



Arno Löwe

Chemische Reaktionstechnik

mit MATLAB und SIMULINK

 **WILEY-VCH**

Weinheim · New York · Chichester
Brisbane · Singapore · Toronto

Aus technischen Gründen bleibt diese Seite leer

Arno Löwe

Chemische Reaktionstechnik
mit MATLAB und SIMULINK

 **WILEY-VCH**

Aus technischen Gründen bleibt diese Seite leer

Arno Löwe

Chemische Reaktionstechnik

mit MATLAB und SIMULINK

 **WILEY-VCH**

Weinheim · New York · Chichester
Brisbane · Singapore · Toronto

Univ.-Prof. i. R. Dr. Arno Löwe
Ahornweg 10
97267 Himmelstadt

Das vorliegende Werk wurde sorgfältig erarbeitet. Dennoch übernehmen Autor und Verlag für die Richtigkeit von Angaben, Hinweisen und Ratschlägen sowie für eventuelle Druckfehler keine Haftung.

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Ein Titeldatensatz für diese Publikation ist bei
Die Deutsche Bibliothek erhältlich

ISBN 3-527-30268-9

© WILEY-VCH Verlag GmbH, D-69469 Weinheim (Federal Republic of Germany), 2001

Gedruckt auf säurefreiem Papier.

Alle Rechte, insbesondere die der Übersetzung in andere Sprachen, vorbehalten. Kein Teil dieses Buches darf ohne schriftliche Genehmigung des Verlages in irgendeiner Form – durch Photokopie, Mikroverfilmung oder irgendein anderes Verfahren – reproduziert oder in eine von Maschinen, insbesondere von Datenverarbeitungsmaschinen, verwendbare Sprache übertragen oder übersetzt werden. Die Wiedergabe von Warenbezeichnungen, Handelsnamen oder sonstigen Kennzeichen in diesem Buch berechtigt nicht zu der Annahme, daß diese von jedermann frei benutzt werden dürfen. Vielmehr kann es sich auch dann um eingetragene Warenzeichen oder sonstige gesetzlich geschützte Kennzeichen handeln, wenn sie nicht eigens als solche markiert sind.

All rights reserved (including those of translation into other languages). No part of this book may be reproduced in any form – by photoprinting, microfilm, or any other means – nor transmitted or translated into a machine language without written permission from the publishers. Registered names, trademarks, etc. used in this book, even when not specifically marked as such, are not to be considered unprotected by law.

Druck: Strauss Offsetdruck, D-69509 Mörlenbach
Bindung: Großbuchbinderei J. Schäffer, D-67269 Grünstadt
Printed in the Federal Republic of Germany

Herrn Professor Dr. KURT DIALER
zum 80. Geburtstag gewidmet

Aus technischen Gründen bleibt diese Seite leer

Inhaltsverzeichnis

Vorwort	XIII
Notation	XIX
0.1 Schriftauszeichnung	XIX
0.2 Symbolverzeichnis	XX
0.3 Abkürzungsverzeichnis	XXIII
1 Einleitung	1
2 MATLAB und SIMULINK	5
2.1 Einführung	5
2.2 Kostproben aus MATLABs Programmier- 2.2.1 Vektorisierung	6
2.2.2 Matrizenmanipulation	7
2.2.3 Zeichenkettenverarbeitung	8
2.2.4 MATLABs Befehl <i>eval</i>	9
2.3 MATLAB-Programme oder SIMULINK-Modelle?	10
2.4 S-Funktionen, C und FORTRAN	17
2.5 Erweiterungen und Toolboxen	23
2.5.1 MATLAB-Compiler	23
2.5.2 <i>Optimization Toolbox</i>	23
2.5.3 <i>Symbolic Math Toolbox</i>	23
2.5.4 Weitere Toolboxen	24
3 Stöchiometrie, Thermodynamik, Reaktionskinetik: Drei Säulen der CRT	25
3.1 Stöchiometrische Gleichungen (Lineare Algebra)	25
3.1.1 Stöchiometrische Gleichungen aus der Element-Spezies- Matrix	25
3.1.2 Stöchiometrische Gleichungen aus der stöchiometrischen Matrix chemischer Reaktionen	34
3.1.3 Beispiel: Katalytische Dampfspaltung von Methan zu Synthesegas	38
3.2 Thermodynamische Gleichgewichte (Nichtlineare AGIs)	40

3.3	Reaktionskinetik (Gewöhnliche, auch steife DGls; DAGls)	43
3.3.1	Wichtige reaktionskinetische Größen und Beziehungen . .	43
3.3.2	Komplexe Reaktionssysteme (DGls)	45
3.3.2.1	Analytische Lösung	46
3.3.2.2	Numerische Lösung	46
3.3.2.3	SIMULINK-Block-Modell	52
3.3.2.4	SIMULINK-Modell mit <i>M-File S-Funktion</i>	54
3.3.2.5	SIMULINK-Modell mit <i>C MEX-File S-Funktion</i>	55
3.3.2.6	SIMULINK-Modell mit kompilierter <i>M-File S-Funktion</i>	62
3.3.3	Beispiel: Spaltung von Kohlenwasserstoffen	63
3.3.4	Zusammenfassung	67
3.4	Probleme und Lösungen	68
3.4.1	Wahl von Nichtschlüsselkomponenten (*)	68
3.4.2	Korrespondierende Manipulation von ESM und Spezies-Vektor (** bis ***; * wenn STOICH als Vorlage benutzt wird)	69
3.4.3	Erzeugung einer String-Matrix aus einem Reaktionsmechanismus (*)	70
3.4.4	Automatische Generierung der Reaktionsmatrix für einen Reaktionsmechanismus (Chemischer Compiler 1) (***)	70
3.4.5	Automatische Generierung der Geschwindigkeitsgleichungen für einen Elementarmechanismus (Chemischer Compiler 2) (**)	72
3.4.6	Wasserdampfsättiger (*)	73
3.4.7	Synthesegaszusammensetzung aus der Analyse des trockenen Produktgases (*)	75
3.4.8	Überprüfung eines Reaktionsgemisches am Reaktorausgang auf Gleichgewicht (*)	78
3.4.9	Gleichgewicht am Reaktorausgang (*)	81
3.4.10	Rechengeschwindigkeit mit vollen und sparsen Matrizen im Vergleich (*)	84
3.4.11	Molwärme eines realen Gases (*)	84
3.4.12	SIMULINK-Modell der Ethan-Spaltung mit <i>MATLAB-Fcn-Block</i> (**)	86
3.4.13	SIMULINK-Modell der Ethan-Spaltung mit <i>Fcn-Blöcken</i> (**, SE bedingt)	88
3.4.14	SIMULINK-Modell der Ethan-Spaltung mit <i>S-Funktion</i> (**)	90
3.4.15	Einfaches System aus nichtlinearen algebraischen und Differential-Gleichungen (DAE- oder DAGl-System) (*) .	96

4	Rührkesselreaktoren	101
4.1	Batch-Reaktor	103
4.1.1	Stoff- und Energiebilanzen	103
4.1.2	Beispiel: Zweistufiger Prozeß (DAGI-Systeme mit Unstetigkeiten)	104
4.2	Semibatch-Reaktor	111
4.2.1	Stoff- und Energiebilanzen	111
4.2.2	Beispiel: Reaktionskalorimeter	111
4.3	Konti-Reaktor	123
4.3.1	Stoff- und Energiebilanzen	123
4.3.2	Beispiel: Optimierung einer Lösungs-Copolymerisation in einer vierstufigen Rührkesselkaskade	123
4.4	Probleme und Lösungen	128
4.4.1	Lösung eines DAGI-Systems durch Kombination von DGI- und algebraischen Gleichungslösern (*)	128
4.4.2	Lösung eines reaktionskinetischen DAGI-Systems durch Überführung der Gleichgewichtsbeziehungen in kinetische Gleichungen (*)	131
4.4.3	Lösung eines DAGI-Systems über Differentiation der algebraischen Gleichungen (**)	133
4.4.4	Lösung des DAGI-Systems 4.4.1 mit einem DAGI-Löser von MATLAB ab Version 5.3	135
4.4.5	Lösung des DAGI-Systems 4.4.1 mit SIMULINK unter Verwendung eines externen DAGI-Lösers (***)	136
4.4.6	Einstellen der Regler am Reaktionskalorimeter für die Sulfonierung von Nitrobenzol (*, nicht SE)	148
4.4.7	Verlustleistung des Reaktionskalorimeters bei den Bedingungen des Problems 4.4.6 (*, nicht SE)	149
4.4.8	Sulfonierung von Nitrobenzol mit Oleum im leistungskompensierten Reaktionskalorimeter mit Differentialkühlung (*, nicht SE)	150
4.4.9	Wärmeübergangskoeffizienten im wandgekühlten Rührkessel durch Modellversuche im Reaktionskalorimeter (*, nicht SE)	153
4.4.10	Produktivitätssteigerung bei einer technischen Cyanethylierung (**)	153
4.4.11	Produktivitätssteigerung bei einer technischen Cyanethylierung (2): optimale Dosierzeit bei konstanter Dosierate (*)	158
4.4.12	CSTR – Chemostat (*)	158
4.4.13	Exotherme Reaktion im wandgekühlten CSTR (**)	160
4.4.14	Styrolpolymerisation im CSTR (***)	163
4.4.15	Einstellen der Zuströme in einem konzentrations-geregelten Differential-Kreislaufreaktor (**)	173
4.4.16	SIMULINK-Modell einer Destillationskolonne anhand der MESH-Gleichungen (***, nicht SE)	175

5	Festbettreaktor – einphasige Modelle	191
5.1	Noch einmal Stöchiometrie	191
5.2	FBR mit axialer Dispersion	199
5.2.1	Modellgleichungen und Diskussion der Randbedingungen	199
5.2.2	Schußverfahren und Diskretisierung	201
5.2.3	Mehrzielmethode (Bulirsch'sches Randwertproblem)	208
5.3	Instationärer FBR mit axialer Dispersion	213
5.4	FBR mit radialer Dispersion	214
5.4.1	Modellgleichungen und Randbedingungen	214
5.4.2	Methode der finiten Bilanzvolumina (Finite Volumenmethode)	215
5.4.3	Linienmethode	219
5.4.4	Orthogonale Kollokation	224
5.5	Probleme und Lösungen	226
5.5.1	Aufstellen der Stoffbilanzen für die katalytische Methanisierung im isothermen FBR (*)	226
5.5.2	Gleichgewichtsberechnungen durch Lösen von Reaktorgleichungen bei großer Verweilzeit (*)	227
5.5.3	Oxidation von o-Xylol im idealen FBR (*)	231
5.5.4	Radiale Temperaturprofile im FBR in parabolischer Näherung (*)	233
5.5.5	SIMULINK-Modell für den FBR mit beliebigen Reaktionen (***, nicht SE)	234
5.5.6	Einfluß axialer Dispersion (**)	239
5.5.7	Auswertung von Verweilzeitmessungen in einem Festbettreaktor (*)	242
5.5.8	Konzentrations- und Temperaturprofile im adiabatischen FBR mit axialer Dispersion (**)	245
5.5.9	FBR mit periodischer Strömungsumkehr (**)	249
5.5.10	Simulation eines FBR nach dem zweidimensionalen Modell mit Hilfe der FVM (**)	253
5.5.11	Lösung der Wärmeleitungsgleichung mit der Linienmethode (**)	258
5.5.12	Berechnung der Nullstellen für die orthogonalen Polynome nach Gl. 5.106 (***, nicht SE)	261
5.5.13	Berechnung der Matrizen A , B , Q und W für die orthogonale Kollokation (*)	262
5.5.14	Oxidation von o-Xylol nach dem zweidimensionalen Modell eines FBR (Orthogonale Kollokation) (***)	263
5.5.15	Oxidation von o-Xylol nach dem zweidimensionalen Modell eines FBR mit Hilfe von FEMLAB (nicht SE)	266

6	Festbettreaktor – zweiphasige Modelle	269
6.1	Transportlimitierungen zwischen den Phasen	269
6.2	Transportlimitierungen in der Feststoffphase	273
6.3	Probleme und Lösungen	279
6.3.1	Konzentrationsprofile zwischen Gas und Katalysatorkorn nach der Filmtheorie bei einer Reaktion vom Typ $A \rightarrow B + C$ mit Hilfe der SM-Gleichungen (*)	279
6.3.2	Konzentrationsdifferenzen zwischen Gas und Katalysator- korn nach der Filmtheorie bei einer Reaktion vom Typ $A \rightarrow B + C$ mit Hilfe der SM-Gleichungen (*)	281
6.3.3	Isothermer eindimensionaler FBR mit externem Stofftransportwiderstand (DAGI-System) (**, nicht SE)	284
6.3.4	Konzentrationsprofil und Nutzungsgrad für eine heterogen-katalytische Reaktion $A \rightarrow B$ erster Ordnung in A (**, SE bedingt)	289
6.3.5	Reaktion erster Ordnung $A \rightarrow B$ im isothermen FBR mit Porendiffusion (**).	292
6.3.6	Reaktion erster Ordnung $A \rightarrow B$ im porösen Katalysatorkorn nach dem Staub-Gas-Modell (**).	293
6.3.7	Reaktion $A \rightleftharpoons B + C$ mit LH-Kinetik im porösen Katalysatorkorn nach dem Staub-Gas-Modell (***)	296
6.3.8	Nichtisothermer Festbettreaktor mit Stofftransport- widerständen im Katalysatorkorn (***)	300
7	Planen und Auswerten reaktionskinetischer Experimente	305
7.1	Parameterschätzung	306
7.1.1	Regression	306
7.1.2	Parameterschätzung in Differentialgleichungen	308
7.1.2.1	Sensitivitätsgleichungen	308
7.1.2.2	Das „inverse“ Problem	309
7.1.3	Differenzieren und Integrieren von Meßwerten	310
7.2	Versuchsplanung	313
7.2.1	Parameterpräzisierung	313
7.2.2	Modellaufbau	314
7.2.3	Modelldiskriminierung	316
7.3	Probleme und Lösungen	317
7.3.1	Kinetische Untersuchungen im isothermen Batch-Reaktor (*)	317
7.3.2	Reaktionskinetische Auswertung einer q_r -Kurve im Reaktionskalorimeter (*)	321
7.3.3	Einzelreaktion erster Ordnung; Häufigkeitsfaktor und Aktivierungsenergie nach der Methode der kleinsten Quadrate (*)	323
7.3.4	Bewertung der Anpassung in Problem 7.3.3: Antwortfläche, Kovarianzmatrix und gemeinsamer Konfidenzbereich der Parameter (**).	325

7.3.5	Eigenwert-Eigenvektor-Zerlegung der Matrix N und Korrelationsmatrix der Parameter aus Problem 7.3.4 (*)	329
7.3.6	Parameterschätzung in DGLs (*)	329
7.3.7	Parameterschätzung in DGLs mit der Mehrzielmethode (***, nicht SE)	333
7.3.8	Differentielle Auswertung von Konzentrations-Zeit-Kurven mit Polynomen und Splines (*, SE nur mit Polynomen)	342
7.3.9	Differentielle Auswertung von Konzentrations-Zeit-Kurven mit ausgleichenden Splines (*, nicht SE)	347
7.3.10	Differentielle Auswertung von Konzentrations-Zeit-Kurven mit rationalen Funktionen (*)	350
7.3.11	Parameterschätzung in einer einfachen DGL mit der Himmelblau-Jones-Bischoff-Methode (*)	353
7.3.12	Parameterschätzung in der Bulirsch-DGL mit der Himmelblau-Jones-Bischoff-Methode (*)	354
7.3.13	Präzisierung der Geschwindigkeitskonstanten in einer Langmuir-Hinshelwood-Gleichung (**, nicht SE)	356
7.3.14	Modellaufbau (**)	359
7.3.15	Diskriminierung zwischen zwei Geschwindigkeitsgleichungen (**)	365
7.3.16	Differentielle Auswertung von axialen Konzentrations- und Temperaturmessungen in einem NH_3 -Versuchsreaktor (***, nicht SE)	369
A	Einige weitere Möglichkeiten von MATLAB	383
A.1	Menü-Führung	383
A.2	GUIs	384
A.3	Neuronale Netze – ein Beispiel für Toolboxen	385
A.4	CFD	389
	Literaturverzeichnis	395
	Index	398

Vorwort

Als Hochschullehrer für Technische Chemie hörte ich von MATLAB zum ersten Mal von einem Kollegen aus der Regelungstechnik. Ich halte das auch heute noch für bezeichnend: Obwohl MATLAB – wie das vorliegende Buch an Beispielen aus der Chemischen Reaktionstechnik zeigen wird – ein universelles Werkzeug der numerischen Mathematik ist, hat es sich auf bestimmten Gebieten schnell durchgesetzt, während es auf anderen höchstens eine Randrolle spielt. Für regelungstechnische Aufgaben z. B. gibt es eine ganze Reihe von Toolboxes, dagegen läßt sich für das Chemie-Ingenieur-Wesen höchstens ein entfernter Verwandter anführen, die *Chemometrics Toolbox*.

Nach kurzer Einarbeitungszeit setzte ich MATLAB in reaktionstechnischen Vorlesungen und Übungen ein, mit dem erhofften Erfolg: Meine fortgeschrittenen Studenten, vom Chemiestudium nicht gerade optimal auf wissenschaftliches Rechnen vorbereitet, scheiterten bis dahin regelmäßig bei dem Versuch, mühsam erarbeitete Lösungsmethoden in Programme einer höheren Programmiersprache umzusetzen, und ich hatte weder Zeit noch Lust, die Fehler in solchen Programmtorsos zu suchen. MATLAB zu lernen und einzusetzen erwies sich als sehr viel einfacher, so daß wir uns mehr auf die vorgelagerten Schritte konzentrieren konnten, also auf die mathematische Modellierung reaktionstechnischer Systeme, das Erkennen der mathematischen Struktur und die Auswahl geeigneter Lösungsalgorithmen.

Mit fortschreitender Übung ließen sich meine Studenten auch auf ein für sie völlig neues Gelände führen, auf die Programmierung in Blockschaltbildern, mit anderen Worten auf den Einsatz von SIMULINK, das wesentlich schnellere Simulationsrechnungen zuläßt.

So wurden MATLAB und SIMULINK zu unserem Standardwerkzeug für wissenschaftliches Rechnen; es wurde in Forschungsarbeiten und in Kooperationen mit Industriepartnern auf komplexe Probleme angesetzt und überzeugte.

Das vorliegende Buch möchte diese Erfahrungen weitergeben. Es ist kein herkömmliches einführendes Lehrbuch der Chemischen Reaktionstechnik (CRT), es vermittelt CRT anhand realistischer Probleme, zu deren Lösung man einen Computer benötigt. Im Vordergrund steht die Anwendung moderner numerischer Methoden, nicht die Vollständigkeit der fachlichen Inhalte; zu Beginn der einzelnen Kapitel werden die Grundlagen der CRT aber im Zusammenhang erweitert und vertieft. Im Gegensatz zu herkömmlichen Aufgabensammlungen, die sich überwiegend auf analytisch lösbare, einfache Übungen beschrän-

ken, konzentriert es sich auf die Computeranwendung; das geschieht mit Hilfe von MATLAB, einer weitverbreiteten, vielseitigen und mächtigen Software, die nicht nur bei einfachen Übungsaufgaben einsetzbar ist, sondern höheren Anforderungen genügt und den Übergang zu den Problemen der Praxis mitmachen kann. Eine Einführung in MATLAB wird nur soweit gegeben, wie man zum Einstieg benötigt, für alles weitere sind die Handbücher und insbesondere auch die Online-Hilfe bestens geeignet; weitere Informationen zu MATLAB gibt der vorletzte Abschnitt dieses Vorworts.

Das Buch ist also in erster Linie gedacht als eine Anleitung zum wissenschaftlichen Rechnen anhand typischer Probleme der CRT, die auf dem Computer bearbeitet werden können. Der Weg vom Problem zur Lösung verläuft über die Hauptstationen Modellierung, Simulation und Interpretation, die in vielfältiger Weise miteinander verwoben sind und nicht isoliert betrachtet werden sollten. In diesem Buch liegt das Hauptgewicht auf dem Simulationsteil, dem wissenschaftlichen Rechnen. Es werden (Muster-)Lösungen vorgestellt und gezeigt, wie MATLAB und SIMULINK als vielseitige Hilfsmittel dienen können, insbesondere auch in Verbindung mit einer symbolischen Programmiersprache wie MAPLE, das als Toolbox in MATLAB eingebunden werden kann, und unter geschicktem Einsatz der Zeichenkettenverarbeitung. Die Probleme sind möglichst realistisch gestaltet, es gibt einfache Probleme zur Übung oder zum Einstieg, solche mittleren Schwierigkeitsgrades und anspruchsvollere, aus denen – so steht zu hoffen – auch der erfahrene Reaktionstechniker Nutzen ziehen kann.

Die Beispiele dieses Buches sind über viele Jahre entstanden; sie stammen aus der akademischen Lehre, aus Kursen für die industrielle Praxis, aus Forschungsarbeiten und aus Kooperationen mit Partnern aus der Industrie.

Zur Organisation des Buches

Die Gliederung und den roten Faden bestimmt die CRT. Es wäre auch denkbar gewesen, die Möglichkeiten von MATLAB vorzustellen und anhand von reaktionstechnischen Beispielen zu demonstrieren – das aber würde dem Weg vom Problem zur Lösung zuwiderlaufen.

In den einzelnen Kapiteln werden stets die reaktionstechnischen Grundlagen erläutert, auf denen eine Lösung der Probleme aufbauen kann; die gedrängte Darstellung der Grundlagen ist zur Wiederholung und zur Vertiefung gedacht, erweitert aber die in den herkömmlichen Lehrbüchern behandelten Gebiete. Ich hoffe, daß an manchen Stellen die Ausführungen zur Klärung von Begriffen oder Sachverhalten beitragen können, die in einschlägigen Werken unvollständig oder gar irreführend behandelt werden.

Von den klassischen Gebieten der CRT fehlen die Mehrphasenreaktoren aus folgendem Grund: Das Zusammenspiel von chemischer Reaktion und physikalischen Transportprozessen wird, was seine Anforderungen an wissenschaftliches Rechnen betrifft, ausreichend im Kapitel über zweiphasige Modelle von Festbettreaktoren behandelt. Darüber hinausgehende, spezifische Anforderungen von Mehrphasensystemen an Rechentechniken gibt es durchaus, wie etwa die Lösung von Integro-Differentialgleichungen, die bei Populationsmodellen auf-

treten, aber auf diesem Feld hat MATLAB so wenig zu bieten wie andere Programmpakete. Anderen Gebieten, wie z. B. dem Verweilzeitverhalten von Reaktoren, sind auch keine eigenen Kapitel gewidmet; sie werden aber in Form von Problemen an geeigneter Stelle gestreift.

Die Probleme selbst sind in eigenen Abschnitten zusammengefaßt, den Kapitelthemen wie Stöchiometrie, Thermodynamik, Kinetik, Reaktoren usw. zugeordnet, aber zwangsläufig häufig übergreifend. Die meisten Kapitel enthalten ein besonders umfangreiches Problem oder Beispiel, das in den Überschriften zusammen mit seinen Hauptinhalten reaktionstechnischer oder rechnerischer Art ausgewiesen ist. Der erfahrene Reaktionstechniker kann sich unmittelbar mit solchen Problemen beschäftigen, um die Möglichkeiten von MATLAB und SIMULINK kennenzulernen. Für weniger Erfahrene sind die meisten großen Probleme in eine Reihe von Teilproblemen aufgespaltet, an denen sie sich zunächst versuchen können. Zur Orientierung ist der Schwierigkeitsgrad angegeben: * steht für leicht, ** für mittelschwer und *** für schwierig; die Einschätzungen sind relativ und subjektiv; je nach Vorkenntnissen kann der Leser¹ anders urteilen.

Es werden einige numerische Verfahren angewendet, die nicht unmittelbar zum Umfang von MATLAB und seinen Toolboxen gehören, aber mit ihrer Hilfe besonders kompakt und einfach eingesetzt werden können, wie z. B. die orthogonale Kollokation oder die finite Volumenmethode zur Lösung von partiellen Differentialgleichungen. Damit der Leser auf sie zurückgreifen kann, sind diese Methoden in eigenen Abschnitten kochbuchartig zusammengestellt.

Für die Beispiele und die Lösungen der Probleme wurden MATLAB-Programme erstellt überwiegend in Form von M-Files oder SIMULINK-Modellen (MDL-Files), die auf der Begleit-CD zu finden sind; ihre Namen sind im laufenden Text in KAPITALCHEN gesetzt. Hier werden (Muster-)Lösungen vorgestellt, die keineswegs unübertrefflich sind. Meistens sind es Ad-hoc-Lösungen, so wie man sie zum Bearbeiten eigener Probleme entwickelt, nicht in erster Linie für andere Benutzer gedacht, insbesondere auch nicht defensiv, d. h. nicht fehlerabweisend ausgebaut. Aber: Verständlich sollten sie in jedem Fall sein.

Um die bedeutende Rolle des Experiments in der Reaktionstechnik wenigstens anklingen zu lassen in einem Buch, das der mathematischen Modellierung und dem wissenschaftlichen Rechnen gewidmet ist, werden gelegentlich Experimente anhand von SIMULINK-Modellen simuliert, z. B. mit einem Reaktionskalorimeter im Abschnitt 4.2.2 oder mit einem Festbettreaktor in Abschnitt 3.1.3, an anderen Stellen werden schlichte Exe-Files verwendet.

In den einzelnen Kapiteln werden die komfortablen Möglichkeiten von MATLAB und SIMULINK zur Ein- und Ausgabe von Daten sowie zur Visualisierung

¹Liebe Leserin, ich benutze hier und im folgenden den Oberbegriff „Leser“. Der männliche Leser mag sich beschweren, daß für ihn keine eigene Bezeichnung gebräuchlich ist, die Leserin, daß sie im Oberbegriff eingeschlossen ist und ihre eigene Bezeichnung nicht genannt wird. Wer hier ein Dilemma sieht, kann mich trotzdem nicht zu Verrenkungen veranlassen, etwa den Text durch „der Leser oder die Leserin“ (in welcher Reihenfolge?) aufzublähen oder gar meine Mutter/Vater-Sprache durch ungeheuerliche Schrägstrichkonstruktionen (Marke: Der/die Leser(in)) zu verhunzen; auch mein privater Ausweg – „das LeserIn-Mensch“ – ist hier sicherlich nicht angebracht. Kurz und gut: Leser beiderlei Geschlechts sind mir willkommen.

vielfältig aufgezeigt. Darüber hinaus bietet MATLAB eine Reihe weiterer Hilfsmittel. Dazu gehört vor allem das *Notebook*, die Verbindung von WinWord und MATLAB: So kann man Dokumentation und Rechnen bequem verknüpfen. Mit MATLAB kann man vollständige GUIs erzeugen, wie eindrucksvoll MATLABs Demos zeigen. Das mag über eine gewisse Spielerei hinaus vorteilhaft sein, z. B. wenn man denselben Rechenablauf häufig benötigt, oder wenn man Mitarbeiter, die nicht mit MATLAB vertraut sind, für Routinerechnungen einsetzen möchte. Auf solche peripheren Möglichkeiten weist beispielhaft ein Kapitel im Anhang hin.

Es ist in diesem Vorwort vielleicht schon deutlich geworden, daß dieses Buch auch und vor allem als professionelles Nachschlagewerk für typische Aufgaben der reaktionstechnischen Modellierung gedacht ist. Um schnelles und sicheres Finden von Lösungen oder Lösungsmöglichkeiten bei Bedarf möglichst einfach zu machen, sind im Inhaltsverzeichnis sämtliche Aufgaben (Probleme) vermerkt; auch beim Stichwortverzeichnis (Index) wurde darauf geachtet, wichtige Begriffe und Methoden herauszustellen.

Dieses Buch ist in mehr als dreijähriger Arbeit entstanden. In dieser Zeit gab es eine Reihe von *updates* für MATLAB und seine Toolboxen; ein besonders großer Schritt war der Wechsel von der Version 4.x auf 5.x. Wo es sinnvoll erschien, wurden Programme auf den neuesten Stand gebracht. Nicht alle neuen Möglichkeiten wurden im Nachhinein eingebaut: Zum Beispiel wird auffallen, daß die Art der Beschriftung von Abbildungen nicht einheitlich ist. Das liegt zum Teil daran, daß die Verwendung von Sonderzeichen oder von TEX-Elementen erst in späteren Versionen möglich war – die Verständlichkeit der Abbildungen ist davon nicht berührt. Gelegentlich wurden auch ältere Lösungswege beibehalten, weil sie zeigen sollen, wie man sich in MATLAB geschickt helfen kann, wenn die vorhandenen Werkzeuge einmal nicht ausreichen oder nicht optimal erscheinen. Dazu gehört etwa das programmierte Ausblenden von Vektorkomponenten mit *Demux*-Blöcken in SIMULINK, vor allem bei dem Modell eines Reaktionskalorimeters im Abschnitt 4.2.2, das man in der SIMULINK-Version 2 zum Teil einfacher mit *Select*-Blöcken erreichen könnte, aber eben nur zum Teil: Das Ausblenden einzelner Vektorelemente ist einfacher, aber nicht das Ausblenden von Teil-Vektoren; erst mit der Version 5.3 kam ein *Bus-Selector*-Block als entsprechendes Hilfsmittel. So blieb die alte Lösung an manchen Stellen erhalten, weil sie einfacher oder zumindest gleichwertig war, oder sie dient zur Demonstration.

Zum Gebrauch des Buches

Die Organisation eines Buches wird natürlich nachhaltig davon beeinflußt, wie sich ein Autor den Gebrauch seines Buches vorstellt. Deshalb enthält der vorige Abschnitt schon eine Reihe von Hinweisen in diese Richtung, die nun im Zusammenhang noch einmal aufgegriffen und erweitert werden sollen.

Grundkenntnisse in MATLAB und in numerischen Methoden vermittelt dieses Buch nicht systematisch. Der Leser sollte sie mitbringen oder sich (parallel) aneignen, z. B. anhand von Kapitel 3, und dann im weiteren Verlauf vertiefen.

Auch reaktionstechnische Vorkenntnisse sind gefragt; sie können anhand der auf die Probleme ausgerichteten Darstellung in den einzelnen Kapiteln wiederholt und vertieft werden.

Die meisten Probleme kann man allein mit Hilfe der Studentenausgaben von MATLAB und SIMULINK bearbeiten, für andere benötigt man die professionellen Ausgaben, und für einige zusätzlich auch die eine oder andere Toolbox. Obwohl die Datenbeschaffung ein ganz wichtiger Bestandteil der Modellierung und Simulation ist, werden die meisten der benötigten Daten angegeben.

Selbst wer noch keinen Zugang zu MATLAB hat, kann das Buch mit Nutzen studieren: Die Aufbereitung der Probleme und Lösungswege bis zur Simulation ist in vielen Fällen weitgehend unabhängig von der Programmierung; man könnte auch in einer der Hochsprachen programmieren oder andere Simulationswerkzeuge benutzen – der eigene Arbeitsaufwand würde allerdings beträchtlich steigen.

Dem Leser kann nicht dringend genug empfohlen werden, zunächst eine eigene Lösung zu versuchen. Erst wenn eine eigene Lösung erarbeitet ist, oder wenn man trotz aller Anstrengung festsitzt, sollte man sich den Musterlösungen zuwenden. Die drei Hauptschritte werden besprochen: Modellierung, Simulation, Interpretation und gegebenenfalls ihre Interdependenzen. Bei den einfachen Beispielen, die auf Ungeübte zugeschnitten sind, wird ausführlicher auf den Lösungsweg eingegangen, insbesondere in den ersten Kapiteln; bei den komplizierteren Aufgaben wird oft nur die Lösungsstruktur skizziert. Zur Diskussion der Simulationsschritte dienen kommentierte Programmausdrucke. (Die Programme auf der Begleit-CD sind aus mehreren Gründen unkommentiert; einige liegen nur in kompilierter Form vor.) Dem fortgeschrittenen Leser wird gelegentlich zugemutet, Variablenamen im Programm entsprechenden Formelzeichen zuzuordnen oder ganze Programmteile ohne weiter Erläuterungen zu verstehen, besonders dann, wenn auf ähnliche Aufgaben verwiesen werden kann.

Die Lösungsvorschläge kann man zwar zur Not auch ohne Computer studieren, aber so bei weitem nicht ausschöpfen. Sie sind zweckmäßig vor einem eingeschalteten Computer zu lesen, auf dem die CD-Programme gestartet und die Ergebnisse der Simulationen – über die schwarz-weiß Grafiken im Buch hinaus – während oder nach dem Simulationslauf visualisiert werden können. Dabei soll man natürlich die Möglichkeit ausgiebig nutzen, Parameter zu ändern und die Auswirkungen zu studieren.

Besondere Bedeutung kommt der Interpretation der Ergebnisse zu. Vor allem: Scheinen sie frei von numerischen Artefakten und sind sie physikalisch sinnvoll? Was ein Computer ausgibt, muß geprüft werden. Das kann auf verschiedenen Wegen geschehen: Ein wichtiger und vom reinem Anwender numerischer Methoden noch am leichtesten zu begehender Weg ist die Vereinfachung seiner Gleichungen auf analytisch lösbare, für die dann die symbolische Toolbox mit Vorteil eingesetzt werden kann. Auch dafür enthält dieses Buch Beispiele.

Informationen über MATLAB und SIMULINK

Wer MATLAB noch nicht kennt und auch keinen Zugang hat (z. B. über ein Rechenzentrum), kann sich im Internet unter <http://www.mathworks.com> umfassend informieren. Von den *demos* könnte man sich zunächst *Basic Matrix Operations in MATLAB* ansehen: Man erreicht sie am besten unter dem *quick link* 'MATLAB in Education'. Die anderen *demos* sind wie die SIMULINK-Demos zwar interessant und beeindruckend, der Anfänger wird aber Mühe haben, sie im einzelnen zu verstehen. Unter *MATLAB in Education* findet man auch den Punkt *Select resources for your particular area of interest, and GO*: Wählt man hier *Chemical Engineering*, so erreicht man *Teaching examples*, von denen ich *Numerical Methods. CENG303: MATLAB for Chemical Engineers, Rice University* empfehlen möchte; es ist ein Kurs, der in kleinen Schritten in MATLAB einführt und am Bildschirm verfolgt werden kann.

Einfacher ist es, sich die Studentenausgaben von MATLAB und SIMULINK zu besorgen (je 120,- bis 130,- DM). Preiswerter wird man wohl kaum an ein so vielseitiges Instrumentarium für das wissenschaftliche Rechnen gelangen; es hat einige Einschränkungen gegenüber den Vollversionen, die aber für das Lernen unerheblich sind; dafür enthält es (MATLAB) zusätzlich das *Notebook* sowie drei Toolboxen, darunter die so wichtige *Symbolic Math Toolbox*, eine Auswahl aus MAPLE. Mit diesen Studentenausgaben kann man fast alle Aufgaben dieses Buches bearbeiten.

Begleitprogramme

Die Begleit-CD enthält zwei Verzeichnisse: In „creums“ sind die Programme für die Vollversionen von MATLAB und SIMULINK, in „creums_se“ die Programme für die Studentenausgaben. **Vorsicht!** Trotz gleichen Namens unterscheiden sich viele Programme aus diesen beiden Verzeichnissen (MATLAB-Programme und SIMULINK-Modelle) – sie sind also **nicht austauschbar**. Am besten kopiert man das zutreffende Verzeichnis in ein neues Unterverzeichnis von MATLAB; wie man dieses Unterverzeichnis zum aktuellen macht oder wie man einen Pfad legt, steht im Handbuch.

Notation

0.1 Schriftauszeichnung

Mit wenigen, angemerkten Ausnahmen bezeichnen kleine fette Buchstaben Vektoren (\mathbf{x}), große fette Buchstaben Matrizen (\mathbf{M}); Buchstabenkombinationen als Matrix- oder Vektorname sind zusätzlich unterstrichen (ESM).

Fremdsprachige Ausdrücke sind mit wenigen Ausnahmen (MATLAB) kursiv gesetzt (*backslash*), ebenso Programmnamen und sonstige Bezeichnungen in MATLAB und SIMULINK (*S-function*); Fachausdrücke (online) werden normal geschrieben, sofern sie im Duden aufgeführt sind.

Variable sind im laufenden Text kursiv gesetzt (*h*), Programmvariable steil (\mathbf{x} zu).

Platzhalter für chemische Verbindungen (A, B usw.) stehen im Text normal steil, in mathematischen, stöchiometrischen und Reaktionsgleichungen kursiv.

Anweisungen (Befehle), auch in Programmausdrucken, sind in serifenloser Schrift gesetzt ($\text{inv}(\mathbf{M})$), Ausgaben im MATLAB-Fenster in *Typewriter*-Schriftart (`ans = 5.87`). Ausnahmen sind einige aus Vorlagen entstandene *C MEX-file S-functions*, die statt in serifenloser Schrift in *Typewriter*-Schriftart gesetzt sind: Die fettgedruckten eigenen Einträge heben sich so besser ab.

Die Namen der beigegeführten Programme und sonstiger Files sind in Kapitälchen gesetzt (STOICH).

Kommentare in Programmausdrucken werden mit % eingeleitet und, wenn sie über eine Zeile gehen, auch mit % abgeschlossen.

0.2 Symbolverzeichnis

Im folgenden werden typische Einheiten angegeben; in den Rechnungen können durchaus auch andere Maßsysteme verwendet werden.

Symbol		Einheit
a	spezifische Austauschfläche	m^2/m^3
B_0	Permeabilität	m^2
Bi	Biot-Zahl	
Bo	Bodenstein-Zahl	
C_p	Molwärme	$\text{kJ}/(\text{kmol K})$
c	Konzentration	kmol/m^3
c_p	spezifische Wärme	$\text{kJ}/(\text{kg K})$
D	Diffusions-(Dispersions-)koeffizient	m^2/s
Da	Damköhler-Zahl	
d	Durchmesser	m
E	Aktivierungsenergie	kJ/kmol
E	Anzahl Elemente	
<u>ESM</u>	Element-Spezies-Matrix	
F	Fläche	m^2
f	Fläche	m^2
G	freie Enthalpie	kJ/kmol
H	Enthalpie	kJ/kmol
H	Hessesche Matrix	
h	Wärmedurchgangskoeffizient	$\text{kW}/(\text{m}^2 \text{K})$
J	Jacobi-Matrix	
K	Gleichgewichtskonstante	versch. Einh.
K_0	Knudsenkoeffizient	m
k	Geschwindigkeitskonstante	versch. Einh.
k_0	Häufigkeitsfaktor	versch. Einh.
L	Länge	m
l	Längenkoordinate	m
m	Masse	kg
\dot{m}	Massestrom	kg/s
N	Matrix der stöchiometrischen Koeffizienten ν_{ij}	
N	Näherung für Hessesche Matrix	
N	Molstrom	kmol/s
N	Molstromdichte	$\text{kmol}/(\text{m}^2 \text{s})$
N	Anzahl Kollokationspunkte	
N	Anzahl Komponenten	
<u>NR</u>	Reaktionsmatrix	
<u>NS</u>	stöchiometrische Matrix	
n	Molzahl	kmol

n_i	Reaktionsordnung	
Pe	Peclet-Zahl	
p	Druck	Pa
q	Wärmestrom	kW
R	Gaskonstante	$\text{kJ}/(\text{kmol K})$
R	individuelle Reaktions- geschwindigkeit, Produktions- oder Verbrauchsgeschwindigkeit	$\text{kmol}/(\text{m}^3 \text{ s})$
R	Anzahl Reaktionen	
Re	Reynolds-Zahl	
r	Reaktionsgeschwindigkeit	$\text{kmol}/(\text{m}^3 \text{ s})$
r	radiale Koordinate	m
S	Anzahl der unabhängigen stöchiometrischen Gleichungen	
Sc	Schmidt-Zahl	
SEM	Spezies-Element-Matrix	
Sh	Sherwod-Zahl	
St	Stanton-Zahl	
s	Schätzwert für Standardabweichung	
s	Zustandsvariable	
s^T	$\Sigma \nu_{i1}, \dots, \Sigma \nu_{iR}$	
T	Temperatur	K
t	Zeit	s
u	Strömungsgeschwindigkeit	m/s
V	Volumen	m^3
\mathbf{V}	Kovarianz-Matrix	
v	Volumenstrom	m^3/s
X	Umsatz	
x	Molanteil	
z	Ortskoordinate	m

Griechische Buchstaben

α	Wärmeübergangskoeffizient	$\text{kW}/(\text{m}^2 \text{ K})$
β	Stoffübergangskoeffizient	$\text{m}^3/(\text{m}^2 \text{ s})$
δ	Grenzschichtdicke	m
ε	relativer Volumenanteil; Porosität	
η	dynamische Zähigkeit	$\text{kg}/(\text{m s})$
θ	Parameter	
Λ	mittlere freie Weglänge	m
λ	Wärmeleitfähigkeitskoeffizient	$\text{kW}/(\text{m K})$
ν	stöchiometrischer Koeffizient	
ξ	Reaktionslaufzahl, (Reaktionsvariable)	kmol/s

XXII Notation

ρ	Dichte	kg/m ³
σ	Standardabweichung	
τ	Raumzeit, Verweilzeit	s
τ	Labyrinthfaktor	
Φ	Zielfunktion	versch. Einh.
ϕ	Thiele-Modul; allgemeine Variable	

Index oben

0	Eingang; Zulauf; Bezug; Standardwert
D	Diffusion
e	Ausgang; Ablauf
eff	effektiv
s	Sollwert; Oberfläche (<i>surface</i>)
v	Volumen (<i>volume</i>)
*	Schätz-(Regressions-)Wert

Index unten

a	axial
ad	adiabatisch
ax	axial
c	Kühlung
e	Gleichgewicht (<i>equilibrium</i>)
g	Gas (<i>gas</i>)
h	Wärme (<i>heat</i>)
i	Komponente i
j	Reaktion j ; Reaktormantel (<i>jacket</i>)
K	Kreislauf
Kn	Knudsen
M	Mantel; Molmasse
m	Masse (<i>mass</i>)
N	Anzahl Komponenten
p	Partikel (<i>particle</i>)
R	Reaktion; Reaktionsvolumen; Anzahl Reaktionen
r	Reaktion; Reaktor; radial
S	Anzahl Schlüsselkomponenten
s	stationär
t	Rohr (<i>tube</i>)
U	Umgebung
v	Volumen (<i>volume</i>)
w	Wand (<i>wall</i>)

0.3 Abkürzungsverzeichnis

BR	Batch-Reaktor
CFD	<i>Computational Fluid Dynamics</i>
CRE	<i>Chemical Reaction Engineering</i>
CRT	Chemische Reaktionstechnik
CSTR	<i>Continuous Stirred Tank Reactor</i>
DAE	<i>Differential-Algebraic Equation</i>
DAGl	Differential-Algebraische Gleichung
DGl	Differentialgleichung
ESM	Element-Spezies-Matrix
FBR	Festbettreaktor
HJB	Himmelblau, Jones, Bischoff
LSM	<i>Least-squares-Methode</i>
MSR	Messen Steuern Regeln
MWR	<i>Method of Weighted Residues</i>
NAG	<i>Numerical Algorithm Group</i>
NDIR	<i>Non-Dispersive Infra-Red</i>
OK	Orthogonale Kollokation
PDGl	Partielle Differentialgleichung
RB	Randbedingung
RTW	Real Time Workshop
R _v	Reaktionsvolumen
SBR	Semibatch-Reaktor
SE	<i>Student Edition</i>
SGM	Staub-Gas-Modell
SM	Stefan-Maxwell
TDMA	<i>Tridiagonal-Matrix Algorithm</i>
UP	Unterprogramm

Aus technischen Gründen bleibt diese Seite leer

Kapitel 1

Einleitung

Muß ein Reaktionstechniker die numerische Mathematik beherrschen?

Die Chemische Reaktionstechnik (CRT) wendet die Erhaltungssätze für Stoff, Energie und Impuls auf chemisch reagierende Systeme an, speziell auf Prozesse der chemischen Industrie oder verwandter Industrien. Die mathematische Modellierung solcher Prozesse führt in der Regel zu stark nichtlinearen Differential- und algebraischen Gleichungen, vor allem, weil die Geschwindigkeit chemischer Reaktionen nach dem Arrhenius-Gesetz exponentiell von der Temperatur abhängt. Gegenüber anderen naturwissenschaftlich-technischen Disziplinen kann man es geradezu als Besonderheit der CRT herausstellen, daß ihre Modellgleichungen in der Regel diese so stark nichtlineare Abhängigkeit von einer Zustandsgröße enthalten. Entsprechend ist man in der CRT bei der Lösung der Modellgleichungen, bei der Simulation, im besonderen Maße auf effiziente numerische Methoden angewiesen.

Um die einleitende Frage aufzugreifen: Muß deswegen der Reaktionstechniker auch ein numerischer Mathematiker sein? Ich meine: nein; seine Aufgaben sind die Modellierung und die Interpretation der Simulationsergebnisse; numerische Methoden gehören zu seinem Handwerkszeug. Ohne gutes Handwerkszeug kann man allerdings nicht adäquat arbeiten, und man muß lernen, mit ihm umzugehen. Für einfache Probleme, für Übungsaufgaben in der Ausbildung oder zum Einstieg in bestimmte Teilgebiete, genügen häufig einfache Werkzeuge: Softwarepakete wie ISIM, MATHCAD oder POLYMATH gehören dazu, um nur einige zu nennen. Für große, komplexe Aufgaben in der Praxis mit ihren Tausenden oder gar Zehntausenden von Modellgleichungen, die meist weit über den Bereich der CRT hinausreichen, stehen leistungsfähige Pakete zur Verfügung, z. B. SPEEDUP oder DIVA. Solche Software bedarf allerdings langer Einarbeitungszeit und andauernden Umgangs. Für die professionelle tägliche Arbeit des Reaktionstechnikers ist etwas anderes vonnöten: ein Instrumentarium, das zugleich mächtig, vielseitig, flexibel und offen ist, nicht so starr wie die großen Softwarepakete, das mehr Raum läßt für ungewöhnliche, kreative Lösungsansätze, und trotzdem einfach zu erlernen und zu bedienen ist, benutzerfreundlich, wie man zu sagen pflegt. Ein solches Werkzeug ist MATLAB mit

seinen Toolboxen und Erweiterungen, von denen SIMULINK eine besondere Rolle spielt. Das exemplarisch aufzuzeigen ist Anliegen dieses Buches.

MATLAB hilft uns, unsere Modellgleichungen zu lösen und numerische Ergebnisse zu bekommen, was wir dem allgemeinen Sprachgebrauch folgend kurz als Simulation bezeichnen. Vor der Simulation steht die Modellierung unseres Problems, also die Überführung in mathematische Gleichungen; nach der Simulation sollten wir die Ergebnisse interpretieren. Über diese drei Schritte – Modellierung, Simulation und Interpretation – könnte man ganze Bücher schreiben, und sie sind geschrieben worden. Alleine schon die Gewichtung dieser drei Schritte würde uns in tiefe Wasser führen; es steht aber außer Frage: Bei der Simulation kann nichts Besseres herauskommen, als man in die Modellierung hineingesteckt hat. Die Simulation ist unser Hauptthema, aber wir können, dürfen und wollen die beiden anderen Schritte nicht außer acht lassen.

Wenn wir mit unserer Programmsammlung, hier also mit MATLAB, einigermaßen vertraut sind, werden wir in vielen Fällen geeignete Lösungsverfahren finden – vorausgesetzt, wir haben die mathematische Struktur unseres Modells richtig erkannt. Meistens gibt es mehrere Lösungswege, und sie werden nicht alle gleich bequem zu begehen sein. Gelegentlich fehlen bekannte Lösungsmethoden in unserer Programmsammlung, und wir müssen versuchen, mit den vorhandenen zu improvisieren. Mit anderen Worten: Die Auswahl eines Lösungsverfahrens erfordert häufig einige Überlegung.

Betrachten wir ein Beispiel zu diesen Vorbemerkungen, das in der rechnerischen Ausführung schon sehr weit führt:

Stellen Sie sich vor, wir haben ein detailliertes Modell für eine Destillationskolonne (mit oder ohne chemische Reaktion); die Bilanzgleichungen sind von der allgemeinen Form:

$$\frac{ds}{dt} = f(\mathbf{s}, \mathbf{x}, \mathbf{u}, t) \quad (1.1)$$

$$0 = \mathbf{g}(\mathbf{s}, \mathbf{x}, t) \quad (1.2)$$

Dabei sind \mathbf{s} die Zustandsvariablen, z. B. Molanteile und Temperatur, \mathbf{x} vorgegebene Größen und \mathbf{u} z. B. Molströme von Dampf und Flüssigkeit, die sich so einstellen müssen, daß die algebraischen Beziehungen erfüllt, z. B. Phasengleichgewichte eingestellt sind. Im Abschnitt 4.4.16 können Sie solche Bilanzen detailliert sehen. Wie löst man ein solches DAGE-System? Eine naheliegende Möglichkeit: Man iteriert mit Hilfe eines Optimierprogramms die $u(t)$ -Variablen über hinreichend klein gewählte t -Abschnitte, bis die algebraischen Gleichungen jeweils erfüllt sind, entweder von Boden zu Boden, oder über die gesamte Kolonne. Beides führt zu unerträglich langen Rechenzeiten. Ein DAGE-Löser wäre sehr vorteilhaft, und wir werden später sehen, wie wir externe Programme in SIMULINK implementieren können, aber: Man müßte zuvor den Index bestimmen – zumindest bei den meisten bis heute verfügbaren Programmen –, und es wird sich herausstellen, daß der Index größer als eins ist und damit eine Routineanwendung des DAGE-Lösers nicht in Frage kommt. MATLAB (nicht SIMULINK!) enthält erst seit der Version 5.3 eingeschränkte Möglichkeiten zur

Lösung von DAGs vom Index 1. Für MATLAB oder SIMULINK läßt sich die Indexbestimmung aber in ein Lösungsverfahren überführen, das nur mit DGLs zu tun hat und daher sehr schnell arbeitet. Wir betrachten das an einem sehr weit abgespeckten formalen Analogon, das eine analytische Lösung hat:

$$\frac{ds}{dt} = u \quad (1.3)$$

$$g = s * (t + 1) - 1 = 0 \quad (1.4)$$

Die analytische Lösung liegt auf der Hand:

$$s = \frac{1}{t + 1}, \quad u = -\frac{1}{(t + 1)^2} \quad (1.5)$$

Wie können wir vorgehen? Die Differentiation von g liefert uns:

$$\frac{dg}{dt} = (t + 1)\frac{ds}{dt} + s = 0 \quad (1.6)$$

Wir setzen den Ausdruck für $\frac{ds}{dt}$ ein und differenzieren erneut:

$$g_2 = (t + 1)u + s = 0 \quad (1.7)$$

$$\frac{dg_2}{dt} = (t + 1)\frac{du}{dt} + u + \frac{ds}{dt} = 0 \quad (1.8)$$

Damit erhalten wir eine DGL für u . Die konsistenten Anfangsbedingungen für das DGL-System sind:

$$t = 0 : s = 1, u = -1 \quad (1.9)$$

Die Anfangsbedingung für u folgt aus g_2 .

Der Leser möge sich anhand dieses einfachen Beispiels überzeugen, wie erheblich die Unterschiede in den Rechenzeiten für das erste Verfahren, die Iteration, und für die Lösung des DGL-Systems (1.3), (1.8) und (1.9) sind; dann wird er die Mühe des zweimaligen Differenzierens gering achten, zumal ihm bei komplizierteren Gleichungen die *Symbolic Math Toolbox* hilfreich zur Seite steht, und die Indexprüfung für einen DAGL-Löser es ebenfalls erfordern würde. (Das DGL-System ist im SIMULINK-Modell `DAGL_DGL.MDL` implementiert.)

Eine iterative Lösung auf dem skizzierten Weg ist nicht nur zeitraubend, was die Rechengeschwindigkeit betrifft, sondern erfordert auch umfangreiche Tests hinsichtlich der Zeitdiskretisierung sowie der Einstellungen der zahlreichen Optionen sowohl beim Optimierverfahren als auch beim DGL-Löser. Versuchen Sie

es selbst, bevor sie auf den Lösungsvorschlag zurückgreifen! (Der Lösungsvorschlag ist in dem Programm DAGL_STEU mit den UPs DAGL_OPT und DAGL enthalten.) Es sei noch angemerkt: Die Aufgabe läßt sich auch als Variationsproblem behandeln: u ist dann als Steuervektor zu verstehen, der eine geeignete Zielfunktion optimiert, hier z. B. $(g(u, t))^2$ minimiert. Bei nichtlinearen Systemen dürfte aber die Bestimmung der optimalen u -Trajektorie, z. B. durch das Verfahren der Steuervektor-Iteration, ebenfalls zu langen Rechenzeiten führen; ich habe das nicht ausprobiert, aber Sie könnten es versuchen. Anleitung finden Sie z. B. in einem Buch über Regelungstechnik ([1]). In diesem Zusammenhang sei angemerkt: Regelungstechnik und Chemische Reaktionstechnik lassen sich recht zwanglos unter dem Dach der Systemtheorie ansiedeln, ihre Methoden weisen sehr viele Ähnlichkeiten auf; diese Verwandtschaft sollte viel mehr in Anspruch genommen werden!

Wir sehen also, die Auswahl von Lösungswegen für die Simulation erfordert Erfahrung und Übung, auch bei einer so benutzerfreundlichen Programmsammlung wie MATLAB, und dazu sollen die folgenden Kapitel verhelfen.