger und hoher Drücke.

Mit der Funktion *VILUCAS* berechnet sich die Viskosität von gasförmigem Methanol bei 550 K und 1 bar zu 180.5 µP. Der experimentelle Wert liegt bei 181 µP [72].

7.2.2 Methode nach Chung

Das Abschätzungsverfahren nach Chung [1, 53] basiert auf der Theorie von Chapman und Enskog. Dabei wird die Wechselwirkung zwischen kollidierenden Molekülen durch eine intermolekulare Potentialfunktion berücksichtigt. Dieser Ansatz wurde in der Funktion *VICHUNG* implementiert. Für die Berücksichtigung der intermolekularen Wechselwirkungen wird die Lennard-Jones 12-6-Funktion verwendet.

Tab 54: I/O-Charakteristik der Funktion *VICHUNG*

<i>VICHUNG</i> :	Abschätzung der Viskosität von Gasen mittels der Methode nach Chung
Eingabe:	Molenbruch, Temperatur, Druck, molares Volumen, kritische Temperatur, kritisches Volumen, Molenbruch, kritischer azentrischer Faktor, Dipolmoment, Assoziationsparameter, binäre Interaktionsparameter, Parameter zur Berechnung des Kollisionsintegrals
Ausgabe:	Viskosität von Gasen

(**Fette Schrift** = optionale Argumente)

Unter Anwendung des Lennard-Jones-Potentials geben Neufeld, Janzen und Aziz eine Funktion zur Berechnung des Kollisionsintegrals an [1, 58].

$$\Omega = A \cdot T^{*-B} + C \cdot \exp(-D \cdot T^*) + E \cdot \exp(-F \cdot T^*) \quad [88]$$

Hier sind die Buchstaben A bis F empirisch bestimmte Konstanten, und T^* ist eine dimensionslose Temperatur, die sich gemäß

$$T^* = \frac{k \cdot T}{\epsilon} \quad [89]$$

berechnet. Weiterhin ist k die Boltzmann Konstante. Für einige Substanzen enthält das Datenfeld *Epsilon* des Datensatzes *data* das Verhältnis ϵ/k .

Die Funktion *VICHUNG* ermöglicht die Abschätzung der Viskosität von reinen Gasen und Gasmischungen bei unterschiedlichen Drücken. Zur Berechnung von Ω mit Gleichung [88] können eigene Parameter eingegeben oder die von Neufeld angegebenen Werte verwendet werden. Der Benutzer kann das Kollisionsintegral Ω mit der Funktion *LENJOPOT* direkt berechnen.

Tab 55: I/O-Charakteristik der Funktion *LENJOPOT*

<i>LENJOPOT</i> :	Berechnung des Kollisionsintegrals, der charakteristischen Energie und der charakteristischen Länge intermolekularer Wechselwirkung nach der Lennard-Jones 12-6-Potentialfunktion
Eingabe:	Temperatur, Vektor der charakteristischen Energien, Vektor der charakteristischen Längen, Parameter zur Berechnung des Kollisionsintegrals
Ausgabe:	Kollisionsintegral, charakteristische Energie und charakteristische Länge

(**Fette Schrift** = optionale Argumente)

Gleichung [88] kommt auch in der Funktion *LENJOPOT* zur Anwendung. Als Eingangsdaten benötigt die Funktion neben der Temperatur noch die substanzspezifischen Größen ϵ/k und σ .

Betrachtet man die intermolekulare Wechselwirkung zweier Teilchen mit ihrem Abstand (s. Abb. 11), so findet sich eine Entfernung, bei der die potentielle Wechselwirkungsenergie ein Minimum annimmt. Dies ist die charakteristische Energie ϵ . Der Kollisionsdurchmesser σ ist definiert als der Abstand zwischen beiden Teilchen, wenn die potentielle Energie $\psi(r)$ gleich Null wird.

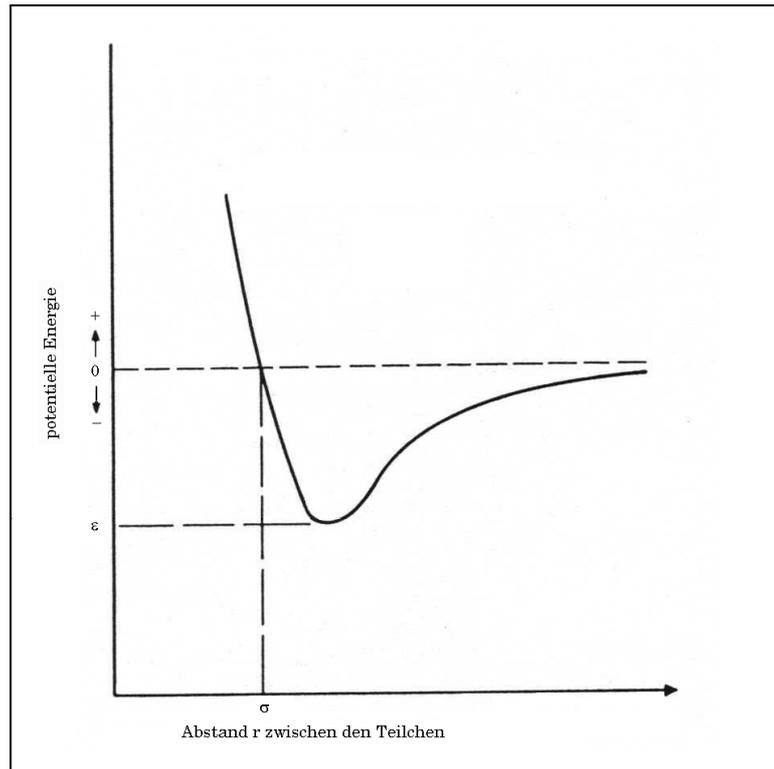


Abb. 11: Intermolekulare Potentialkurve [1]

Neben dem Kollisionsintegral Ω_D berechnet *LENJOPOT* aus den einzelnen Substanzgrößen σ_A und σ_B bzw. ε_A/k und ε_B/k die Interaktionsgrößen σ_{AB} und ε_{AB}/k . Dabei kommt ein lineares und ein quadratisches Mittel zur Anwendung. Neben dem Verhältnis ε/k ist auch der Kollisionsdurchmesser σ in dem Datenfeld *Sigma* des Datensatzes *data* enthalten.

Die Viskosität von Ammoniak bei 420 K und 300 bar wird in der Literatur [72] mit 571 μP angegeben. Unter Anwendung der Funktion *VILUCAS* berechnet sich eine Gasviskosität von 600 μP , die somit einen Fehler von 4.8 % aufweist.

Die Abweichungen der Schätzwerte der Funktionen *VICHUNG* und *VILUCAS* liegen durchschnittlich bei 3 %. Höhere Ungenauigkeiten ergeben sich, wenn Gase nahe des kritischen Punktes betrachtet werden. Bei der Berechnung der Viskosität von Gasmischungen unter hohem Druck nach Chung sind Abweichungen von bis zu 9 Prozent gefunden worden [1].

7.2.3 Abschätzung von Flüssigkeitsviskositäten

Basiert die Abschätzung der Viskosität von Gasen grundsätzlich auf einer Erweiterung der kinetischen Gastheorie, so findet sich für die Behandlung von Flüssigkeiten keine entsprechende theoretische Basis.

Reid, Prausnitz und Poling [1] geben eine von Van Velzen, Yaws und Duhne veröffentlichte Gleichungen zur Behandlung von η_{liqu} an. Sie ermöglichen die Abschätzung der Viskosität von Flüssigkeiten in Abhängigkeit der Temperatur:

$$\eta_{\text{liqu}} = A \cdot T^B, \quad [90]$$

$$\eta_{\text{liqu}} = A + \frac{B}{T}, \quad [91]$$

$$\eta_{\text{liqu}} = A + \frac{B}{T} + C \cdot T + D \cdot T^2. \quad [92]$$

Neben den Gleichungen [90] bis [92] sind auch die Konstanten A bis D in der Zusammenstellung [1] enthalten und in den Datenfeldern *VIS_A*, *VIS_B*, *VIS_C* und *VIS_D* des Datensatzes *data* abgelegt (s. Tab. 57). Weiterhin ist ein Temperaturbereich angegeben, in dem die Gleichungen mit den tabellierten Konstanten einsetzbar sind. Die obere und untere Gültigkeitsgrenze ist in den Datenfeldern *VIS-Ta* und *VIS-Te* enthalten.

Basierend auf diesen Daten wurde das File *VISLIQUID* erstellt. Es ermöglicht eine Abschätzung der Viskosität von reinen, flüssigen Substanzen.

Tab 56: I/O-Charakteristik der Funktion *VISLIQUID*

<i>VISLIQUID</i> :	Abschätzung der Viskosität von reinen Flüssigkeiten nach Daten von Van Velzen, Yaws und Duhne
Eingabe:	Temperatur, Gleichungsschalter, Parameter A bis D
Ausgabe:	Flüssigkeitsviskosität

Um zwischen den genannten Gleichungen zu unterscheiden, enthält *data* ein Datenfeld, dessen Inhalt eine Zuordnung der Konstanten zur verwendeten Gleichung ermöglicht.

Tab 57: Auf die Abschätzung der Viskosität bezogene Datenfelder in *data*

Datenfeld	Inhalt des Datenfeldes
<i>VIS_Eq</i>	Zahl von 1 bis 3 zur Zuordnung der Konstanten zu einer Gleichung
<i>VIS_A</i>	Konstante A für Gleichung [90] bis [92]
<i>VIS_B</i>	Konstante B für Gleichung [90] bis [92]
<i>VIS_C</i>	Konstante C für Gleichung [90] bis [92]
<i>VIS_D</i>	Konstante D für Gleichung [90] bis [92]
<i>VIS-Ta</i>	obere Temperaturgrenze
<i>VIS-Te</i>	untere Temperaturgrenze
<i>VISCOSITY</i>	Viskosität in cP
<i>VIS-T</i>	zu <i>VISCOSITY</i> gehörende Temperatur in K

Sind diese Verfahren zur Behandlung der Viskosität von reinen, flüssigen Substanzen stoffspezifische Näherungen, so stellen die Methoden zur Abschätzung von Flüssigkeitsmischungen zumeist Interpolationsmethoden dar.

Die zur Betrachtung von Flüssigkeitsmischungen implementierte Funktion *VITEJARICE* basiert auf einer Methode der übereinstimmenden Zustände. Nach Teja und Rice [59, 60, 61] gilt:

$$\ln(\eta_m \cdot \varepsilon_m) = \ln(\eta \cdot \varepsilon)_{r1} + [\ln(\eta \cdot \varepsilon)_{r2} - \ln(\eta \cdot \varepsilon)_{r1}] \cdot \frac{\omega_m - \omega_{r1}}{\omega_{r2} - \omega_{r1}}, \quad [93]$$

wobei sich die $r1$ und $r2$ auf zwei Referenzfluide beziehen. Dies können sowohl Bestandteile der Mischung als auch Fremdstoffe sein. η ist hier die Viskosität in cP und ω der azentrische Faktor. ε stellt einen Parameter dar, der sich gemäß folgender Gleichung berechnet:

$$\varepsilon = \frac{\sqrt[3]{V_c^2}}{\sqrt{T_c \cdot M}}. \quad [94]$$

Teja und Rice schlagen zur Bestimmung der Mischungsgrößen folgendes Vorgehen vor:

$$V_{c_m} = \sum_{i=1}^N \sum_{j=1}^N x_i \cdot x_j \cdot V_{c_{ij}}, \quad [95]$$

$$T_{c_m} = \frac{\sum_{i=1}^N \sum_{j=1}^N x_i \cdot x_j \cdot T_{c_{ij}} \cdot V_{c_{ij}}}{V_{c_m}}, \quad [96]$$

$$M_m = \sum_{i=1}^N x_i \cdot M_i, \quad [97]$$

$$\omega_m = \sum_{i=1}^N x_i \cdot \omega_i, \quad [98]$$

$$V_{c_{ij}} = \frac{\left(\sqrt[3]{V_{c_i}} + \sqrt[3]{V_{c_j}}\right)^3}{8}, \quad [99]$$

$$T_{c_{ij}} \cdot V_{c_{ij}} = \psi_{ij} \cdot \sqrt{T_{c_j} \cdot V_{c_j} \cdot T_{c_i} \cdot V_{c_i}}. \quad [100]$$

Der in Gleichung [100] auftretende Wechselwirkungsparameter ψ_{ij} muß experimentell ermittelt werden. Die Viskosität der Referenzfluide η_{r1} und η_{r2} in Gleichung [93] müssen bei Temperaturen bestimmt werden, die von der Systemtemperatur abweichen. Die Referenztemperaturen berechnen sich gemäß Gleichung [101]:

$$T_{r1} = \frac{T}{T_{c,r1}/T_{c_m}}, \quad T_{r2} = \frac{T}{T_{c,r2}/T_{c_m}}. \quad [101]$$

Die Funktion *VITEJARICE* ermöglicht neben der Berechnung der Viskosität der flüssigen Mischung auch die Bestimmung der Referenztemperaturen T_{r1} und T_{r2} .

Tab 58: I/O-Charakteristik der Funktion *VITEJARICE*

<i>VITEJARICE</i> :	Abschätzung der Viskosität von Flüssigkeitsmischungen nach der Methode nach Teja und Rice
Eingabe:	Vektor der Molenbrüche, Temperatur, Vektor der kritischen Temperatur, Vektor der kritischen Volumen, Interaktionsparameter, Vektor der Reinstoff Viskositäten, Vektor der Molmassen, Vektor der Dipolmomente
Ausgabe:	Viskosität flüssiger Mischungen

(**Fette Schrift** = optionale Argumente)

Um T_{r1} und T_{r2} zu bestimmen, müssen die Viskositäten der Referenzfluide, die Molmassen und die Dipolmomente der Mischungsteilnehmer beim Aufruf von *VITEJARICE* unberücksichtigt bleiben.

Im folgenden Beispiel wird zunächst die Viskosität einer Mischung aus Wasser und 1,4-Dioxan bei 60°C abgeschätzt. Die Namen der verwendeten Konstanten sind wiederum willkürlich gewählt und für die Berechnung nicht ausschlaggebend.

```
» load data
» T=333.14;psi=[1,1.37;1.37,1];
» Tc=[data(val).Tc];Vc=[data(val).Vc];
» M=[data(val).MolWt]; Omega=[data(val).Omega];
» x=[0.83,0.17];ni=[0.712,1.63];
» VITEJARICE(x,T,Tc,Vc,psi,ni,M,Omega)
```

ans =

0.8931

Der Literaturwert der Viskosität [1] wird mit 0.89 cP angegeben. Durch einen zweiten Aufruf werden dann die Temperaturen der Referenzsubstanzen T_{r1} und T_{r2} berechnet.

```
» VITEJARICE(x,T,Tc,Vc,psi)
```

ans =

308.6929 279.9362

Die Methode nach Teja und Rice zeigt gute Ergebnisse. Bei unpolaren Systemen ergeben sich mittlere Abweichungen von 1 %. Mischungen unpolarer und polarer Fluide und rein polare Mischungen zeigen Abweichungen von durchschnittlich 2.5 %. Wäßrige Systeme ergeben mittlere Fehler von 9 % [1].

7.3 Diffusionskoeffizienten

Die in diesem Kapitel betrachteten Diffusionskoeffizienten beziehen sich ausschließlich auf die durch Konzentrationsgradienten induzierte, isotherme und isobare Diffusionen. Zur Bestimmung der Diffusionskoeffizienten wird ein Bilanzrahmen mit einem Gesamtstrom von Null betrachtet. Damit ergeben sich für eine binäre Mischung aus A und B folgende Zusammenhänge zwischen dem molaren Fluß und dem Konzentrationsgradienten:

$$J_A^M = -c \cdot D_{AB} \cdot \frac{dx_A}{dz}, \quad [102]$$

$$J_B^M = -c \cdot D_{BA} \cdot \frac{dx_B}{dz}. \quad [103]$$

Dabei ist c die molare Gesamtkonzentration der Mischung und D_{AB} bzw. D_{BA} sind die wechselseitigen Diffusionskoeffizienten.

7.3.1 Methode nach Chapman und Enskog

Durch Lösung der Boltzmann Gleichung entwickelten Chapman und Enskog unabhängig voneinander eine Gleichung, die die Abschätzung binärer Diffusionskoeffizienten von Gasen unter geringen Drücken ermöglicht:

$$D_{AB} = \frac{3}{16} \cdot \frac{\sqrt{4 \cdot \pi \cdot k \cdot T / M_{AB}}}{n \cdot \pi \cdot \sigma_{AB}^2 \cdot \Omega} \cdot f_D. \quad [104]$$

In Gleichung [104] ist M_{AB} gegeben durch den Zusammenhang [105]:

$$M_{AB} = \frac{2}{\frac{1}{M_A} + \frac{1}{M_B}}. \quad [105]$$

Hier sind M_A und M_B die molaren Massen, n die Teilchenzahl pro Volumeneinheit der Mischung, k die Boltzmann-Konstante und T die Temperatur in K. Das Kollisionsintegral Ω ist eine Funktion der Temperatur und hängt, ebenso wie die charakteristische Länge σ_{AB} , von der Wahl der intermolekularen Potentialfunktion ab. Der Korrekturterm f_D ist zumeist gleich eins. f_D kann geringfügig von eins abweichen, wenn die molaren Massen M_A und M_B stark unterschiedlich sind und die leichtere Komponente nur in sehr geringen Konzentrationen vorhanden ist. Der Wert für f_D liegt dann zwischen 1.0 und 1.1 [1].

Die Funktion *DCHAPENS* greift diese Formeln auf und ermöglicht eine Abschätzung von D_{AB} .

Tab 59: I/O-Charakteristik der Funktion *DCHAPENS*

<i>DCHAPENS</i> :	Abschätzung binärer Gas Diffusionskoeffizienten bei geringen Drücken nach den Gleichungen von Chapman und Enskog
Eingabe:	Temperatur, Vektor der Molmassen, Vektor der charakteristischen Länge, Kollisionsintegral, bifunktionale Argumente A und B
Ausgabe:	binäre Diffusionskoeffizienten

(**Fette Schrift** = optionales Argument)

Wird n durch das ideale Gasgesetz ausgedrückt und f_D gleich eins angenommen, so ergibt sich folgender Ausdruck:

$$D_{AB} = \frac{0.00266 \cdot T^{3/2}}{P \cdot \sqrt{M_{AB}} \cdot \sigma_{AB}^2 \cdot \Omega} . \quad [106]$$

Die Funktion *DCHAPENS* ermöglicht es für die Berechnung von D_{AB} sowohl Gleichung [104] als auch Gleichung [106] zu verwenden. Wird nur eines der bifunktionalen Argumente A und B (s. Tab. 59) eingegeben, so interpretiert die Funktion Argument A als den Druck P und verwendet zur Berechnung von D_{AB} Gleichung [106].

Werden der Funktion hingegen beide Argumente A und B Werte übergeben, so interpretiert *DCHAPENS* diese als n und f_D und verwendet zur Berechnung von D_{AB} Gleichung [104].

Die weiteren Eingangsargumente σ_{AB} und Ω der Funktion *DCHAPENS* können mit *LENJOPOT* (s. Kapitel 7.2.2) erhalten werden.

7.3.2 Methode nach Wilke und Lee

Neben der grundlegenden Theorie von Chapman und Enskog wurden weitere Methoden zur Abschätzung von Diffusionskoeffizienten binärer Gasgemische bei geringem Druck vorgeschlagen. In der Programmsammlung wurden nur zwei - in der Literatur als zuverlässig beurteilte - Methoden implementiert [1, 69, 70].

Die Funktion *DWILKELEE* realisiert die Methode nach Wilke - Lee [63]. Basierend auf Gleichung [106] geben die Autoren folgende Gleichung an:

$$D_{AB} = \frac{\left[3.03 - \frac{0.98}{\sqrt{M_{AB}}} \right] \cdot 10^{-3} \cdot T^{3/2}}{P \cdot \sqrt{M_{AB}} \cdot \sigma_{AB}^2 \cdot \Omega} . \quad [107]$$

Dabei werden σ_A und σ_B aus den molaren Flüssigkeitsvolumen der betrachteten Substanzen ermittelt:

$$\sigma = 1.18 \cdot \sqrt[3]{V} . \quad [108]$$

Desweiteren berechnen Wilke und Lee $(\varepsilon/k)_A$ bzw. $(\varepsilon/k)_B$ über die normale Siedetemperatur der betrachteten Substanz:

$$\left(\frac{\varepsilon}{k} \right) = 1.15 \cdot T_b . \quad [109]$$

Tab 60: I/O-Charakteristik der Funktion *DWILKELEE*

<i>DWILKELEE</i> :	Abschätzung binärer Gasdiffusionskoeffizienten bei geringen Drücken nach dem Ansatz von Wilke und Lee
Eingabe:	Temperatur, Vektor der Molmassen, Druck, molares Volumen der Gasmischung, Vektor der Siedepunkte
Ausgabe:	binäre Diffusionskoeffizienten

Die binären Größen σ_{AB} und $(\varepsilon/k)_{AB}$ ergeben sich durch ein quadratisches und ein lineares Mittel:

$$\varepsilon_{AB} = \sqrt{\varepsilon_A \cdot \varepsilon_B}, \quad [110]$$

$$\sigma_{AB} = \frac{\sigma_A + \sigma_B}{2}. \quad [111]$$

Unter Anwendung der Funktion *DWILKELEE* berechnet sich der Diffusionskoeffizient von Allylchlorid, C_3H_5Cl , in Luft bei $25^\circ C$ und einem Druck von einem bar zu $0.099 \text{ cm}^2/\text{s}$. Von dem Literaturwert [70] von $0.098 \text{ cm}^2/\text{s}$ weicht der berechnete Wert somit 1 % ab.

7.3.3 Methode nach Fuller, Schettler und Giddings

Eine weitere Methode ist in der Funktion *DFULSCHEGID* implementiert. Ausgehend von Gleichung [106] geben Fuller, Schettler und Giddings eine Inkrementmethode an:

$$D_{AB} = \frac{0.00143 \cdot T^{1.75}}{P \cdot \sqrt{M_{AB}} \cdot \left[\sqrt[3]{\Sigma_{v,A}} + \sqrt[3]{\Sigma_{v,B}} \right]^2}. \quad [112]$$

Die Parameter Σ_v werden für jede Komponente durch Summation von sogenannten atomaren Diffusionsvolumen erhalten. Die Werte dieser elementspezifischen Inkremente sind in den Tabellen 61 und 62 enthalten und werden mit Gleichung [113] berechnet.

$$(\Sigma_v) = \sum_{i=1}^N (A_{hi} \cdot \Sigma_{v_i}) \quad h = 1, \dots, L \quad [113]$$

Hier wird die Notation entsprechend Gleichung [37] angewendet, das heißt A_{hi} ist der Koeffizient des Elementes h in der Summenformel der Komponente S_i .

Tab 61: Atomare Diffusionsvolumen nach Fuller, Schettler und Giddings

Element	Σ_v	Element	Σ_v
C	15.9	Br	21.9
H	2.31	I	29.8
O	6.11	S	22.9
N	4.54	aromatische Ringe	-18.3
F	14.7	heterocyclische Ringe	-18.3
Cl	21.0		

So ergibt sich zum Beispiel (Σ_v) für Allylchlorid, C_3H_5Cl , nach obiger Tabelle und Gleichung [113] zu

$$(\Sigma_v) = 3 \cdot C + 5 \cdot H + Cl = 3 \cdot 15.9 + 5 \cdot 2.31 + 21 = 80.25 .$$

Tab 62: I/O-Charakteristik der Funktion *DFULSCHEGID*

<i>DFULSCHEGID</i> :	Abschätzung binärer Diffusionskoeffizienten von Gas bei geringen Drücken nach den Gleichungen von Fuller, Schettler und Giddings
Eingabe:	Temperatur, Vektor der Molmassen, Druck, Summe der atomaren Diffusionsvolumen
Ausgabe:	binäre Diffusionskoeffizienten

Die Diffusionsvolumen einfacher Moleküle sind in Tabelle 63 enthalten.

Tab 63: Diffusionsvolumen einfacher Moleküle

Molekül	Σ_v	Molekül	Σ_v
He	2.67	CO	18.0
Ne	5.98	CO ₂	26.9
Ar	16.2	N ₂ O	35.9
Kr	24.5	NH ₃	20.7
Xe	23.7	H ₂ O	13.1
H ₂	6.12	SF ₆	71.3
D ₂	6.84	Cl ₂	38.4
N ₂	18.5	Br ₂	69.0
O ₂	16.3	SO ₂	41.8
Luft	19.7		

Als Anwendungsbeispiel soll hier der Diffusionskoeffizient von Allylchlorid in Luft bei 25°C und 1 bar berechnet werden. Der experimentelle Wert findet sich in der Literatur [70] mit 0.098 cm²/s. Mit einem berechneten Wert von 0.0992 cm²/s ergibt sich damit ein Abschätzungsfehler von ca. 1 %.

Die implementierten Methoden zur Abschätzung von binären Diffusionskoeffizienten von Gasen sind mit mittleren Fehlern um 7 % recht zuverlässig. Ausgehend von binären Diffusionskoeffizienten können mit verschiedenen Ansätzen Diffusionskoeffizienten für Mehrkomponenten Gasgemische erhalten werden [1]. Die Temperatur- und Druckabhängigkeit von Gasdiffusionskoeffizienten werden nicht berücksichtigt. Auch sind keine Abschätzungsverfahren für Diffusionskoeffizienten von Flüssigkeiten implementiert worden, da sie keine ausreichende Verlässlichkeit bieten.

Das Programmsystem MATLAB kann aufgrund seiner hoch flexiblen Programmiersprache als gut geeignet für die programmtechnische Realisierung der in diesem Kapitel angesprochenen Formelsysteme beurteilt werden. Die Implementierung der teilweise sehr umfang-

reichen Formeln (s. Anhang) ist mit MATLAB ohne Schwierigkeit möglich.

8 Diskussion

8.1 Diskussion der Programmsammlung

Die im Rahmen der Arbeit erstellte Programmsammlung ermöglicht nicht nur die Ermittlung thermodynamischer Basisdaten für reaktionstechnische Simulationen und den Auf- bzw. Ausbau einer breiten Datengrundlage für diese Anwendungen, sondern auch die Berechnung komplexer Simultangleichgewichte unterschiedlichen Realitätsgrades. Inhaltlich wird MATLAB in mehreren Bereichen erweitert durch

- ein System zur benutzerfreundlichen Verwaltung von Basisdaten unter MATLAB (vgl. Kap. 2),
- Methoden für thermodynamische Berechnungen, insbesondere durch Funktionen zur Lösung von Zustandsgleichungen (vgl. Kap. 3, 4),
- Verfahren für umfassende stöchiometrische Betrachtungen (vgl. Kap. 5.2),
- kompakte und robuste Methoden zur Berechnung chemischer Gleichgewichte unterschiedlicher Realitätsgrade (vgl. Kap. 5.3),
- Programme zur Berechnung von molekularen Fluid-Transportprozessen (vgl. Kap. 7) und
- Funktionen zur Berechnung von Dampf-Flüssigkeits-Gleichgewichten über g^E -Methoden (vgl. Kap. 6).

Diese inhaltlichen Ergänzungen werden zum einen durch die konsequente Anwendung von MATLAB Funktionen realisiert, zum anderen sind eine Vielzahl von Algorithmen und verschiedene numerische Verfahren programmiert bzw. modifiziert worden. Im Zuge dieser Ergänzungen wurde die Leistungsfähigkeit von MATLAB in grundlegenden Bereichen erweitert durch

- die Erstellung einer neuen Suchfunktion für die Datenbasis *data* (s. Kap. 2.3),
- die Entwicklung und Anwendung einer neuen Strategie zum Lösen der BWRS-Zustandsgleichung (s. Kap. 4.1),

- die Programmierung einer alternativen Methode für die Berechnung von DAGI unter MATLAB 5.2 (s. Kap. 5.3.5) und
- die Ausnutzung der robusten DGI-Löser für die Berechnung chemischer Gleichgewichte (s. Kap. 5.3.4).

Bei der Entwicklung der Programmsammlung ist weitgehend ein Baukasten-Prinzip zur Anwendung gekommen (s. Kap. 3). Die realisierten Verfahren wurden nach Möglichkeit optional implementiert und können durch andere Methoden ersetzt, ergänzt und miteinander kombiniert werden. Ausnahmen bilden Funktionen, die zwingend von der Verwendung einer anderen Funktion abhängen (s. Kap. 4.1, 4.2). Die Realisierung dieses Baukasten-Systems wird durch das Funktionen-Konzept von MATLAB deutlich unterstützt. Weiterhin wurden bei der Implementierung von Methoden, die den Einsatz substanzspezifischer Konstanten oder binärer Interaktionsparameter voraussetzen, bewußt offene Strukturen realisiert (s. Kap. 3, 4).

Diesen Vorteilen steht jedoch gegenüber, daß die den Funktionen zu übergebenden Argumentelisten lang und unübersichtlich werden können. Unter Verwendung von *varargin* (s. Kap. 3.2) sind jedoch optionale Parameter überall dort ermöglicht worden, wo der Benutzer in den Programmablauf eingreifen kann, dies aber nicht tun muß. Die Funktionen bleiben somit für einen möglichen stärker differenzierten Einsatz flexibel. Der Benutzer muß jedoch nicht immer alle Argumente selbst belegen, sondern kann bei optionalen Argumenten voreingestellte Werte nutzen.

Da sich die erstellten Programme in das Funktionen-Konzept von MATLAB einfügen, setzt die Handhabung der Programmsammlung von seiten des Benutzers keine tiefgreifenden Kenntnisse im Umgang mit MATLAB voraus. Vielmehr erfordert der effektive Einsatz der Programmsammlung die Beurteilung der zur Verfügung gestellten Methoden und Verfahren. Somit sind fundierte Kenntnisse der betrachteten thermodynamischen Zusammenhänge unerlässlich.

Die Einsatzfähigkeit der realisierten Programme wird teilweise durch die Berechnung von Anwendungsbeispielen unter Beweis gestellt. Jedoch ist die Leistungsfähigkeit einiger der implementierten Methoden nicht abhängig von der Programmierung, sondern allein von dem zugrundeliegenden Modell bzw. den Eingangsdaten. Die Berechnung von chemischen Gleichgewichten und des molaren Volumens sind jedoch deutlich von der Implementierung und der Leistungsfähigkeit der verwendeten Verfahren abhängig.

Zusammenfassend läßt sich feststellen, daß sich die Programmsammlung durch ihre thematische Ausrichtung auf thermodynamische, reaktionstechnische Fragestellungen von dem gegenwärtigen MATLAB-Angebot abgrenzt. Zur Verwirklichung der Programmsammlung sind die von MATLAB zur Verfügung gestellten Programme gezielt genutzt, kombiniert und gegebenenfalls erweitert worden. Die in der Aufgabenstellung definierte Zielsetzung, die Schaffung einer Programmsammlung zur Berechnung thermodynamischer Basisdaten, wurde damit erfüllt. Darüber hinaus stellen die implementierten Verfahren für die Berechnung stöchiometrischer Zusammenhänge und chemischer Gleichgewichte eine sinnvolle Ergänzung zu der betrachteten Themenstellung dar.

8.2 Beurteilung der Eignung von MATLAB für die Implementierung der Programmsammlung

Die Zielsetzung der Schaffung einer Programmsammlung zur Berechnung thermodynamischer Daten für chemisch technische Simulationsrechnungen kann mit dem verwendeten Software-System MATLAB realisiert werden. Die von MATLAB zur Verfügung gestellten Funktionen und Datentypen sind weitreichend einsetzbar und leistungsfähig. Dennoch sind einige Problemstellungen nur indirekt, über Umwege oder mit neuen Lösungswegen umsetzbar.

Zur flexiblen und leistungsfähigen Realisierung einer Datenbasis für thermodynamische Berechnungen zeigt sich der Datentyp des

structure array als sehr gut geeignet (s. Kap.2.1). Die bei einem großen Datenbestand notwendige Funktion zur Suche von Substanzen muß jedoch mit der Funktion *DATASEARCH* neu erstellt werden, da die Funktion *FIND* Daten dieses Datentyps nicht verarbeiten kann (s. Kap. 2.4). Die in diesem Zusammenhang angewendete Technik zur Erstellung von graphischen Benutzerschnittstellen, kurz GUI's genannt, ist gewöhnungsbedürftig und recht aufwendig (s. Kap. 2.4).

Die unter anderem bei der Betrachtung von Zustandsgleichungen notwendige Lösung von nicht-linearen Gleichungen ist mit der für diesen Einsatz vorgesehenen Funktion *FSOLVE* nur in einfachen Fällen (s. Kap. 4.1) oder mit guten Startwerten möglich (s. Kap. 4.1 und 5.3.2). Sind kubische Zustandsgleichungen mit der Funktion *ROOTS* komfortabel lösbar, so ist dieser Ansatz bei der Betrachtung von komplexeren Zustandsgleichungen - wie der BWRS-Gleichung - nicht mehr anwendbar. Da MATLAB keine Funktion zur Verfügung stellt, die das Auffinden aller Lösungen der BWRS-Gleichung ermöglicht [34], wird unter Einsatz der Funktion *CONSTR* eine eigene Lösungsstrategie entwickelt und implementiert (s. Kap. 4.1).

Bei der Behandlung chemischer Gleichgewichte ergeben sich ähnliche Probleme (s. Kap. 5.3.2); zu deren Lösung wurden die von MATLAB zur Verfügung gestellten DGI-Löser herangezogen (s. Kap. 5.3.4). Diese numerischen Verfahren stellen einen wichtigen Bestandteil des Software-Systems MATLAB dar. Die DGI-Löser sind umfassend einsetzbar, leistungsfähig und robust.

Die Lösung von DAGls ist unter MATLAB 5.2 nur durch den Einsatz externer DAGl-Löser möglich (s. Kap. 4.3.5). Durch die Implementierung eines alternativen Lösungsverfahrens wird dieser Weg umgangen. In diesem Zusammenhang kommt die *Symbolic Math Toolbox* zur Anwendung. Dabei muß auf die Kumulation geringer numerischer Ungenauigkeiten bei der Evaluation umfangreicher symbolischer Formelsysteme hingewiesen werden. Dies führt beim Einsatz in einem ODE-File zu einer Segmentverletzung und zum

Absturz des DGI-Lösers. Die Weiterentwicklung der DGI-Löser unter MATLAB 5.3 macht dieses Vorgehen jedoch überflüssig, da nun auch DAGs bearbeitet werden können.

Sehr gut geeignet für die Implementierung der meisten thermodynamischen Berechnungen ist die von MATLAB angebotene Möglichkeit der Programmierung eigener Funktionen und M-Files. Dem Benutzer steht in einer leicht verständlichen Editorumgebung eine hochflexible und mathematisch-technisch orientierte Programmiersprache zur Verfügung (s. Kap. 1.1). Die Definition von Unterfunktionen und die Verwendung von *varargin* sind dafür nur zwei Beispiele (s. Kap. 3.1.1 und 3.1.2).

Zusammenfassend kann MATLAB als gut geeignet für die Berechnung thermodynamischer Zusammenhänge beurteilt werden (s. Kap. 3, 6 und 7). Diese besondere Eignung resultiert aus der Kombination einer hoch flexiblen mathematisch orientierten Programmiersprache mit sehr umfassenden Funktionsbibliotheken für technisch-wissenschaftliche Hochleistungsrechnungen. Dabei ist hervorzuheben, daß die *toolboxes* konzeptionell in das Softwaresystem integriert, damit schnell verfügbar und problemlos in bestehende Strukturen einzubeziehen sind.

Daneben muß jedoch auf Unzulänglichkeiten bei der Anwendung der *Symbolic Math Toolbox* und der Lösung nicht-linearer Gleichungen hingewiesen werden. Dabei wurde letztere mit einem alternativen Lösungsansatz umgangen. Dennoch ist eine Optimierung dieser Funktionalitäten wünschenswert. Auch die Handhabung des GUI-Entwicklungseditors *GUIDE* ist verbesserungswürdig.

Die diskutierten Vor- und Nachteile lassen den Schluß zu, daß MATLAB sich auf dem Gebiet der chemischen Technik als Werkzeug für Hochleistungsrechnungen etablieren kann.

9 Ausblick

Die Programmsammlung ermöglicht thermodynamische Berechnungen mit besonderem Augenmerk auf reaktionstechnische Fragestellungen. Unterstützt werden die implementierten Methoden durch eine "Datenbank", die Basisdaten für die Berechnungen verfügbar macht. Die Einsatzfähigkeit der Verfahren wird mit Beispielrechnungen dokumentiert, jedoch bleibt ein Test der Leistungsfähigkeit der implementierten Verfahren bei der Anwendung auf umfangreichere Beispiele aus. In dieser Hinsicht ist insbesondere das Verhalten der Methoden zur Bestimmung chemischer Gleichgewichte sehr interessant. Dabei ist eine eingehendere Betrachtung der G-Minimierung zur Berechnung chemischer Gleichgewichtszusammensetzungen im Vergleich zu den bereits implementierten Methoden von Interesse.

Inhaltlich ist die Programmsammlung um drei Punkte zu erweitern. Dies sind: die Betrachtung von Flüssig-Flüssig Gleichgewichten, die Berechnung chemischer Gleichgewichtszusammensetzungen mit gleichzeitiger Berücksichtigung von Phasengleichgewichten und Verfahren zur Anpassung von Meßdaten an die implementierten Methoden. Mit der Implementierung einer Meßdatenanpassung wird der Einsatz von Zustandsgleichungen und g^E -Methoden zur Betrachtung von Phasengleichgewichten deutlich flexibler.

Durch eine solche Erweiterung der Programmsammlung würde eine leistungsfähige Ergänzung zu MATLAB entstehen. Ob ein solches Programmpaket bereits als Toolbox zu bezeichnen wäre, ist nicht sicher. Die aus der inhaltlichen Abgrenzung resultierende Begrenzung auf bestimmte thermodynamische Betrachtungen innerhalb der Reaktionstechnik spricht dem entgegen. In Kombination mit anderen Programmen, zum Beispiel für kinetische Betrachtungen [71] und Methoden zur Modellbildung wäre es jedoch durchaus gerechtfertigt, von einer reaktionstechnischen *toolbox* zu sprechen.

10 Verzeichnis der wichtigsten Funktionen

Funktion	Bedeutung
<i>ABBWR</i>	Berechnung der Mischungs- oder Reinstoffkonstanten für die BWRS- und die BWR-Gleichungen
<i>ABBWRMIX</i>	Berechnung der Mischungskonstanten für die BWRS- und die BWR-Gleichungen
<i>ABMIX</i>	Berechnung von Mischungskonstanten für kubische Zustandsgleichungen
<i>ABPR</i>	Berechnung der Konstanten für kubische Zustandsgleichungen nach Peng, Robinson
<i>ABRK</i>	Berechnung der Konstanten für kubische Zustandsgleichungen nach Redlich-Kwong
<i>ABSRK</i>	Berechnung der Konstanten für kubische Zustandsgleichungen nach Soave-Redlich-Kwong
<i>ABVDW</i>	Berechnung der Konstanten für kubische Zustandsgleichungen nach Van der Waals
<i>CPIDEAL</i>	Berechnung der Wärmekapazität idealer Gase
<i>DATASEARCH</i>	Suche von Indexnummern im Datensatz anhand von Substanznamen bzw. Summenformeln
<i>DCHAPENS</i>	Abschätzung binärer Gasdiffusionskoeffizienten bei geringen Drücken nach den Gleichungen von Chapman und Enskog
<i>DEPARTCPBWR</i>	Berechnung der Exzeß-Wärmekapazität mittels der BWR-Gleichung
<i>DEPARTPCUB</i>	Berechnung der Exzeß-Wärmekapazität mittels kubischer Zustandsgleichungen mit reduzierter Menüsteuerung
<i>DEPARTPCUBIC</i>	Berechnung der Exzeß-Wärmekapazität mittels kubischer Zustandsgleichungen mit erweiterter Menüsteuerung
<i>DEPFUNCBWR</i>	Berechnung der Entropie- und Helmholtzenergie-Abweichungsfunktionen und des Kompressibilitätskoeffizienten z mittels der BWRS bzw. der BWR Zustandsgleichungen

<i>DEPFUNCCUB</i>	Berechnung der Entropie- und Helmholtzenergie-Abweichungsfunktionen und des Kompressibilitätskoeffizienten z mittels kubischer Zustandsgleichungen
<i>DFULSCHEGID</i>	Abschätzung binärer Gasdiffusionskoeffizienten bei geringen Drücken nach den Gleichungen von Fuller, Schettler und Giddings
<i>DTREAD</i>	Einlesen von IKC Files in ein MATLAB kompatibles Datenformat
<i>DWILKELEE</i>	Abschätzung binärer Gasdiffusionskoeffizienten bei geringen Drücken nach den Gleichungen von Wilke und Lee
<i>EE</i>	Berechnung chemischer Simultangleichgewichte durch Integration von DGI mit möglicher Berücksichtigung realer Einflüsse durch g^E -Methoden
<i>EEM</i>	Berechnung chemischer Simultangleichgewichte durch Integration von DAGI mit möglicher Berücksichtigung realer Einflüsse durch Fugazitätskoeffizienten
<i>EENL</i>	Berechnung chemischer Simultangleichgewichte durch Lösen nicht linearer Gleichungen
<i>ELSPEMA</i>	Erstellen der Element-Spezies-Matrix
<i>EMPTY</i>	Überprüfung ob ein Datenfeld einer fraglichen Substanzen Daten enthält
<i>EQU2STR</i>	Bestimmung der stöchiometrischen Koeffizienten und der Summenformeln aus Reaktionsgleichungen
<i>ESM2EQU</i>	Ausgabe von Schlüsselreaktionen in Stringformat, ausgehend von der Element-Spezies-Matrix
<i>FUGBWR</i>	Berechnung der Fugazitätskoeffizienten mittels der BWRS- bzw. der BWR-Gleichung
<i>FUGCUB</i>	Berechnung der Fugazitätskoeffizienten durch kubische Zustandsgleichungen mit reduzierter Menüsteuerung und Verarbeitung eigener kubischer Konstanten
<i>FUGCUBIC</i>	Berechnung der Fugazitätskoeffizienten durch kubische Zustandsgleichungen mit erweiterter Menüsteuerung

<i>GP</i>	Berechnung der freien Gibbsschen Exzeß-Enthalpie mittels kubischer Zustandsgleichungen mit erweiterter Menüsteuerung
<i>GPRES</i>	Berechnung der freien Gibbsschen Exzeß-Enthalpie mittels kubischer Zustandsgleichungen mit reduzierter Menüsteuerung
<i>GPRESBWR</i>	Berechnung der freien Gibbsschen Exzeß-Enthalpie mittels der BWRS- bzw. der BWR- Zustandsgleichungen
<i>GRX</i>	Berechnung der temperaturabhängigen freien Gibbsschen Reaktionsenthalpie
<i>GTEMP</i>	Berechnung der Änderung der freien Gibbsschen Enthalpie in Abhängigkeit von der Temperatur
<i>HP</i>	Berechnung der Exzeß-Enthalpie mittels kubischer Zustandsgleichungen mit erweiterter Menüsteuerung
<i>HPRES</i>	Berechnung der Exzeß -Enthalpie mittels kubischer Zustandsgleichungen mit reduzierter Menüsteuerung
<i>HPRESBWR</i>	Berechnung der Exzeß-Enthalpie mittels der BWRS- bzw. der BWR- Zustandsgleichungen
<i>HRX</i>	Berechnung der temperaturabhängigen Reaktionsenthalpie
<i>HTEMP</i>	Berechnung der Enthalpieänderung in Abhängigkeit von der Temperatur
<i>INMATRIX</i>	Gesteuerte Erstellung einer $n * n$ Matrix aus benutzerdefinierten Eingaben
<i>KEYESM</i>	Suche von Schlüsselkomponenten Kombinationen in der Element-Spezies-Matrix
<i>KEYREAC</i>	Bilden eines Satzes von Schlüsselreaktionen
<i>KT</i>	Berechnung der temperaturabhängigen Gleichgewichtskonstanten chemischer Reaktionen
<i>LENJOPOT</i>	Berechnung des Kollisionsintegrals, der charakteristischen Energie und der charakteristischen Länge intermolekularer Wechselwirkung nach der Lennard-Jones 12-6 Potentialfunktion

<i>MARGULES</i>	Berechnung von Fugazitätskoeffizienten nach Margules
<i>N2EQU</i>	Ausgabe von Schlüsselreaktionen in Stringformat, ausgehend von der stöchiometrischen Matrix
<i>N2RXGLMAT</i>	nach Vorzeichen sortierte Ausgabe von stöchiometrischen Koeffizienten einer stöchiometrischer Matrix N
<i>NRTL</i>	Berechnung von Aktivitätskoeffizienten nach der NRTL-Methode
<i>PVAP</i>	Dampfdruck-Berechnung
<i>RX2N</i>	Bestimmung von Reaktionsmatrizen und Indexvektoren aus Reaktionsgleichungen
<i>SP</i>	Berechnung der Exzeß-Entropie mittels kubischer Zustandsgleichungen mit erweiterter Menüsteuerung
<i>SPRES</i>	Berechnung der Exzeß-Entropie mittels kubischer Zustandsgleichungen mit reduzierter Menüsteuerung
<i>SPRESBWR</i>	Berechnung der Exzeß-Entropie mittels der BWRS- bzw. der BWR- Zustandsgleichungen
<i>SRX</i>	Berechnung der temperaturabhängigen Reaktionsentropie
<i>STEMP</i>	Berechnung der Entropieänderung in Abhängigkeit von der Temperatur
<i>STOIMA</i>	Erstellen der stöchiometrischen Matrix
<i>TCCHUNG</i>	Abschätzung thermischer Leitfähigkeiten mittels der Methode nach Chung
<i>TCELYHANLEY</i>	Abschätzung thermischer Leitfähigkeiten mittels der Methode nach Ely und Hanley
<i>TCLATINI</i>	Abschätzung thermischer Leitfähigkeiten reiner Flüssigkeiten mittels der Methode nach Latini
<i>TCLI</i>	Abschätzung thermischer Leitfähigkeiten von Flüssigkeitsmischungen mittels der Methode nach Li
<i>TCROWLEY</i>	Abschätzung thermischer Leitfähigkeiten von Flüssigkeitsmischungen mittels der Methode nach Rowley

<i>TCSATORIEDEL</i>	Abschätzung thermischer Leitfähigkeiten reiner Flüssigkeiten mittels der Methode nach Sato und Riedel
<i>UNIQUAC</i>	Berechnung von Fugazitätskoeffizienten nach der UNIQUAC Methode
<i>VANLAAR</i>	Berechnung von Aktivitätskoeffizienten nach Van Laar
<i>VBWR</i>	Berechnung des molaren Volumens mittels der BWRS- bzw. der BWR- Zustandsgleichungen, mit optionaler graphischer Ausgabe der Ergebnisse
<i>VICHUNG</i>	Abschätzung der Viskosität von Gasen mittels der Methode nach Chung
<i>VILUCAS</i>	Abschätzung der Viskosität von Gasen mittels der Methode nach Lucas
<i>VISLIQUID</i>	Abschätzung der Viskosität von reinen Flüssigkeiten nach Daten von Van Velzen, Yaws und Duhne
<i>VITEJARICE</i>	Abschätzung der Viskosität von Flüssigkeitsmischungen nach der Methode nach Teja und Rice
<i>VOLCUB</i>	Berechnung des molaren Volumens durch Lösen kubischer Gleichungen mit reduzierter Menüsteuerung
<i>VOLCUBIC</i>	Berechnung des molaren Volumens durch Lösen kubischer Gleichungen mit erweiterter Menüsteuerung
<i>WILSON</i>	Berechnung von Aktivitätskoeffizienten nach Wilson

11 Symbolverzeichnis

Die in diesem Verzeichnis zusammengestellten Symbol/Bedeutungs Relationen gelten bis auf ausdrücklich im Text anders definierte Fälle:

Symbol

A	Helmholtz Energie, [J/mol]
<u>A</u>	Element-Spezies-Matrix
a, b, c, d	Konstanten der kub. Zustandsgleichung (a,b) und

	der BWRS-Zustandsgleichung (a, b, c, d)
A_0, B_0, C_0, D_0	Konstanten der BWRS-Zustandsgleichung
E_0	
A_{12}/A_{21}	Parameter der Gleichung nach Margules und der Gleichung nach van Laar
a_i	Aktivität der Komponente i
b_h bzw. \underline{b}	konstante Molzahl bezogen auf das Element h bzw. vektoriell auf alle L Elemente
B_i	Parameter in [160]
c	Konzentration, [mol/L]
c_p	isobare Wärmekapazität, [J/(mol*K)]
c_v	Wärmekapazität bei konstantem Volumen, [J/(mol*K)]
D	Diffusionskoeffizient
f	Fugazität, [bar]
F_c	Parameter definiert unter [190]
f_i	Skalierparameter in [120]
F_P°	Korrekturterm definiert unter [176]
F_Q°	Korrekturterm definiert unter [177]
G	Freie Gibbssche Enthalpie, [J/mol]
g	molare freie Gibbssche Enthalpie, [J/mol]
G_1/G_2	Parameter definiert unter [157], [158], Parameter definiert unter [194], [195]
g_{ji}/g_{ij}	binäre Interaktionsparameter der NRTL-Gleichung
H	Enthalpie, [J/mol]
h	Elementzähler in [37] und [39]
H_i	Parameter in [124]
$\Delta_f H$	Bildungsenthalpie, [J/mol]
$\Delta_r c_p$	isobare Wärmekapazität bezogen auf eine Reaktion, [J/(mol*K)]
$\Delta_r H$	Reaktionsenthalpie, [J/mol]
\underline{I}_R	$R \times R$ Einheitsmatrix in Gleichung [42]
h_i	Skalierparameter in [121]
J	Fluß, J_N Massen Fluß, [mol/(cm ² *s)], J_{μ} Impuls Fluß, [N/cm ²], J_w Energie Fluß, [J/(cm ² *s)]
K	chemische Gleichgewichtskonstante

k	Boltzmann-Konstante, $1.38 \cdot 10^{-23}$ J/K
k_1/k_2	Geschwindigkeitskonstanten chemischer Reaktionen
k_{ij}	binärer Wechselwirkungsparameter
L	Element Anzahl
M	Molare Masse, [g/mol]
M(t)	"mass matrix"
n	Stoffmenge, [mol]
N	Anzahl Komponenten in einer Mischung
\underline{N}	stöchiometrische Matrix
N_0	Avogadrosche Konstante, [kmol ⁻¹]
N_{OH}	Anzahl an OH Gruppen im betrachteten Alkohol
O	Parameter definiert unter [164]
P	Druck, [bar]
P_{vp}	Dampfdruck, [bar]
Q	Parameter definiert unter [182]
q	Parameter definiert unter [154]
q_i	UNIQUAC-Volumenparameter der Komponente i
R	Gaskonstante, 8.314 [J/(mol*K)]
R_A	Rang der Element-Spezies-Matrix
r_i	UNIQUAC-Volumenparameter der Komponente i
S	Entropie, [J/(mol*K)]
$\Delta_r S$	Reaktionsentropie, [J/mol]
T	Temperatur, [K]
t	Zeit, [s]
T^+	Parameter der reduzierten Temperatur
T_0	Aquivalenttemperatur in [122]
U	Innere Energie, [J/mol]
u, v, w, u1, u2, u3, u4	Konstanten bezogen auf Zustandsgleichungen
u_{ij}/u_{ji}	Parameter der UNIQUAC-Gleichung
V	molares Volumen, [cm ³ /mol]
V_i^L	molares Volumen der reinen Komponente i, [cm ³ /mol]
x	Molenbruch
X	Korrekturterm in [146]
y	Molenbruch

y	Parameter definiert unter [155]
Y	Parameter definiert unter [175]
Z	Kompressibilitätskoeffizient
Z	Parameter definiert unter [156]
\underline{Z}	Lösungsmatrix des durch \underline{A} bestimmten Gleichungssystems, s. Gleichung [42]
z	Ortskoordinate, [m]
Z_1	Parameter definiert unter [162]
Z_2	Parameter definiert unter [168] und [171]

Griechische Symbole

Γ	Thermodynamische Größe
ρ	Molare Dichte, [mol/cm ³]
φ	Fugazitätskoeffizient
φ	Formfaktor in [119]
η	Viskosität, [bar]
λ	Thermische Leitfähigkeit, [W/(m*K)]
η^*	Viskositätsparameter in [125] und Parameter in [186]
η^{**}	Viskositätsparameter in [196]
ρ_0	Dichteparameter in [140]
η_0	Viskositätsparameter in [123]
η°	Viskositätsparameter in [174]
Δ	Veränderung einer thermodynamischen Größe
Σ_v	Diffusionsvolumen nach Fuller, Schettler, Giddings
Ω	Kollisionsintegral
α	Parameter definiert unter [153]
α_{ij}	Nicht-Zufalls-Parameter der NRTL-Gleichung
α_0, γ_0	Konstanten der BWRS-Zustandsgleichung
β	Parameter definiert unter [170]
ε	Charakteristische Interaktionsenergie, [J]
γ	Aktivitätskoeffizient
κ	Assoziationsfaktor nach Chung, s. Tab. 56
λ	Thermische Leitfähigkeit
λ'	Kollisions- und Translationsanteil der thermischen Leitfähigkeit in [113]

λ''	Anteil an thermischer Leitfähigkeit durch interne Freiheitsgrade in [113]
$\lambda_{ij}/\lambda_{ji}$	Interaktionsparameter der Gleichung nach Wilson
μ	Dipolmoment, [Debye]
ν	Stöchiometrischer Koeffizient
θ	Formfaktor in [119]
σ	Aus Kugelmodell resultierender Moleküldurchmesser, [Å]
ω	Pitzers azentrischer Faktor
ξ	Parameter in [171]
ψ	Parameter in [151]

Tiefgestellte Zeichen

298	Bezogen auf eine Temperatur von 298 K
a	Auf Aktivitäten bezogen
b	Bezogen auf den Siedepunkt bei 1.013 bar
c	Kritische Größe
E	Auf das Gleichgewicht (<u>E</u> quilibrium) bezogen
f	Auf Fugazitäten bezogen
i, j	Nummer einer Komponente in einer Mischung
k	Schlüssel (key) – Komponente
L	Auf die Flüssigkeit bezogen
m	Mischungsgröße
n-k	Nicht-Schlüssel (key)-Komponente
p	Isobare Größe
p	Auf Partialdrücke bezogen
r	Reduzierte Größe, z. B. $T_r = T/T_c$
x	Auf Molenbrüche bezogen
γ	Auf Fugazitätskoeffizienten bezogen
ϕ	Auf Aktivitätskoeffizienten bezogen

Hochgestellte Zeichen

°	Referenz- oder idealer Gaszustand
'	Matrixinversion
E	Exzeß-Größe
L	Flüssigkeitsgröße

12 Anhang

In diesem Anhang sind längliche oder komplexe mathematische Formelsysteme wiedergegeben, die zum Verständnis der Programmsammlung eine untergeordnete Rolle spielen, jedoch zur Vollständigkeit der Arbeit wichtig sind.

12.1 Anhang A: Thermische Leitfähigkeit nach Ely-Hanley

Bei dem Abschätzungsverfahren [1, 38, 39] werden zwei Beiträge berücksichtigt. Es wird zwischen der Übertragung von Energie durch Kollision und Translation, λ' , und einem Energiebeitrag durch interne Freiheitsgrade λ'' unterschieden. Betrachtet man Gasmischungen bei erhöhten Drücken, so berechnet sich die thermische Leitfähigkeit gemäß folgender Gleichung:

$$\lambda_m = \lambda'_m + \lambda''_m. \quad [114]$$

Weiterhin gelten folgende Gleichungen:

$$\lambda''_m = \sum_{i=1}^N \sum_{j=1}^N y_i \cdot y_j \cdot \lambda''_{ij}, \quad [115]$$

$$\lambda''_{ii} = \lambda''_i, \quad [116]$$

$$\lambda''_{ij} = \frac{2 \cdot \lambda''_i \cdot \lambda''_j}{\lambda''_i + \lambda''_j}, \quad [117]$$

$$\lambda''_i = \frac{\eta_i^*}{M_i} \cdot 1.32 \cdot \left(c_{vi} - \frac{3 \cdot R}{2} \right). \quad [118]$$

Das Molekulargewicht M_i wird hier in kg/mol und die Wärmekapazität bei konstantem Volumen in $J \cdot (\text{mol} \cdot \text{K})^{-1}$ angegeben. R ist mit $8.314 J \cdot (\text{mol} \cdot \text{K})^{-1}$ die allgemeine Gaskonstante. Zur Berechnung von η_i^* sind zunächst die reduzierten Temperaturen $T_{r,i} = T/T_{c,i}$ und damit ein Parameter T_i^+ zu bestimmen, für den gilt:

$$\begin{aligned} T_i^+ &= T_{r,i} \text{ bei } T_{r,i} \leq 2 \\ T_i^+ &= 2 \text{ bei } T_{r,i} > 2 \end{aligned} \quad [119]$$

Daraus berechnen sich dann Formfaktoren der Komponenten relativ zu Methan:

$$\begin{aligned} \theta_i &= 1 + (\varpi_i - 0.011) \cdot \left(0.56553 - 0.86276 \cdot \ln T_i^+ - \frac{0.69852}{T_i^+} \right) \\ \varphi_i &= [1 + (\varpi_i - 0.011) \cdot (0.38560 - 1.1617 \cdot \ln T_i^+)] \cdot \frac{0.288}{Z_{c,i}} \end{aligned} \quad [120]$$

Die Werte 0.011 und 0.288 repräsentieren den azentrischen Faktor und die kritische Temperatur der Referenzsubstanz Methan. Mit den so erhaltenen Formfaktoren berechnen sich Skalierparameter der Temperatur und des Volumens:

$$f_i = \frac{T_{c,i}}{190.4} \cdot \theta_i, \quad [121]$$

$$h_i = \frac{V_{c,i}}{99.2} \cdot \varphi_i. \quad [122]$$

Dabei sind 190.4 die kritische Temperatur in Kelvin und der Wert 99.2 das kritische Volumen in cm³/mol von Methan. Der Parameter f aus Gleichung [121] wird benutzt um durch eine Äquivalenttemperatur T₀ die Viskosität η_{0,i} von Methan bei geringem Druck abzuschätzen:

$$T_{0,i} = \frac{T}{f_i}, \quad [123]$$

$$\eta_{0,i} = 10^{-7} \cdot \sum_{n=1}^9 C_n \cdot T_{0,i}^{(n-4)/3}. \quad [124]$$

Die Koeffizienten C_n für die Reihe in Gleichung [124] sind wie folgt definiert:

$$C = \begin{bmatrix} 2.9077 \cdot 10^6 & -4.33190 \cdot 10^5 & 4.32517 \cdot 10^2 \\ -3.31287 \cdot 10^6 & 7.06248 \cdot 10^4 & -1.44591 \cdot 10^1 \\ 1.6081 \cdot 10^6 & -7.11662 \cdot 10^3 & 2.03712 \cdot 10^{-1} \end{bmatrix}.$$

Das so erhaltene $\eta_{0,i}$ wird verwandt, um η_i^* für Gleichung [118] zu bestimmen:

$$H_i = \left(\frac{16.04 \cdot 10^{-3}}{M_i} \right)^{0.5} \cdot f_i^{0.5} \cdot h_i^{-2/3}, \quad [125]$$

$$\eta_i^* = \eta_{0,i} \cdot H_i \cdot \frac{M_i}{16.04 \cdot 10^{-3}}. \quad [126]$$

Mit Gleichung [126] ist somit η_i^* und damit λ_i'' in Gleichung [117] zugänglich. Zur Berechnung des Kollisions- und Translationsanteils der thermischen Leitfähigkeit λ_m' werden zunächst auch die Parameter T_i^+ und V_i^+ definiert:

$$\begin{aligned} T_i^+ &= T_{r,i} \quad \text{bei} \quad T_{r,i} \leq 2 \\ T_i^+ &= 2 \quad \text{bei} \quad T_{r,i} > 2 \end{aligned} \quad [127]$$

$$\begin{aligned} V_i^+ &= V_{r,i} \quad \text{bei} \quad 0.5 < V_{r,i} < 2 \\ V_i^+ &= 0.5 \quad \text{bei} \quad V_{r,i} \leq 0.5 \\ V_i^+ &= 2 \quad \text{bei} \quad V_{r,i} \geq 2 \end{aligned} \quad [128]$$

Auch sind die Formfaktoren θ_i' und ϕ_i' für jede Komponente i erforderlich:

$$\theta_i = 1 + (\varpi_i - 0.011) \cdot [0.09957 - 0.86276 \cdot \ln T_i^+ + (0.31664 - \frac{0.46568}{T_i^+}) \cdot (V^+ - 0.5)] \quad , \quad [129]$$

$$\phi_i = \{1 + (\varpi_i - 0.011) \cdot [0.39490 \cdot (V_i^+ - 1.02355) - 0.92381 \cdot (V_i^+ - 0.75464) \cdot \ln T_i^+]\} \cdot \frac{0.288}{Z_{c,i}} \quad [130]$$

Diese Formfaktoren werden in Relation gesetzt zu den entsprechenden Werten für Methan und ergeben sich durch die Gleichungen [121] und [122]. Auf die so berechneten Werte werden nun die folgenden Mischungsregeln angewendet:

$$h_m = \sum_{i=1}^N \sum_{j=1}^N y_i \cdot y_j \cdot h_{ij} \quad , \quad [131]$$

$$h_{ii} = h_i \quad , \quad [132]$$

$$h_{ij} = \frac{1}{8} \cdot [\sqrt[3]{h_i} + \sqrt[3]{h_j}]^3 \quad , \quad [133]$$

$$f_m = \frac{\sum_{i=1}^N \sum_{j=1}^N y_i \cdot y_j \cdot f_{ij} \cdot h_{ij}}{h_m} \quad , \quad [134]$$

$$f_{ii} = f_i \quad , \quad [135]$$

$$f_{ij} = \sqrt{f_i \cdot f_j} \quad , \quad [136]$$

$$M_m = \left[\sum_{i=1}^N \sum_{j=1}^N y_i \cdot y_j \cdot M_{ij}^{-1/2} \cdot f_{ij}^{1/2} \cdot h_{ij}^{-4/3} \right]^{-2} \cdot (f_m \cdot h_m^{-8/3}) \quad , \quad [137]$$

$$M_{ii} = M_i \quad , \quad [138]$$

$$M_{ij} = \frac{2 \cdot M_i \cdot M_j}{M_i + M_j} \quad [139]$$

Mit diesen Mischungsgrößen werden dann die Äquivalenztemperatur T_0 und die Dichte ρ_0 erhalten.

$$T_0 = \frac{T}{f_m} \quad [140]$$

$$\rho_0 = \frac{16.04}{V_m} \cdot h_m \quad [141]$$

Mit Gleichung [124] berechnet sich η_0 . Die erste Komponente der thermischen Leitfähigkeit λ'_m ergibt sich damit zu

$$\lambda(1) = 1944 \cdot \eta_0 \quad [142]$$

Die zweite Komponente λ'_m wird durch Gleichung [143] erhalten:

$$\lambda(2) = \left\{ b_1 + b_2 \cdot \left[b_3 - \ln \left(\frac{T_0}{b_4} \right) \right]^2 \right\} \cdot \rho_0 \quad [143]$$

Für die Berechnung des dritten Anteils von λ'_m mit Gleichung [144]

$$\begin{aligned} \lambda(3) = & \exp \left(a_1 + \frac{a_2}{T_0} \right) \cdot \left\{ \exp \left[\left(a_3 + \frac{a_4}{T_0^{3/2}} \right) \cdot \rho_0^{0.1} \right. \right. \\ & \left. \left. + \left(\frac{\rho_0}{0.1617} - 1 \right) \cdot \rho_0^{1/2} \cdot \left(a_5 + \frac{a_6}{T_0} + \frac{a_7}{T_0^2} \right) \right] - 1.0 \right\} \cdot 10^{-3} \end{aligned} \quad [144]$$

und Gleichung [143] werden die Konstanten a und b benötigt:

$$a = \begin{bmatrix} -7.19771 \\ 85.67822 \\ 12.47183 \\ -984.6252 \\ 0.3594685 \\ 69.79841 \\ -872.8833 \end{bmatrix}, \quad b = \begin{bmatrix} -2.5276 \\ 3.3433 \\ 1.12 \\ 1.680 \cdot 10^2 \end{bmatrix}.$$

Damit ergibt sich der erste Term aus Gleichung [114] zu

$$H_m = \left(\frac{16.04 \cdot 10^{-3}}{M_m} \right) \cdot \frac{f_m^{1/2}}{h_m^{7/2}} \cdot X_m, \quad [145]$$

$$\lambda'_m = [\lambda(1) + \lambda(2) + \lambda(3)] \cdot H_m, \quad [146]$$

Für höhere Genauigkeit haben Ely und Hanley einen Korrekturterm X (s. Gl. [145]) vorgeschlagen, der sich wie folgt berechnet:

$$X = \left\{ \left[1 - \frac{T}{f_i} \cdot \left(\frac{\partial f_i}{\partial T} \right)_v \right] \cdot \frac{0.288}{Z_{ci}} \right\}^{3/2} \quad [147]$$

mit

$$\left(\frac{\partial f_i}{\partial T} \right)_v = \frac{T_{ci}}{190.4} \cdot \left(\frac{\partial \theta}{\partial T} \right)_{vi}, \quad [148]$$

wobei für $T^+ = \text{const.} = 2$ gilt $(\partial \theta / \partial T)_v = 0$. Ist hingegen $T^+ = T_r$, so ergibt sich

$$\left(\frac{\partial\theta}{\partial T}\right)_{vi} = (\omega_i - 0.011) \cdot \left[\frac{-0.86276}{T} + (V_i^+ - 0.5) \cdot \frac{0.46568 \cdot T_{ci}}{T^2} \right] \quad [149]$$

und

$$X_m = \sum_{i=1}^n x_i \cdot X_i \quad [150]$$

12.2 Anhang B: Thermische Leitfähigkeit nach Chung

Auch die Abschätzung der thermischen Leitfähigkeit nach Chung [1, 53] basiert auf der Trennung der Einflüsse in einen translatorischen Beitrag und einen Beitrag an interner Energie:

$$\lambda = \frac{31.2 \cdot \eta \cdot \psi}{M'} \cdot (G_2^{-1} + B_6 \cdot y) + q \cdot B_6 \cdot y^2 \cdot \sqrt{T_r} \cdot G_2 \quad [151]$$

Dabei gilt weiterhin

$$\psi = 1 + \alpha \cdot \frac{(0.215 + 0.28288 \cdot \alpha - 1.061 \cdot \beta + 0.26665 \cdot Z)}{0.6366 + \beta \cdot Z + 1.061 \cdot \alpha \cdot \beta} \quad [152]$$

und

$$\beta = 0.7862 - 0.7109 \cdot \omega + 1.3168 \cdot \omega^2 \quad [153]$$

Hier ist β die von Chung angegebene empirische Korrelation für den Beitrag der inneren Energie zu λ . Sie ist jedoch nur für unpolare Substanzen anwendbar. Für polare Gase stellt dieser Term einen substanzspezifischen Wert dar. Weiterhin sind:

$$\alpha = \left(\frac{c_v}{R} \right) - \frac{3}{2}, \quad [154]$$

$$q = 3.586 \cdot 10^{-3} \cdot \frac{\sqrt{T_c/M}}{V_c^{2/3}}, \quad [155]$$

$$y = \frac{V_c}{6 \cdot V}, \quad [156]$$

$$Z = 2.0 + 10.5 \cdot T_r^2, \quad [157]$$

$$G_1 = \frac{1 - 0.5 \cdot y}{(1 - y)^3} \quad [158]$$

und

$$G_2 = \frac{B_1/y \cdot (1 - \exp(-B_4 \cdot y)) + B_2 \cdot G_1 \cdot \exp(B_5 \cdot y) + B_3 \cdot G_1}{B_1 \cdot B_4 + B_2 + B_3} \quad [159]$$

Hier sind T die Temperatur und T_c die kritische Temperatur in Kelvin, V das molare Volumen und V_c das kritische molare Volumen in cm³/mol und M gibt die molare Masse in kg/mol wieder. ω ist hier der dimensionslose azentrische Faktor. Wird weiterhin die Viskosität η⁰ des Gases bei geringem Druck in N*s/m² angegeben, so ergibt sich die thermische Leitfähigkeit λ in W/(m*K).

Die Koeffizienten B_i sind Funktionen des azentrischen Faktors ω, des reduzierten Dipolmomentes μ_r

$$\mu_r = \frac{131.3 \cdot \mu}{\sqrt{T_c \cdot V_c}} \quad [160]$$

und eines Assoziationsfaktors κ , für den Chung Werte angegeben hat (s. Kapitel 7.2.2, Anhang 12.4):

$$B_i = a_i + b_i \cdot \varpi + c_i \cdot \mu_r^4 + d_i \cdot \kappa . \quad [161]$$

Die Parameter der Gleichung [161] sind in Tabelle 64 definiert.

Tab 64: Numerische Werte der Parameter in Gleichung [161]

i	a _i	b _i	c _i	d _i
1	2.4166	7.4824*10 ⁻¹	-9.1858*10 ⁻¹	1.2172*10 ²
2	-5.0924*10 ⁻¹	-1.5094	-4.9991*10 ¹	6.9983*10 ¹
3	6.6107	5.6207	6.4760*10 ¹	2.7039*10 ¹
4	1.4543*10 ¹	-8.9139	-5.6379	7.4344*10 ¹
5	7.9274*10 ⁻¹	8.2019*10 ⁻¹	-6.9369*10 ⁻¹	6.3173
6	-5.8634	1.2801*10 ¹	9.5893	6.5529*10 ¹
7	9.1089*10 ¹	1.2811*10 ²	-5.4217*10 ¹	5.2381*10 ²

Werden Gasgemische betrachtet, so müssen Mischungsregeln in die Abschätzung einbezogen werden. Die Viskosität der Mischung bei geringem Druck η^0_m ergibt sich entsprechend der in Kapitel 7.2.2 dargestellten Zusammenhänge. Die notwendigen Mischungsgrößen sind die Molaremasse M_m , die kritische Temperatur T_{cm} und das Dipolmoment ω_m . Die zu berücksichtigenden Mischungsregeln sind in Anhang 12.4 erläutert. Dabei wird die Molaremasse in kg/mol angegeben; für die Wärmekapazität bei konstantem Volumen c_{vm} gilt folgende Mischungsregel:

$$c_{vm} = \sum_{i=1}^N y_i \cdot c_{vi} . \quad [162]$$

12.3 Anhang C: Gas-Viskosität nach Lucas

Lucas schlägt zur Abschätzung der Viskosität folgendes Vorgehen vor. Zunächst wird ein Parameter Z_1 bestimmt:

$$Z_1 = (0.807 \cdot T_r^{0.618} - 0.357 \cdot \exp(-0.449 \cdot T_r) + 0.340 \cdot \exp(-4.058 \cdot T_r) + 0.0818) \cdot F_{P_m}^\circ \cdot F_{Q_m}^\circ, \quad [163]$$

wobei für T_r Gleichung [164] gilt:

$$T_r = \frac{T}{T_{c_m}}. \quad [164]$$

Der Index m bezieht sich hier auf Mischungsgrößen:

$$T_{c_m} = \sum_{i=1}^N y_i \cdot T_{c_i}, \quad [165]$$

$$F_{P_m}^\circ = \sum_{i=1}^N y_i \cdot F_{P_i}^\circ \quad [166]$$

und

$$F_{Q_m}^\circ = \left(\sum_{i=1}^N y_i \cdot F_{Q_i}^\circ \right) \cdot \theta. \quad [167]$$

Wenn die Indizes 1 und 2 die Mischungskomponente mit der größten bzw. kleinsten molaren Masse repräsentiert, so gilt für θ folgender Zusammenhang: Ist $M_1/M_2 > 9$ und $0.05 < y_1 < 0.7$, so ist

$$\theta = 1 - 0.01 \cdot \left(\frac{M_1}{M_2} \right)^{0.87}, \quad [168]$$

sonst gilt $\theta = 1$.

Weiterhin wird ein zweiter Parameter Z_2 bestimmt. Ist $T_r \leq 1$ und $P_r < (P_{vp}/P_{cm})$, so gilt für Z_2

$$Z_2 = 0.6 + 0.76 \cdot P_r^\alpha + (6.99 \cdot P_r^\beta - 0.6) \cdot (1 - T_r), \quad [169]$$

mit α und β als

$$\alpha = 3.262 + 14.98 \cdot P_r^{5.508}, \quad [170]$$

$$\beta = 1.39 + 5.746 \cdot P_r. \quad [171]$$

Gilt hingegen, daß $1 < T_r < 40$ und $0 < P_r < 100$, so ergibt sich Z_2 gemäß des folgenden Ausdruckes:

$$Z_2 = \eta^\circ \cdot \xi \cdot \left[1 + \frac{a \cdot P_r^e}{b \cdot P_r^f + (1 + c \cdot P_r^d)^{-1}} \right]. \quad [172]$$

Dabei ist P_r der reduzierte Druck durch

$$P_r = \frac{P}{P_{cm}} \quad [173]$$

gegeben und P_{cm} ergibt sich gemäß folgender Gleichung:

$$P_{cm} = R \cdot T_{cm} \cdot \frac{\sum_i y_i \cdot Z_{c_i}}{\sum_i y_i \cdot V_{c_i}}. \quad [174]$$

Der Ausdruck $\eta^\circ \xi$ in Gleichung [172] wird über Gleichung [163] erhalten:

$$\eta^{\circ} \cdot \xi = Z_1. \quad [175]$$

Für die in Gleichung [172] verwandten Konstanten a bis d gelten die folgenden Zusammenhänge:

$$a = \frac{a_1}{T_r} \cdot \exp(a_2 \cdot T_r^{\gamma}),$$

$$b = a \cdot (b_1 \cdot T_r - b_2),$$

$$c = \frac{c_1}{T_r} \cdot \exp(c_2 \cdot T_r^{\delta}),$$

$$d = \frac{d_1}{T_r} \cdot \exp(d_2 \cdot T_r^{\varepsilon}),$$

$$e = 1.3088,$$

$$f = f_1 \cdot \exp(f_2 \cdot T_r^{\zeta}).$$

Tab 65: Konstanten für die Abschätzung der Viskosität nach Lucas

Parameter	i = 1	i = 2		
a _i	1.245*10 ⁻³	5.1726	γ	-0.3286
b _i	1.6553	1.2723		
c _i	0.4489	3.0578	δ	-37.7332
d _i	1.7368	2.2310	ε	-7.6351
f _i	0.9425	-0.1853	ζ	0.4489

Zur Berechnung der Korrekturterme F_P und F_Q wird der Hilfsfaktor Y definiert:

$$Y = \frac{Z_2}{Z_1}. \quad [176]$$

Daraus ergeben sich dann die Korrekturterme gemäß

$$F_p = \frac{1 + (F_p^\circ - 1) \cdot Y^{-3}}{F_p^\circ} \quad [177]$$

und

$$F_Q = \frac{1 + (F_Q^\circ - 1) \cdot (Y^{-1} - 0.007 \cdot (\ln Y)^4)}{F_Q^\circ}. \quad [178]$$

Zur Berechnung der Faktoren F_p° und F_Q° ist ein reduziertes Dipolmoment μ_{ri} notwendig:

$$\mu_{ri} = 52.46 \cdot \frac{\mu^2 \cdot P_{ci}}{T_{ci}^2}. \quad [179]$$

In Abhängigkeit von μ_{ri} werden dann die Faktoren F_{pi}° bestimmt. Es gilt mit $0 \leq \mu_{ri} \leq 0.022$

$$F_{pi}^\circ = 1, \quad [180]$$

mit $0.022 \leq \mu_{ri} \leq 0.075$

$$F_{pi}^\circ = 1 + 30.55 \cdot (0.292 - Z_c)^{1.72}, \quad [181]$$

und mit $0.075 \leq \mu_{ri}$

$$F_{pi}^\circ = 1 + 30.55 \cdot (0.292 - Z_c)^{1.72} \cdot |0.96 + 0.1 \cdot (T_r - 0.7)|. \quad [182]$$

Die Faktoren F_{Qi}° berechnen sich gemäß folgender Formel:

$$F_Q^o = 1.22 \cdot Q^{0.15} \cdot M_m \sqrt{1 + 0.00385 \cdot (T_r - 12)^2} \cdot S, \quad [183]$$

wobei gilt $S = 1$, wenn $(T_r - 12) > 0$ und $S = -1$, wenn $(T_r - 12) < 0$.
Für Q gelten die folgenden Werte:

Substanz	He	H ₂	D ₂
Q	1.38	0.76	0.52

Die molare Masse der Mischung M_m ergibt sich durch eine Beziehung, analog Gleichung [165]:

$$M_m = \sum_{i=1}^N y_i \cdot M_i. \quad [184]$$

Die Viskosität resultiert damit durch die Gleichungen [166] und [167] gemäß folgendem Zusammenhang:

$$\eta = \frac{Z_2 \cdot F_p \cdot F_Q}{\xi}. \quad [185]$$

Dabei wird ξ über Gleichung [86] erhalten.

12.4 Anhang D: Gas-Viskosität nach Chung

Ausgehend von Gleichung [83] wendet Chung empirische Korrelationen an, um einer höheren Flüssigkeitsdichte Rechnung zu tragen. Es ergibt sich folgender Ansatz:

$$\eta = \eta^* \cdot \frac{36.244 \cdot \sqrt{M \cdot T_c}}{V_c^{2/3}}. \quad [186]$$

Hier sind η die Viskosität des betrachteten Fluids in μP , M_m die Molaremasse in g/mol , T_c die kritische Temperatur in Kelvin und

V_c das kritische Volumen in cm^3/mol . η^* wird durch folgenden Ausdruck erhalten:

$$\eta^* = \frac{\sqrt{T^*}}{\Omega} \cdot (F_c \cdot (G_2^{-1} + E_6 \cdot y)) + \eta^{**} . \quad [187]$$

Chung wendet für den Kugeldurchmesser σ einen, Gleichung [188] entsprechenden, Ausdruck an:

$$\sigma = 0.809 \cdot V_c^{1/3} . \quad [188]$$

Mit

$$\frac{\varepsilon}{k} = \frac{T_c}{1.2593} \quad [189]$$

und Gleichung [89] erhält Chung einen Ausdruck für T^* :

$$T^* = 1.2593 \cdot T_r . \quad [190]$$

Das Kollisionsintegral Ω wird mit Gleichung [88] erhalten. In Gleichung [187] ist F_c gegeben durch

$$F_c = 1 - 0.2756 \cdot \omega + 0.059035 \cdot \mu_r^4 + \kappa . \quad [191]$$

Hier sind ω der azentrische Faktor, κ ein Korrekturfaktor für stark polare Substanzen und μ_r ein reduziertes Dipolmoment. Dieses ist zugänglich gemäß

$$\mu_r = 131.3 \cdot \frac{\mu}{\sqrt{V_c \cdot T_c}} . \quad [192]$$

Die von Chung für den substanzspezifischen Korrekturfaktor κ angegebenen Werte finden sich in Tabelle 66.

Tab 66: Korrekturfaktor κ für Gleichung [191]

Substanz	κ	Substanz	κ
Methanol	0.215	n-Pentanol	0.122
Ethanol	0.175	n-Hexanol	0.144
n-Propanol	0.143	n-Heptanol	0.109
i-Propanol	0.143	Essigsäure	0.0916
n-Butanol	0.132	Wasser	0.076
i-Butanol	0.132		

Für andere - nicht in Tabelle 66 enthaltene - Alkohole schlägt Chung eine Funktion für κ vor:

$$\kappa = 0.0682 + 0.2767 \cdot \left[17 \cdot \frac{N_{\text{OH}}}{M} \right], \quad [193]$$

wobei N_{OH} die Anzahl an -OH Gruppen des Alkohols darstellt. Die weiteren in Gleichung [187] auftretenden Größen definieren sich wie folgt:

$$y = \frac{\rho \cdot V_c}{6}. \quad [194]$$

ρ ist die Flüssigkeitsdichte in mol/cm³,

$$G_1 = \frac{1 - 0.5 \cdot y}{(1 - y)^3}, \quad [195]$$

$$G_2 = \frac{E_1 \cdot \frac{1 - e^{-E_4 \cdot y}}{y} + E_2 \cdot G_1 \cdot e^{E_5 \cdot y} + E_3 \cdot G_1}{E_1 E_4 + E_2 + E_3}, \quad [196]$$

$$\eta^{**} = E_7 \cdot y^2 \cdot G_2 \cdot \exp\left(E_8 + \frac{E_9}{T^*} + \frac{E_{10}}{T^{*2}}\right). \quad [197]$$

Die in den Gleichungen [187], [196] und [197] verwendeten Parameter E_i sind als Funktion von ω , μ_r und κ zugänglich. Die dazu notwendigen Koeffizienten sind in Tabelle 67 zusammengefaßt:

$$E_i = a_i + b_i \cdot \kappa + c_i \cdot \mu_r^4 + d_i \cdot \kappa. \quad [198]$$

Tab 67: Koeffizienten für E_i in Gleichung [198]

i	a_i	b_i	c_i	d_i
1	6.324	50.412	-51.68	1189.0
2	$1.2 \cdot 10^{-3}$	$-1.154 \cdot 10^{-3}$	$-6.257 \cdot 10^{-3}$	0.03728
3	5.283	254.209	-168.48	3898.0
4	6.623	38.096	-8.464	31.42
5	19.745	7.63	-14.354	31.53
6	-1.9	-12.537	4.985	-18.15
7	24.275	3.45	-11.291	69.35
8	0.7972	1.117	0.01235	-4.117
9	-0.2382	0.0677	-0.8163	4.025
10	0.06863	0.3479	0.5926	-0.727

Wird ein Flüssigkeitsgemisch betrachtet, so sind verschiedene Mischungsregeln zu berücksichtigen:

$$T_{c_m} = 1.2593 \cdot \left(\frac{\varepsilon_m}{k} \right), \quad [199]$$

wobei gilt

$$\left(\frac{\varepsilon_m}{k}\right) = \frac{\sum_{i=1}^N \sum_{j=1}^N x_i \cdot x_j \cdot \left(\frac{\varepsilon_{ij}}{k}\right) \cdot \sigma_{ij}^3}{\sigma_m^3}. \quad [200]$$

Weiterhin sind

$$\sigma_m^3 = \sum_{i=1}^N \sum_{j=1}^N x_i \cdot x_j \cdot \sigma_{ij}^3, \quad [201]$$

$$\frac{\varepsilon_{ij}}{k} = \frac{\varepsilon_i}{k} = \frac{T_{c_i}}{1.2593}, \quad [202]$$

$$\frac{\varepsilon_{ij}}{k} = \zeta_{ij} \cdot \sqrt{\frac{\varepsilon_i}{k} \cdot \frac{\varepsilon_j}{k}} \quad [203]$$

und

$$\sigma_{ii} = \sigma_i = 0.809 \cdot V_{c_i}^{1/3} \quad [204]$$

zu berücksichtigen. Weitere Mischungsgrößen sind das kritische Volumen

$$V_{c_m} = \left(\frac{\sigma_m}{0.809}\right)^3, \quad [205]$$

der azentrische Faktor

$$\omega_m = \frac{\sum_{i=1}^N \sum_{j=1}^N x_i \cdot x_j \cdot \omega_{ij} \cdot \sigma_{ij}^3}{\sigma_m^3}, \quad [206]$$

und die molare Masse

$$M_m = \frac{\left(\sum_{i=1}^N \sum_{j=1}^N x_i \cdot x_j \cdot \left(\frac{\varepsilon_{ij}}{k} \right) \cdot \sigma_{ij}^2 \cdot \sqrt{M_{ij}} \right)^2}{\left(\frac{\varepsilon_m}{k} \right) \cdot \sigma_m^2}, \quad [207]$$

wobei für M_{ij} folgender Zusammenhang gilt:

$$M_{ij} = \frac{2 \cdot M_j \cdot M_i}{M_j + M_i}. \quad [208]$$

Abschließend sind noch die Mischungsgrößen μ und κ zu betrachten:

$$\mu_m^4 = \sigma_m^3 \cdot \sum_{i=1}^N \sum_{j=1}^N \left(\frac{x_i \cdot x_j \cdot \mu_i^2 \cdot \mu_j^2}{\sigma_{ij}^3} \right), \quad [209]$$

$$\kappa_m = \sum_{i=1}^N \sum_{j=1}^N x_i \cdot x_j \cdot \kappa_{ij}, \quad [210]$$

wobei für letztere Gleichung gilt:

$$\kappa_{ij} = \sqrt{\kappa_i \cdot \kappa_j} \quad \kappa_{ii} = \kappa_i. \quad [211]$$

13 Literatur

- [1] REID, R. C.; PRAUSNITZ, J. M.; POLING, B. E.: *The Properties of Gases & Liquids*. 4. Aufl., McGraw-Hill, New York, 1987
- [2] BAERNS, M.; HOFMANN, H.; RENKEN, A.: *Chemische Reaktionstechnik*. 2. Aufl., Thieme, Stuttgart, 1992
- [3] SMITH, J. M.; VAN NESS, H. C.; ABBOTT, A. A.: *Introduction to chemical engineering thermodynamics*. McGraw-Hill, New York, 1996
- [4] SMITH, W. R.; MISSEN, R. W.: *Chemical reaction equilibrium analysis, theory and algorithms*. Wiley, New York, 1982
- [5] OTTMANN, T.; WIDMAYER, P.: *Algorithmen und Datenstrukturen*. 3. Aufl., Spektrum Akademischer Verlag, Heidelberg, 1996
- [6] WALAS, S. M.: *Phase equilibrium in chemical engineering*. Butterworth-Heinemann, Boston, 1985
- [7] GMEHLING, J.; KOLBE, B.: *Thermodynamik*. 2. Aufl., VCH, Weinheim, 1992
- [8] TAKAHASHI, S.: *J. Chem. Eng. Japan*. 7: 417, 1974
- [9] DALLMANN, H.; ELSTER, K.-H.: *Einführung in die höhere Mathematik*. Vieweg & Sohn, Braunschweig, 1968
- [10] BARNETT, S.: *Matrices: methods and applications*. Oxford applied mathematics and computing science series., 1990
- [11] STULL, D. R.; WESTRUM, E. F.; SINKE, G. C.: *The Chemical Thermodynamics of Organic Compounds*. Wiley, New York, 1969
- [12] WEDLER, G.: *Lehrbuch der Physikalischen Chemie*. 2. Aufl. VCH, Weinheim, 1987
- [13] STEPHAN, K.; MAYINGER, F.: *Thermodynamik Band 1 Einstoffsysteme*. 14. Aufl. Springer, Berlin, 1992

- [14] STEHPHAN, K.; MAYINGER, F.: *Thermodynamik Band 2 Mehrstoffsysteme und chemische Reaktionen*. 13. Aufl. Springer, Berlin, 1992
- [15] BRENAN, K. E., CAMPBELL, S. L., PETZOLD, L. R.: *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Classics in applied Mathematics 14, SIAM, Philadelphia, 1996
- [16] JOHNSON, D. W.; COLVER, C. P.: *Hydrocarbon Processing*. 12: 47, 1968
- [17] JOHNSON, D. W.; COLVER, C. P.: *Hydrocarbon Processing*. 1: 48, 1969
- [18] STARLING, K.E.: *Fluid thermodynamic properties for light petroleum systems*. Gulf publishing company, Houston, 1973
- [19] PRAUSNITZ, J.; ANDERSON, T.; GRENS, E.; ECKERT, C.; HSIEH, R.; O'CONNELL, J.: *Computer calculation for multicomponent vapor-liquid and liquid-liquid equilibria*. Prentice-Hall, Englewood Cliffs, 1980
- [20] PANTELIDES, C. C.: *SIAM J. Sci. Stat. Comput.* 9, 2, 1988
- [21] BRÚLE, M. R.; STARLING, K. E.: *Ind. Eng. Chem. Process Design Develop.* 23: 833, 1984
- [22] ASCHER, U. M.; PETZOLD, L. R.: *Computer methods for ordinary differential equations and differential-algebraic equations..* SIAM, Philadelphia, 1988
- [23] ATKINS, P.: *Physikalische Chemie..* VCH, Weinheim, 1990
- [24] GMEHLING, J.; ONKEN, U.: *Vapor-Liquid Equilibrium Data Collection*. DECHEMA Chemistry Data Series, VI, DECHEMA 1977 ff.
- [25] POLING, B. E.; GRENS II, E. A.; PRAUSNITZ, J. M.: *Ind. Eng. Chem. Process Des. Dev.* 20, 11, 127-130, 1981
- [26] MILLS, M. B.; WILLS, M. J.; BHIRUD, V. L.: *AIChE Journal*, 26, 6, 902 – 910, 1980

- [27] SERES, L.; ZALOTAL, L.; MARTA, L.: Acta. Phys. Chem., 23, 4, 433, 1977
- [28] SERES, L.: Acta. Phys. Chem., 27, 1-4, 31, 1981
- [29] MODELL, M.; REID, R. C.: *Thermodynamics and its Application in chemical engineering*, 2. Aufl., Prentice-Hall, Englewood Cliffs, 159, 1983
- [30] SLEDJESKI, E. W.: Ind. Eng. Chem., 43, 12, 2913-2915, 1951
- [31] SOAVE, G.: Chem. Eng. Sci., 27: 1197, 1972
- [32] PENG, D. Y.; ROBINSON, D. B.; Ind. Eng. Chem. Fundam., 15, 59, 1976
- [33] PRESS, W. H.; FLANNERY, B. P.; TEUKOLSKY, S. A.; VETTERLING, W. T.: *Numerical Recipes, The art of scientific computing (FORTRAN Version)*. Cambridge university press, Cambridge, 1989
- [34] REGENSBURGER, J.: MATHWORKS INC.; Email Kontakt vom 30.3.1999
- [35] ANTOINE, C.: C.R., 107, 681, 836, 1888
- [36] MEINTJES, K.; MORGAN, A. P.: *Chemical equilibrium systems as numerical test problems*, ACM Tran. Math. Softw. 16, 2, 143-151, 1990
- [37] WAGNER, W.: *Cryogenics*, 13,470, 1973
- [38] ELY, J. F.; HANLEY, H. J. M.: Ind. Eng. Chem. Fundam., 20, 323, 1981
- [39] ELY, J. F.; HANLEY, H. J. M.: Ind. Eng. Chem. Fundam., 22, 90-97, 1983
- [40] SANDERSON, R. V.; CHIEN, H. Y.: Ind. Eng. Chem. Des. Develop., 12, 1, 81-85, 1973
- [41] MEINTJES, K.; MORGAN, A.: Appl. Math. Comput., 22, 333-361, 1987
- [42] LÖWE, A.: *Chemische Reaktionstechnik mit MATLAB und Simulink*, VCH, Weinheim, 2000
- [43] HOFFMANN, J.: Chem. Tech., 29, 304, 1977

- [44] ZACHMANN, H. G.: *Mathematik für Chemiker*, 4.Aufl., VCH, Weinheim, 1990
- [45] MATLAB-Optimization Toolbox User's Guide, The MathWorks, Inc., 1996
- [46] MATLAB-The language of technical computing User's Guide, Version 5, The MathWorks, Inc., 1996
- [47] MÜLLER-ERLWEIN, E.: *Computeranwendungen in der chemischen Reaktionstechnik*, VCH, Weinheim, 1991
- [48] MATLAB-Symbolic Math Toolbox User's Guide, The MathWorks, Inc., 1997
- [49] LÖWE, A; KLEIN, O.; OHRENBERG, A.: *Simulation der enzymkatalysierten 7-ACS-Synthese in Fortran unter optionaler Verwendung von Matlab/Simulink*. Forschungsprojekt, Institut für Technische Chemie, Braunschweig, 1997
- [50] MATLAB-Building GUIs with Matlab, The MathWorks, Inc., 1996
- [51] KLEIN, O.: *Simulation von Reaktionskolonnen*. Technische Universität, Diplomarbeit, Braunschweig, 1997
- [52] CHRISTIANSEN, L. J.: *Thermodynamic Properties in Ammonia Synthesis*. Haldor Topsøe A/S Copenhagen, in *Ammonia*, NIELSEN, A. (Hrsg.), Springer, Berlin 1995
- [53] STEPHAN, K.; HECKENBERGER, T.: *Viscosity and thermal conductivity of fluid mixtures*. DECHEMA Chemistry Data Series, X, 1, DECHEMA, 1988 ff.
- [54] ROWLEY, R. L.: *Chem. Eng. Sci.*, 37: 897, 1982
- [55] CHUNG, T.-H.; LEE, L. L.; STARLING, K. E.; *Ind. Eng. Chem. Fundam.* 23, 8, 1984
- [56] HANLEY, H. J. M.: *Cryogenics*, 16, 11, 643, 1976
- [57] BARONCINI, C.; DI FILLIPPO, P.; LATINI, G.; PACETTI, M.: *Intern. J. Thermophys.*, 2, 1, 21, 1981
- [58] NEUFELD, P. D.; JANZEN, A. R.; AZIZ, R. A.: *J. Chem. Phys.*, 57, 1100, 1972

- [59] TEJA, A. S.: *AIChE J.*, 26, 337 1980
- [60] TEJA, A. S.; SANDLER, S. I.: *AIChE J.*, 26, 341, 1980
- [61] TEJA, A. S.; RICE, P.: *Chem. Eng. Sci.*, 36, 7, 1981
- [62] TEJA, A. S.; RICE, P.: *Ind. Chem. Eng. Fundam.*, 20, 77, 1981
- [63] WILKE, C. R.; LEE, C. Y.: *Ind. Eng. Chem.*, 47, 1253, 1955
- [64] SCHULER, H. (Hrsg.): *Prozeßsimulation*. 1. Aufl., VCH, Weinheim, 1995
- [65] CUTLIP, M. B.; SHACHAM, M.: *Problem Solving in Chemical Engineering with numerical methods*. Prentice Hall, NJ, 1999
- [66] SCIENTIFIC COMPUTERS (Veranst.): *MATLAB Konferenz, Konferenzunterlagen*, Stuttgart, 1997
- [67] THE MATHWORKS home. *MATLAB-Homepage*. Internet <http://www.mathworks.com>, 27.7.2000
- [68] HOFFMANN, U.; HOFMANN, H.: *Einführung in die Optimierung mit Anwendungsbeispielen aus dem Chemie-Ingenieur-Wesen*. VCH, Weinheim, 1971
- [69] PATHAK, B. K.; SINGH, V. N.; SINGH, P. C.; *Can.J. Chem. Eng.*, 59, 362, 1981
- [70] LUGG, G. A.; *Anal. Chem.*, 40, 7, 1072, 1968
- [71] OHRENBERG, A.: *Angewandte Reaktionstechnik mit MATLAB – Entwicklung einer Programmsammlung*. Dissertation, Institut für Technische Chemie, Braunschweig , 2000
- [72] STEPHAN, K.; LUCAS, K.: *Viscosity of dense fluids*, Plenum, New York, 1979
- [73] THE MATHWORKS; *Datenblatt zur Database Toolbox 1.0*, 1999
- [74] THE MATHWORKS home. *MATLAB-Homepage*. Internet <http://www.mathworks.com>, 4.9.2000
- [75] THE MATHWORKS; *Datenblatt zu DBAccess*, 1999
- [76] HEUER, A.; SAAKE, G.: *Datenbanken Konzepte und Sprachen*. MITP-Verlag, Bonn, 1997
- [77] JAKUBITH, M.: *Grundoperationen und chemische Reaktionstechnik*. VCH, Weinheim, 1998

- [78] DOHRN, R.: *Berechnung von Phasengleichgewichten*. Vieweg, Braunschweig, 1994
- [79] FITZER, E.; FRITZ, W.; EMIG, G.: *Technische Chemie Einführung in die Chemische Reaktionstechnik*. Springer, Berlin, 1995
- [80] ANTEZANA, F., J.; CHEH, H., Y.: *Ind. Eng. Chem., Fundam.* 14, 3, 224-231, 1975
- [81] GROTENDORST, J.; DORNSEIFFER, J.: *Calculation of Chemical Equilibrium Compositions*. ZAM, KFA, Jülich, 1994
- [82] BANDERMANN, F.: *Ullmanns Encyklopädie der technischen Chemie*. Bd. 1, S. 362, VCH, Weinheim, 1972
- [83] LARSON, J.: *J. Am. Chem. Soc.* 46, 1, 367-372, 1924
- [84] SUNDMACHER, K.; HOFFMANN, U.: *Chem. Eng. Sci.* 51, 10, 2359-2368, 1996
- [85] VERKADE, J.; GMEHLING, J.: *Chem. Ing. Tech.* 66, 10, 1372-1375, 1994