INAUGURAL-DISSERTATION
zur
Erlangung der Doktorwürde
der
Naturwissenschaftlich-Mathematischen Gesamtfakultät
der
Ruprecht-Karls-Universität Heidelberg

vorgelegt von
Diplom-Mathematiker Stefan Körkel
aus Heidelberg

Tag der mündlichen Prüfung: 8. Oktober 2002

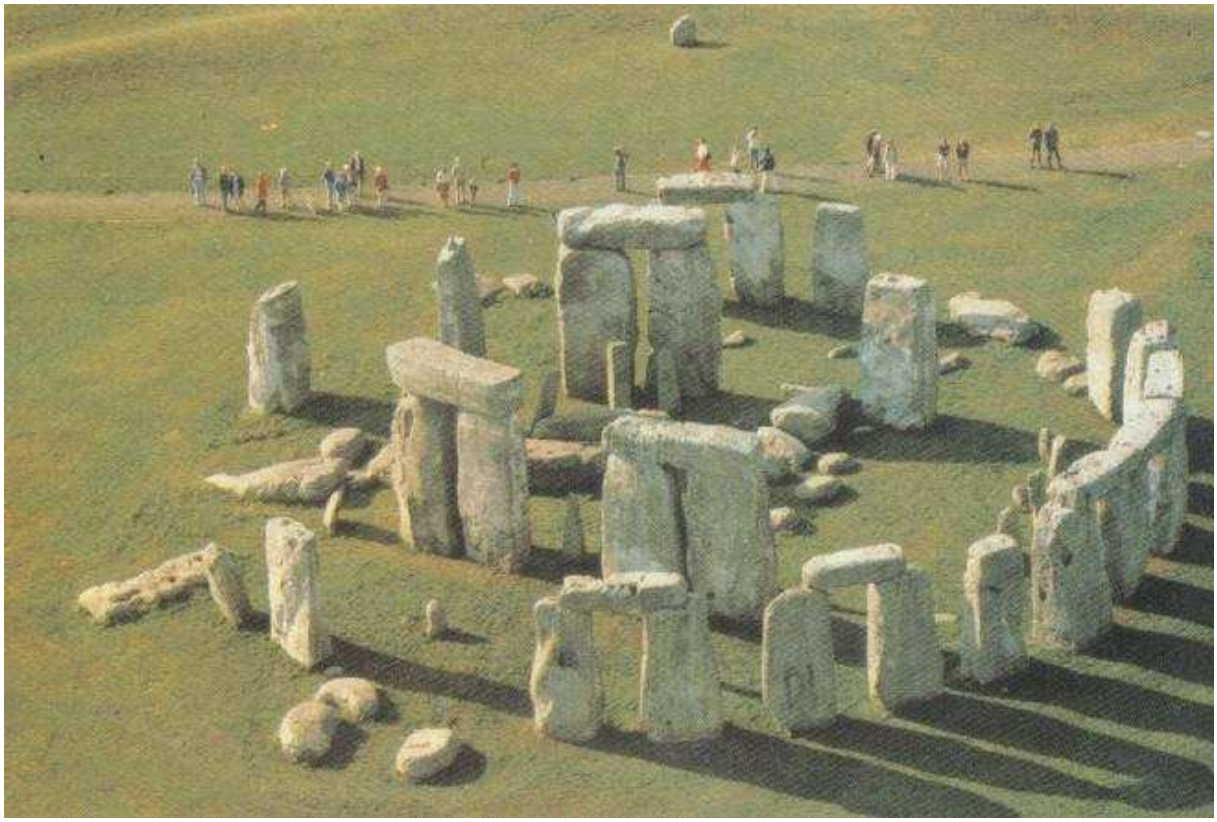
Numerische Methoden
für Optimale Versuchsplanungsprobleme
bei nichtlinearen DAE-Modellen

Gutachter:

Prof. Dr. Dr. h. c. Hans Georg Bock
Prof. Dr. Gerhard Reinelt

Stefan Körkel

Numerische Methoden für Optimale Versuchsplanungsprobleme bei nichtlinearen DAE-Modellen



Interdisziplinäres Zentrum
für Wissenschaftliches Rechnen
der Universität Heidelberg

Zur Abbildung auf der Titelseite:

Stonehenge ist das berühmteste Megalithmonument Englands und das bedeutendste prähistorische Bauwerk Europas. Einer Theorie zufolge könnte es dazu benutzt worden sein, die Sommer- und Wintersonnenwende, die Frühlings- und Herbsttagundnachtgleiche sowie Sonnen- und Mondfinsternisse vorauszusagen. Vielleicht hat es auch zur Vorhersage der verschiedenen Stellungen von Sonne und Mond zur Erde und damit zur Vorhersage der Jahreszeiten gedient. Die Anordnung der Steine bei seiner Errichtung ungefähr 2200 v. Chr. ist damit ein frühes Beispiel von Versuchsplanung [39].

(Abbildung von http://www.astro.virginia.edu/images/astronomy/stonehenge.jpg)

Zusammenfassung

In dieser Arbeit werden Probleme der Optimalen Versuchsplanung zur Parameterschätzung bei nichtlinearen DAE-Modellen behandelt.

Es wird eine allgemeine mathematische Problemformulierung hergeleitet, zu deren Lösung geeignete numerische Methoden bereitgestellt werden. Diese wurden in einem Softwarepaket implementiert, dessen praktische Anwendung die erfolgreiche Funktionsweise der Methodik demonstriert.

Wir betrachten dynamische Prozesse, die durch Systeme Differentiell-Algebraischer Gleichungen modelliert werden können. Exemplarisch untersuchen wir Anwendungen für chemische Reaktionssysteme.

Zur Modellvalidierung wird durch nichtlineare Parameterschätzung das Prozeßmodell an experimentelle Daten angepaßt. Die Signifikanz der Schätzung beschreiben wir mittels Sensitivitätsanalyse durch Näherungen der Konfidenzgebiete und durch die Kovarianzmatrix.

Zur Minimierung von Gütekriterien auf der Kovarianzmatrix formulieren wir Nichtlineare Optimale Versuchsplanungsprobleme. Optimierungsvariablen sind die Prozeßsteuerungen und das Meßlayout. Es handelt sich dabei um beschränkte Optimalsteuerungsprobleme. Zur Lösung setzen wir SQP-Verfahren ein. Die Zielfunktion hängt von Sensitivitäten der Modellfunktionen nach den Parametern ab. Wir benutzen Matrixableitungskalkül und Interne Numerische Differentiation in Verbindung mit Automatischer Differentiation, um alle benötigten Ableitungen effizient bereitzustellen.

Wir formulieren Mehrfachexperimentprobleme, dabei können wir die Informationen aus Vorexperimenten mitberücksichtigen. Für die Behandlung der ganzzahligen Variablen zur Modellierung des Meßlayouts geben wir geeignete Relaxierungen und Heuristiken an. Da nichtlineare optimale Versuchspläne von der Unsicherheit der Modellparameter abhängen, untersuchen wir Ansätze zur Robusten Versuchsplanung.

Zur Behandlung von allgemeinen Aufgaben dieser Problemklasse haben wir das Softwarepaket VPLAN entwickelt. Durch dessen Anwendung auf vier ausgewählte Praxisbeispiele aus chemischer Reaktionskinetik und Verfahrenstechnik können wir zeigen, daß die Methodik erfolgreich zur Planung effizienter und effektiver Experimente eingesetzt werden kann.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Gegenstand dieser Arbeit ist die Behandlung von Optimalen Versuchsplanungsproblemen zur Parameterschätzung bei nichtlinearen DAE-Modellen. Wir spannen den Bogen dabei von der Formulierung der mathematischen Aufgabenstellung für eine sehr allgemeine Problemklasse über die Bereitstellung numerischer Methoden zu deren Lösung und die Implementierung der Algorithmen in einem für den praktischen Einsatz geeigneten Softwarepaket. Auch der Anwendungsaspekt spielt eine wichtige Rolle. Anhand konkreter Beispiele aus der industriellen Praxis für chemische Reaktionskinetik und Verfahrenstechnik wird der erfolgreiche Einsatz der Methodik demonstriert.

1.1 Aufgabenstellung

1.1.1 Ein Virtuelles Laboratorium für dynamische Prozesse

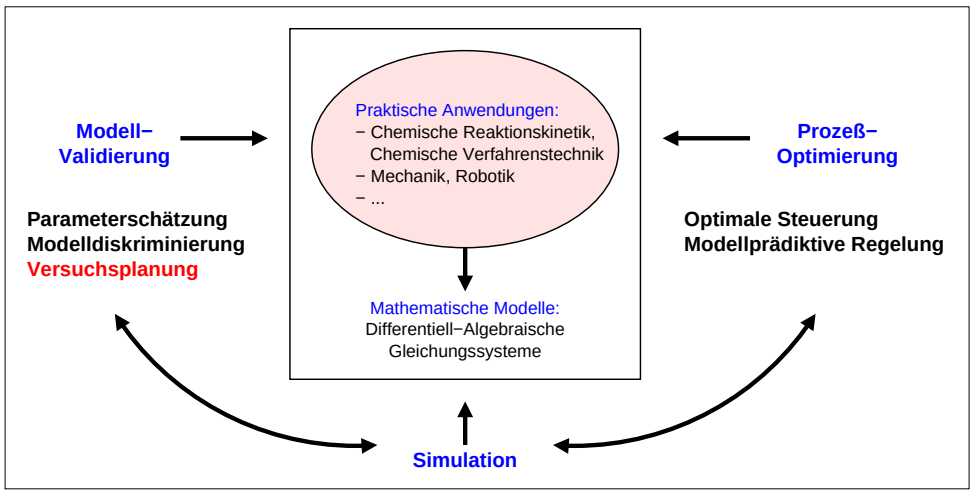


Abbildung 1.1: Modellierung, Simulation und Optimierung bei dynamischen Prozessen

Modellgestützte Simulation und Optimierung bei dynamischen Prozessen

Die Beschreibung von Prozessen durch mathematische Modelle ist in Wissenschaft, Technik und Industrie von großer Bedeutung. Anhand von Modellen lassen sich

- mittels Simulationen unter unterschiedlichen Bedingungen Reproduktionen oder Vorhersagen verschiedener Szenarien des Prozeßverhaltens durchführen.
- Aus diesen Erkenntnissen lassen sich Anlagen planen und bauen, in denen der Prozeß ablaufen soll.
- Schließlich kann das Prozeßverhalten optimiert werden, zum Beispiel bezüglich wirtschaftlicher oder technischer Kriterien wie Ausbeute oder Reinheit von Produkten, Energieverbrauch, Prozeßdauer, Kosten usw. und unter Einhaltung von gegebenen Beschränkungen. Prozßoptimierung kann entweder offline geplant werden oder online als Real-Time-Optimierung erfolgen.

Voraussetzung für den zuverlässigen Einsatz mathematischer Modelle ist jedoch, daß diese das tatsächliche Prozeßverhalten korrekt wiedergeben. Oft enthalten die Modelle Größen – wir bezeichnen diese in dieser Arbeit als *Parameter*, deren Werte nicht oder nur ungenau bestimmt sind. Wenn die Parameter stark nichtlinear in die Modellgleichungen eingehen, können auch kleine Änderungen ihrer Werte stark unterschiedliche Simulationsergebnisse bewirken.

Modellvalidierung

Daher ist es unerläßlich, die Modelle zu validieren. Dies geschieht dadurch, daß man das durch das Modell vorhergesagte Prozeßverhalten mit experimentell ermittelten Meßdaten vergleicht. Um das Modell an das wahre Prozeßverhalten anzupassen, findet die Methodik der *Parameterschätzung* Einsatz.

Experimente sind in der Regel kostspielig und zeitintensiv. Daher ist es erstrebenswert, die Versuche so zu planen und durchzuführen, daß aus den experimentellen Daten maximale Information bezüglich der Parameter geliefert wird. Dabei sind oft Nebenbedingungen an das zur Verfügung stehende Budget, die experimentellen Apparaturen oder bezüglich Sicherheit oder Umweltverträglichkeit einzuhalten. Erfahrene Experimentatoren können bei einfachen Prozessen durch Trial-and-Error effektive Versuche vorschlagen. Bei komplexen nichtlinearen Modellen ist dieser Ansatz jedoch zum Scheitern verurteilt. Wir setzen dagegen den modellgestützten Entwurf von optimalen Versuchsstrategien unter Anwendung von mathematischen und numerischen Methoden. Dies ist Gegenstand der *Optimalen Versuchsplanung.*

Nichtlineare DAE-Modelle

Naturgemäß lassen sich bei der Modellierung naturwissenschaftlicher Prozesse nicht alle Aspekte berücksichtigen. Spätestens in extremen mikro- oder makroskopischen Maßstäben

sind durch Quantenmechanik und Relativitätstheorie Grenzen gesetzt. So erhält man immer nur Approximationen des wahren Verhaltens. Wenn man für die Modellierung naturwissenschaftliche Gesetze zugrundelegt, erhält man aber Modelle, die einen größeren Gültigkeitsbereich besitzen, als wenn nur heuristisch Ursache-Wirkung-Beziehungen abgebildet werden. Eine Extrapolierbarkeit vom Labor- in den Produktionsanlagenmaßstab kann dann durchaus möglich sein.

Wir betrachten in dieser Arbeit Prozesse, die durch Systeme von nichtlinearen *Differentiell-Algebraischen Gleichungssystemen* beschrieben werden können. Exemplarisch werden wir Anwendungen aus der chemischen Reaktionskinetik und Verfahrenstechnik untersuchen, die durch die Gesetze der Physikalischen Chemie modelliert werden. Methodik und Software lassen sich aber genauso für andere Prozesse einsetzen, bei denen DAE-Systeme auftreten. Beispiele dafür sind mechanische Mehrkörpersysteme, biologische Wachstumsmodelle oder Simulationen von volkswirtschaftlichen Abläufen.

Bei der Modellierung von Anwendungen aus chemischer Reaktionskinetik und Verfahrenstechnik sind diese DAEs im allgemeinen nichtlinear in Zustandsgrößen, Parametern und Steuerungen und außerdem steif. Wir nehmen an, daß sie den Index 1 haben.

Neben den Parametern enthalten die Modelle in der Regel Variablen, die vom Experimentator eingestellt oder gewählt werden können, um das Prozeßverhalten zu beeinflussen. Wir nennen diese in dieser Arbeit *Steuerungen*. Sie und die Größen zur Beschreibung des Meßlayouts werden uns als Optimierungsvariablen zur Optimalen Versuchsplanung dienen.

The experimenter controls the experimental conditions,
wheras "nature" determines the parameters. [93]

$$\begin{aligned} A(t, y(t), z(t), p, q)\ \dot{y}(t) &= f(t, y(t), z(t), p, q, u(t)) \\ 0 &= g(t, y(t), z(t), p, q, u(t)) \end{aligned}$$

Abbildung 1.2: Die Modellierung der in dieser Arbeit betrachteten Prozesse ergibt Differentiell-Algebraische Gleichungssysteme. Siehe Kapitel 2 bis 3.

1.1.2 Optimale Versuchsplanung zur Parameterschätzung

Zur Anpassung des Modells an die experimentellen Daten lösen wir *beschränkte nichtlineare Parameterschätzprobleme*. Da die Eingangsdaten der Parameterschätzung, also die Meßdaten, in der Regel mit statistischen Fehlern behaftet sind, sind auch die Ausgabedaten, also die geschätzten Parameter, Zufallsgrößen. Die Güte der Parameterschätzung läßt sich durch *Konfidenzgebiete* des Schätzers quantifizieren. Die Größe und Gestalt der

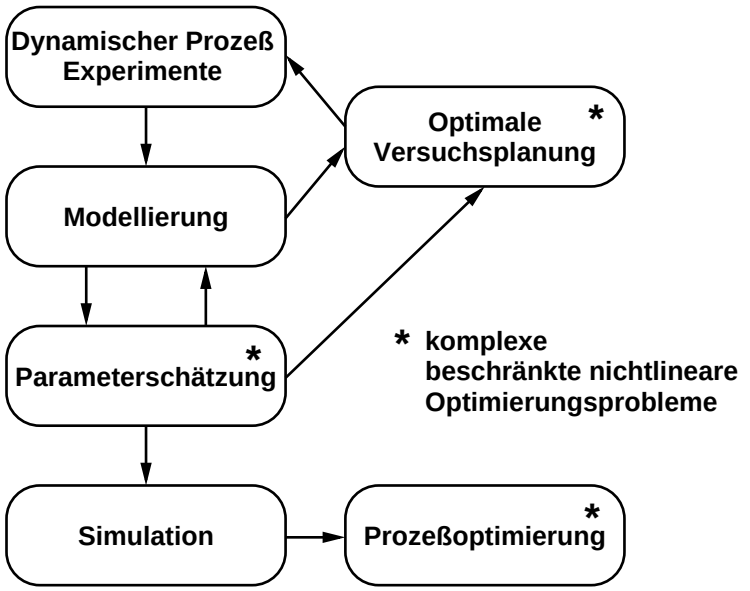


Abbildung 1.3: Auftreten nichtlinearer Optimierungsprobleme

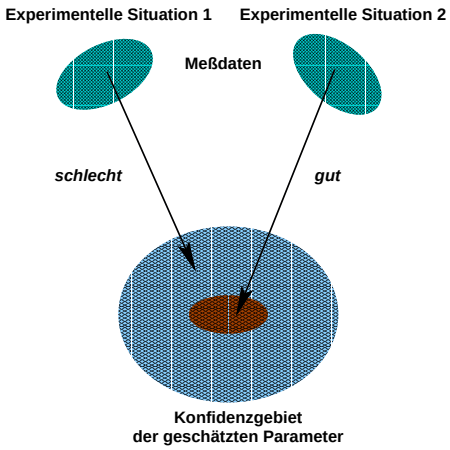


Abbildung 1.4: Da die Meßdaten mit einem statistischen Fehler behaftet sind, ergeben sich für die Parameteschätzung — je nach experimenteller Situation verschieden große — (linearisierte) Konfidenzgebiete. Siehe Kapitel 4.

Konfidenzgebiete hängt von den experimentellen Bedingungen und dem Meßlayout ab, siehe Abbildung 1.4.

In dieser Arbeit beschreiben wir Näherungen der Konfidenzgebiete durch die *Kovarianzmatrix* im Lösungspunkt des Parameterschätzproblems und formulieren *Optimale Versuchsplanungsprobleme* als die Minimierung von Gütekriterien auf der Kovarianzmatrix unter gegebenen Beschränkungen an die Versuchsplanungsgrößen (Steuerungen und Meßentwurfsgrößen) sowie an das Verhalten des Prozesses.

$$C = \begin{pmatrix} I & 0 \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_1^T J_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-T} \begin{pmatrix} I \\ 0 \end{pmatrix}$$

Abbildung 1.5: Kovarianzmatrix eines beschränkten Parameterschätzproblems. Siehe Kapitel 4 bis 5.

Dadurch erhalten wir nichtlineare beschränkte Optimalsteuerungsprobleme. Eine besondere Schwierigkeit besteht darin, daß die Zielfunktion von Sensitivitäten des Parameterschätzproblems und damit von Ableitungen der Lösung des zugrundeliegenden DAE-Systems nach den Parametern abhängt.

Zur Lösung dieser Probleme mit SQP-Verfahren ist eine effiziente Ableitungserzeugung erforderlich, die insbesondere auch gemischte zweite Ableitungen der Trajektorien nach Parametern und Steuerungen bereitstellen muß. Wir benutzen dazu Techniken der Internen Numerischen Differentiation und der Automatischen Differentiation.

Eine weitere Schwierigkeit besteht in der Ganzzahligkeit der Meßentwurfsgrößen. Diese behandeln wir mit geeigneten Relaxierungen und Heuristiken.

Häufig müssen nicht nur einzelne Experimente, sondern ganze Versuchsreihen entworfen werden, wobei auch schon durchgeführte Vorexperimente mitberücksichtigt werden sollen. Unsere Algorithmen sind speziell zur Behandlung solcher *Mehrfachexperimentprobleme* konzipiert, so daß in den numerischen Berechnungen die sich daraus ergebenden Strukturen effizient ausgenutzt werden können.

Bei nichtlinearen Modellen hängt die Kovarianzmatrix nicht nur von den Versuchsplanungsgrößen, sondern auch von den — a priori unbekannten oder schlecht bestimmten — Parameterwerten ab. Um Versuchspläne zu erhalten, die unempfindlich gegen die Unsicherheit der Parameterwerte sind, schlagen wir zwei Ansätze der *Robusten Versuchsplanung* vor.

1.1.3 Implementierung und praktische Anwendung

Ein wichtiger Bestandteil dieser Arbeit ist die praktische Implementierung der Methodik in dem Softwarepaket VPLAN.

$$\min_{p,x} \frac{1}{2} \sum_{i=1}^{M} \left(\frac{\eta_i - h_i(t_i, x(t_i), p, q)}{\sigma_i} \right)^2$$

s. t.

$$\begin{aligned} A(t,y,z,p,q)\,\dot{y} &= f(t,y,z,p,q,u) \\ 0 &= g(t,y,z,p,q,u) \\ 0 &= \hat{d}(x(t_1),\ldots,x(t_{\hat{K}}),p,q) \end{aligned}$$

$$\min_{q,u,w,x} \quad \phi(C)$$

C ist die Kovarianzmatrix im Lösungspunkt des beschränkten Parameterschätzproblems

$$\begin{aligned} \min_{p,s} \quad & \frac{1}{2} \sum_{i=1}^{M} w_i \cdot \frac{(\eta_i - h_i(t_i, x(t_i,p,q,u,s),p,q))^2}{\varsigma_i(x(t_i,p,q,u,s),q)^2} \\ 0 \;&=\; d(x(t_1,p,q,u,s),\ldots,x(t_K,p,q,u,s),p,q,s). \end{aligned}$$

Nebenbedingungen sind das DAE-System

$$\begin{aligned} A(t,y,z,p,q)\,\dot{y} &= f(t,y,z,p,q,u) \\ 0 &= g(t,y,z,p,q,u) \end{aligned}$$

sowie

$$\begin{aligned} lo &\le \hat{\psi}(t,x,p,q,u,w) \le up, \\ 0 &= \hat{\chi}(t,x,p,q,u,w) \end{aligned}$$

und

$$w \in \{0,1\}^M.$$

Abbildung 1.6: Optimierungsprobleme zur Parameterschätzung (links) und Optimalen Versuchsplanung (rechts). Siehe Kapitel 4 bis 5.

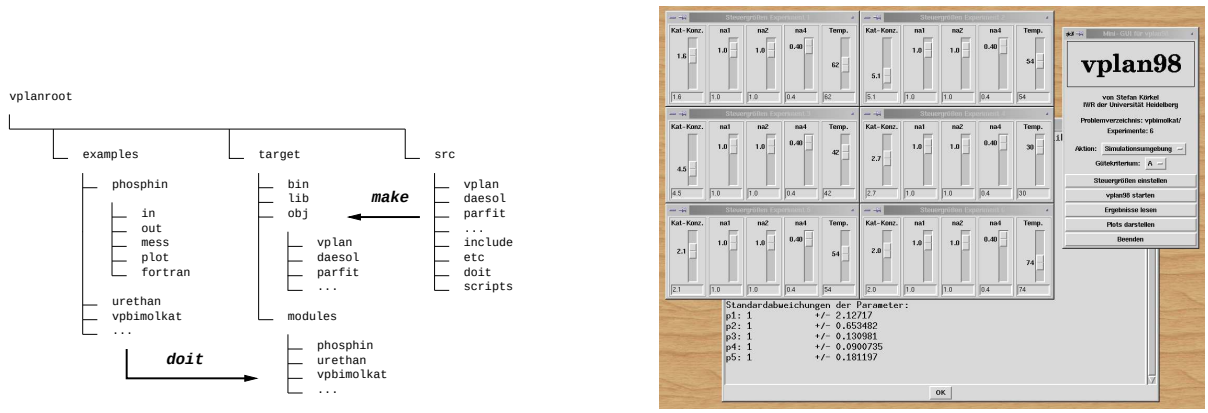


Abbildung 1.7: Verzeichnisstruktur der Software VPLAN (links). Screenshot einer grafischen Benutzeroberfläche (rechts). Siehe Kapitel 9.

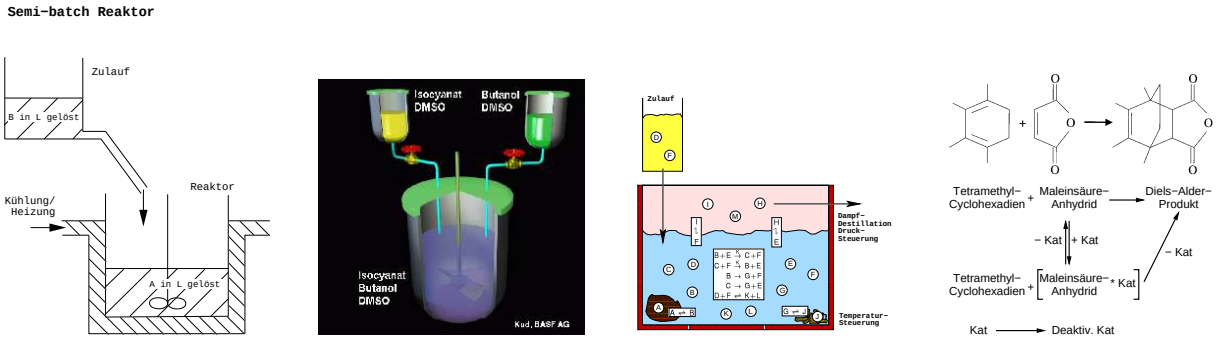


Abbildung 1.8: Reaktoren bzw. Reaktionsschemata der vier betrachteten Anwendungsbeispiele: Phosphinreaktion, Urethanreaktion, mehrphasige Veresterungsreaktion, Bimolekulare Katalyse (von links nach rechts). Siehe Kapitel 10.

Mit diesem Programm können allgemeine Anwendungen der Problemklasse *Optimale Mehrfachexperiment-Versuchsplanungsprobleme für nichtlineare DAE-Systeme* modelliert und gelöst werden. Außerdem beinhaltet die Software auch die Behandlung von Simulations- und Parameterschätzaufgaben. VPLAN besitzt eine flexible und benutzerfreundliche Bedienungsschnittstelle, grafische Benutzeroberflächen und Visualisierungstools.

Mit VPLAN wurden bereits zahlreiche Anwendungsbeispiele gerechnet. In dieser Arbeit stellen wir eine Auswahl dar und demonstrieren daran die erfolgreiche Funktionsweise von Methodik und Software. VPLAN wird erfolgreich in der industriellen Praxis eingesetzt.

1.2 Einordnung der Arbeit in das Fachgebiet

1.2.1 Lineare und Nichtlineare Versuchsplanung

Zum Entwurf von Experimenten ist die *Lineare Versuchsplanung* seit langem etabliert. Bei ihr werden lineare Regressionsmodelle betrachtet, das heißt die Modellantwort hängt linear von den Parametern p ab:

$$\eta_i = h_i(q)^T p + \epsilon_i, \quad i = 1, \ldots, M.$$

Das Lehrbuch von Mead [78] behandelt statistische Prinzipien zur Versuchsplanung bei linearen Modellen. Eine Einführung in die Optimale Lineare Versuchsplanung gibt das Buch von Atkinson und Donev [7]. Die Standardwerke von Fedorov [44] und Pukelsheim [93] enthalten ausführliche Darstellungen zur Theorie und Anwendung dieser Methodik.

Einführungs- und Übersichtsartikel liegen außerdem vor von Atkinson [9, 5, 6], Nishii [82] sowie Draper und Pukelsheim [39]. Letztere geben dabei insbesondere eine historische Einführung in die Versuchsplanung.

Cook und Fedorov [33] sowie Fedorov und Hackl [46] behandeln beschränkte lineare Versuchsplanungsprobleme und geben numerische Methoden zu deren Lösung an. Fedorov und Hackl [45] untersuchen speziell Spatial Sampling, die Plazierung der Messungen bzw. Beobachtungen.

Die Methodik kann für nichtlineare Regressionsmodelle

$$\eta_i = h_i(p, q) + \epsilon_i, \quad i = 1, \ldots, M$$

verallgemeinert werden. Eine unfassende Darstellung über nichtlineare Parameterschätzung gibt zum Beispiel das Lehrbuch von Seber und Wild [107]. Sehr kleine Beispiele zur *Nichtlinearen Optimalen Versuchsplanung*, bei denen Gleichungen und Optimierung analytisch gelöst werden, werden zum Beispiel von Haines [60] oder Rudolph und Herrendörfer [101] betrachtet. Dovíc, Reverberi und Acevedo-Duarte [37] behandeln ein Beispiel aus der chemischen Reaktionskinetik, bei dem Reaktionsgeschwindigkeiten direkt gemessen werden, so daß keine Differentialgleichungen formuliert oder gelöst werden müssen.

Will man dynamische Prozesse modellieren, hängt die Modellfunktion im allgemeinen von der Lösung eines Differentialgleichungssystems (hier Differentiell-Algebraisches Gleichungssystem) ab:

$$\eta_i = h_i(t, x(t), p, q, u) + \epsilon_i, \quad i = 1, \ldots, M,$$

wobei $x(t) = (y(t), z(t))$ ein DAE-System erfüllt

$$\begin{aligned} A(t, y(t), z(t), p, q)\ \dot{y}(t) &= f(t, y(t), z(t), p, q, u(t)) \\ 0 &= g(t, y(t), z(t), p, q, u(t)). \end{aligned}$$

Für spezielle Beispiele aus der chemischen Reaktionskinetik, bei denen sich die Differentialgleichungen analytisch lösen lassen, berechnen Reilly, Bajramovic, Blau, Branson und Sauerhoff [98], Qureshi, Ng und Goodwin [95] oder Oinas, Turunen und Haario [86] Versuchspläne. Bei Lohmann, Bock und Schlöder [75] bzw. Lohmann [74] hängt die Dynamik nicht von den Versuchsplanungsgrößen ab, optimiert wird nur über die Meßentwurfsgrößen.

Baltes, Schneider, Sturm und Reuss [10] modellieren ein unstrukturiertes Wachstumsmodell aus der Bioverfahrenstechnik mit einer gewöhnlichen Differentialgleichung und betrachten dafür unbeschränkte Parameterschätzprobleme und deren Kovarianzmatrix. Eine Zielfunktion darauf minimieren sie durch ableitungsfreie „Optimierung“ mit dem Nelder-Mead-Verfahren. In der Arbeit von Hilf [65] wird eine Anwendung aus der Mechanik untersucht. Da das Modell nicht steif ist, kann ein expliziter Integrator verwendet werden. Dafür wird eine spezielle Ableitungserzeugung hergeleitet und zur Versuchsplanung für unbeschränkte Parameterschätzung eingesetzt.

Bock [26] gibt effiziente numerische Methoden für beschränkte nichtlineare Parameterschätzprobleme in Systemen gewöhnlicher Differentialgleichungen an. Zur Sensitivitätsanalyse für diese Probleme formuliert er erstmals die Kovarianzmatrix für den beschränkten Fall, siehe auch Lohmann [73]. Raum [97] berechnet Ableitungen der Kovarianzmatrix nach Steuergrößen für unbeschränkte und beschränkte Parameterschätzprobleme.

Bei nichtlinearen Modellen hängt die Kovarianzmatrix von den Modellparametern ab. Daher werden in der Literatur Ansätze zur *Robusten Versuchsplanung* vorgeschlagen, die diese Abhängigkeit berücksichtigen sollen.

Fedorov [44] schlägt eine sequentielle Vorgehensweise vor. Rasch [96] führt anhand eines konkreten Beispiels mittels Diskretisierung eine Untersuchung auf Robustheit durch. Draper und Hunter [38] berücksichtigen die Verteilung der Parameter durch eine Modifikation der Zielfunktion. Bei Ford, Torsney und Wu [49] kann in einer speziellen Formulierung das Design explizit in Abhängigkeit der Parameter dargestellt werden. Pronzato und Walter [92] benutzen Diskretisierungsmethoden, um den Erwartungswert oder den Worst Case zu minimieren. Alle diese Ansätze beschränken sich auf die Anwendung auf kleinste Lehrbeispiele und kommen ohne Dynamik oder Numerik aus.

Eine verwandte Aufgabenstellung ist die Versuchsplanung zur *Modelldiskriminierung.* Dabei werden Experimente geplant, um unter konkurrierenden Modellen dasjenige zu bestimmen, das den Prozeß am besten beschreibt. Zur Theorie dazu sei auf die Arbeiten von

Atkinson und Fedorov [8], Ponce De Leon und Atkinson [71] oder O'Brien und Rawlings [85] verwiesen. Die Anwendung dieser Methodik auf Probleme mit dynamischen Prozeßmodellen wird von Dieses [35] oder Asprey [3] behandelt.

1.2.2 Eigene Veröffentlichungen

Ein Teil der vorliegenden Arbeit entstand während der Mitarbeit des Autors im BMBF-Projekt „Optimale Versuchsplanung für nichtlineare Prozesse“der Projektpartner Interdisziplinäres Zentrum für Wissenschaftliches Rechnen der Universität Heidelberg, Aventis, BASF und Fachhochschule Frankfurt. Die Ergebnisse dieses Projekts sind in einer schriftlichen Projektpräsentation [87] und in einem Projektbericht [20] publiziert.

Während der Arbeit an dieser Dissertation entstanden über Teilaspekte mehrere Veröffentlichungen. Einen Überblick über Problemformulierung, Methodik und Anwendungen gibt [19], die Behandlung verschiedener Anwendungsbeispiele ist in [18] dargestellt. Die Artikel [15] bzw. [68] behandeln Interne Numerische Differentiation für zweite Ableitungen bzw. sequentielle Vorgehensweise und Mehrfachexperimente. Eine ausfürliche Darstellung der Methodik mit Schwerpunkt auf der Ableitungserzeugung und einer Anwendung auf ein Beispiel der Reaktionskinetik gibt [16].

Unter Verwendung unserer Software VPLAN wurden weitere Anwendungsbeispiele behandelt. Dieses [36] setzt Methodik und Software zur Versuchsplanung beim Schadstofftransport in Böden ein, Estler et al. [42] zur Untersuchung einer Enzymkinetik.

Diese Arbeit ordnet sich ein in das Gebiet des *Wissenschaftlichen Rechnens.* Dieses besteht in der Verbindung von

- interdisziplinärer Untersuchung und Modellierung von praktischen Anwendungsproblemen zusammen mit Experten aus anderen naturwissenschaftlichen Fachgebieten,
- Neuentwicklungen in mathematischer Methodik und numerischen Verfahren,
- der Implementierung der Algorithmen und Datenstrukturen
- und der Entwicklung eines leistungsfähigen Softwarepakets,
- das zur Lösung einer allgemeinen Klasse praxisrelevanter Probleme eingesetzt werden kann
- und auch tatsächlich eingesetzt wird.

1.3 Überblick über den Inhalt der Arbeit

In Kapitel 2 geben wir einen Überblick über die Modellierung chemischer Reaktionssysteme. Wir erhalten dabei nichtlineare Systeme von Differentiell-Algebraischen Gleichungen,

die in der Regel steif sind. Die in diesem Kapitel diskutierten Prinzipien wenden wir in dieser Arbeit an, um die konkreten Anwendungsbeispiele in Kapitel 10 zu modellieren.

In Kapitel 3 betrachten wir zur Lösung von DAE-Systemen lineare Mehrschrittverfahren. Dabei zeigt sich, daß BDF-Verfahren für die Probleme dieser Arbeit besonders geeignet sind.

Kapitel 4 behandelt nichtlineare beschränkte Parameterschätzprobleme. Wir gehen auf Gauß-Newton-Verfahren zu deren Lösung ein und diskutieren Ansätze zur Parametrisierung der DAE-Modellgleichungen. Die Sensitivitätsanalyse im Lösungspunkt der Parameterschätzung liefert uns die Kovarianzmatrix und Näherungen der Konfidenzgebiete.

In Kapitel 5 formulieren wir darauf aufbauend Nichtlineare Optimale Versuchsplanungsprobleme. Als Methode zu deren Lösung schlagen wir SQP-Verfahren vor. Wir diskutieren außerdem die Behandlung der Ganzzahligkeitsproblematik.

Die Berechnung und Bereitstellung der benötigten Ableitungen wird in Kapitel 6 behandelt. Wir bewerkstelligen dies durch den Einsatz von Matrixableitungskalkül und Interner Numerischer Differentiation in Verbindung mit Automatischer Differentiation.

In Kapitel 7 verallgemeinern wir die Problemformulierungen auf Mehrfachexperimentprobleme und zeigen, wie diese effizient unter Ausnutzung der auftretenden Strukturen gelöst werden können.

Kapitel 8 beschäftigt sich mit Robuster Versuchsplanung. Wir schlagen zwei Ansätze dafür vor, eine sequentielle Vorgehensweise aus Optimaler Versuchsplanung, Versuchsdurchführung und Parameterschätzung sowie einen Worst-Case-Ansatz zur Einbeziehung der Parameterstreuung.

Die Implementierung der numerischen Methoden zur Nichtlinearen Optimalen Versuchsplanung im Softwarepaket VPLAN ist Gegenstand von Kapitel 9. Wir stellen Leistungsumfang und Bedienung des Programms vor und geben einen Überblick über die Struktur des Programmpakets und spezielle Techniken bei der Softwareentwicklung.

Schließlich demonstrieren wir in Kapitel 10 anhand vier ausgewählter Praxisanwendungen aus chemischer Reaktionskinetik und Verfahrenstechnik den erfolgreichen Einsatz der Methodik und Software. Wir können dabei zeigen, daß die sequentielle Optimale Versuchsplanung einer intuitiven Versuchsplanung bezüglich Effizienz und Effektivität weit überlegen ist, daß auch sehr komplexe mehrphasige Modelle erfolgreich behandelt werden können und daß die Robuste Versuchsplanung erfolgreich zur Reduzierung der Parameterabhängigkeit von optimalen Designs eingesetzt werden kann.

1.4 Danksagungen

Mein Dank für die interessante Aufgabenstellung und die Betreuung dieser Arbeit gilt Prof. Dr. Dr. h. c. Hans Georg Bock und Dr. Johannes Schlöder. Die angenehme Atmosphäre in ihrer Arbeitsgruppe am Interdisziplinären Zentrum für Wissenschaftliches Rechnen der Universität Heidelberg hat sehr stark zum Erfolg dieser Arbeit beigetragen.

Dieses Promotionsprojekt wurde finanziert von der Deutschen Forschungsgemeinschaft (DFG) im Rahmen eines Stipendiums im Graduiertenkolleg „Modellierung und Wissenschaftliches Rechnen in Mathematik und Naturwissenschaften“ und einer wissenschaftlichen Mitarbeiterstelle im Sonderforschungsbereich 359 „Reaktive Strömungen, Diffusion und Transport“, vom Bundesministerium für Bildung und Forschung (BMBF) im Projekt „Optimale Versuchsplanung für nichtlineare Prozesse“, sowie von der BASF AG durch freie Mitarbeitertätigkeit.

An die BASF geht Dank für die hervorragende Zusammenarbeit, besonders an Dr. Alexander Kud, mit dem gemeinsam die Anwendungsbeispiele modelliert wurden und der durch seine Benutzung des Programms VPLAN und die daraus resultierenden Verbesserungsvorschläge stark zur Reife der Software beigetragen hat. Die intuitiven Versuchspläne für die Phosphin- bzw. Urethanreaktion wurden von Dr. M. Heilig bzw. Dr. O. Wörz von der BASF entworfen.

Bei Dr. Irene Bauer möchte ich mich für die erfolgreiche Zusammenarbeit im BMBF-Projekt bedanken. Dr. Ekaterina Kostina hat wichtige Ideen zur Robusten Versuchsplanung beigesteuert.

Bei der Entwicklung von VPLAN haben verschiedene Kollegen im Rahmen von Diplom-Praktikums- und Hilfskraftarbeiten mitgewirkt. Gerd Rücker hat im Rahmen seiner Diplomarbeit das Modellierungs- und Ableitungserzeugungswerkzeug mit grafischer Benutzeroberfläche MGAD entwickelt und außerdem an den plattformunabhängigen Installationsskripten mitgearbeitet. Sebastian Sager hat in langjähriger Mitarbeit im Projekt die Einlese- und Ausgaberoutinen und vor allem das Programm doit zum vollautomatischen Aufruf von ADIFOR programmiert. Bei beiden möchte ich mich für das ständige hohe Engagement ganz herzlich bedanken. An der Softwareentwicklung haben außerdem Stefan Hetzel, Elmar Körding, Ulrich Stemmermann und Jörg Weller erfolgreich mitgearbeitet.

Für des freundschaftliche Verhältnis und die interessanten Diskussionen möchte ich mich bei meinen Kollegen Hoang Duc Minh, Tran Hong Thai, Hinderk Buß, Peter Riede, Jan Schrage und Dr. Angelika Dieses bedanken, die mir immer mit Rat und Tat beigestanden haben.

Und schließlich gilt mein ganz besonderer Dank meinen Eltern, die mir durch Förderung, Anregung und Unterstützung mein Studium und die Promotion überhaupt erst möglich gemacht haben.

Kapitel 2

Modellierung chemischer Reaktionssysteme

In diesem Kapitel werden grundlegende Begriffe und Zusammenhänge aus der Physikalischen Chemie und der Chemischen Verfahrenstechnik vorgestellt. Sie bilden den Rahmen zur Aufstellung mathematischer Modelle für chemische Reaktionssysteme. Wir gehen hier insbesondere ein auf Reaktionskinetik, chemische Gleichgewichte, Thermodynamik, Reaktionstechnik und Bilanzen. Darstellungen, die auch weiterführende Aspekte und Hintergründe behandeln, findet man zum Beispiel in [12], [4], [116], [91], [80]. In Kapitel 10 stellen wir mehrere Beispielanwendungen vor, die nach den Prinzipien dieses Kapitels modelliert wurden und auf die die in dieser Arbeit behandelten Methoden zur Simulation, Parameterschätzung und Versuchsplanung angewendet wurden.

2.1 Überblick

Die zeitliche Änderung der Zusammensetzung des Gemischs eines chemischen Reaktionssystems wird durch die *Reaktionskinetik* beschrieben. Ihre Geschwindigkeitsgesetze ergeben Differentialgleichungen für die Stoffmengen n oder die Konzentrationen c der an den Reaktionen beteiligten Spezies.

Das Reaktionsverhalten ist in der Regel temperaturabhängig. Entweder kann die Reaktionstemperatur direkt vom Experimentator gesteuert werden. Oder man modelliert die Wärmebilanz des Prozesses mittels Reaktionswärme und Wärmeaustausch mit den Gesetzen der *Thermodynamik*. Dadurch erhält man eine Differentialgleichung für die Temperatur T oder die Wärme Q.

Wichtig bei der Modellierung ist die Beachtung von *Bilanzen* und Erhaltungssätzen, zum Beispiel Massenerhaltung oder Stoffmengenbilanz.

Für 0-D-instationäre oder 1-D-stationäre Prozesse erhält man Systeme aus Differentialgleichungen und algebraischen Gleichungen

$$\begin{aligned} A(t, y(t), z(t), p, q)\ \dot{y}(t) &= f(t, y(t), z(t), p, q, u(t)) \\ 0 &= g(t, y(t), z(t), p, q, u(t)). \end{aligned}$$

Dabei ist t für 0-D-instationäre Prozesse die Zeit, für 1-D-stationäre Prozesse steht es für die Ortskoordinate. y bzw. z bezeichnen die differentiellen bzw. algebraischen Zustandsvariablen. Die mathematische und numerische Behandlung solcher *DAEs* diskutieren wir in Kapitel 3. Stationäre Prozesse mit räumlichen Inhomogenitäten in zwei oder mehr Dimensionen und instationäre Prozesse in ein oder mehr Dimensionen ergeben Systeme von partiellen Differentialgleichungen. Oft können diese mit der Linienmethode [103, 118] in DAE-Systeme transformiert werden.

Meist enthalten die Modelle Größen, deren Werte man nicht oder nur ungenau kennt. Wir nennen diese Variablen *Parameter* p. Zum Beispiel können dies Kinetikkonstanten, Reaktionsordnungen, Reaktionsenthalpien oder Gleichgewichtskonstanten sein. Ziel der Untersuchungen in dieser Arbeit ist die möglichst genaue Bestimmung dieser Parameter. Mathematische Methoden zu deren Schätzung aus experimentellen Daten behandeln wir in Kapitel 4.

Bestandteil der Modelle ist aber auch die *Reaktionstechnik*. Die Durchführung des Prozesses hängt erheblich vom Reaktortyp und der Reaktionsführung ab. Die Beeinflussung des Prozeßverhaltens durch den Experimentator wird im Modell durch die *Steuergrößen* q angegeben. Diese beschreiben typische Reaktionsbedingungen wie zum Beispiel das Reaktorvolumen, die Mengen der Einwaagen im Reaktor und in den Einträgen oder im isothermen Fall die Reaktionstemperatur. Manche Versuchsbedingungen können zeitlich variabel geführt werden, zum Beispiel die Steuerung der Zuläufe oder das Profil der Kühlung oder Heizung. Diese Variablen nennen wir *Steuerfunktionen* $u(t)$. Steuergrößen und Steuerfunktionen sind in der Optimalen Versuchsplanung neben der Auswahl der Messungen die Variablen, die man optimieren kann, um möglichst aussagekräftige Experimente bei gegebenen experimentellem Aufwand zu bestimmen, siehe Kapitel 5.

2.2 Kinetik chemischer Reaktionen

2.2.1 Reaktionsgeschwindigkeit

Bei einer chemischen Reaktion

$$(-\nu_1)A_1 + \ldots + (-\nu_{m_1})A_{m_1} \rightarrow \nu_{m_1+1}A_{m_1+1} + \ldots + \nu_m A_m$$

reagieren die *Edukte* $A_1, \ldots, A_{m_1}$ zu den *Reaktionsprodukten* $A_{m_1+1}, \ldots, A_m$. In der Reaktionsgleichung sind die *stöchiometrischen Koeffizienten* ν_i negativ für die Edukte und positiv für die Produkte.

Im allgemeinen treten *Reaktionssysteme* mit n Reaktionen und m Spezies auf. Hierbei können die Stoffe A_1 bis A_m in verschiedenen Reaktionen sowohl als Edukte als auch als Produkte vorkommen. Die Stöchiometriekoeffizienten ν_{ji}, $i = 1, \ldots, m$, $j = 1, \ldots, n$ kann man in der *Stöchiometriematrix*

$$\nu = \begin{pmatrix} \nu_{11} & \cdots & \nu_{1m} \\ \vdots & & \vdots \\ \nu_{n1} & \cdots & \nu_{nm} \end{pmatrix}$$

zusammenfassen. Dabei ist $\nu_{ji} < 0$, falls Spezies i in Reaktion j ein Edukt ist, $\nu_{ji} > 0$, falls Spezies i in Reaktion j ein Produkt ist, und $\nu_{ji} = 0$, falls Spezies i nicht an Reaktion j teilnimmt.

Zur Definition der Reaktionsgeschwindigkeit ist es sinnvoll, zunächst einzelne Reaktionen zu betrachten.

Die Molzahl- oder *Stoffmengenänderungen* der Spezies sind gegeben durch

$$\Delta n_i := n_i - n_{i,0}, \quad i = 1, \ldots, m$$

mit den Anfangsstoffmengen $n_{i,0}$, $i = 1, \ldots, m$. Es gilt dann: $\Delta n_i < 0$ für Edukte und $\Delta n_i > 0$ für Produkte.

Wenn keine Komponenten zu- oder abgeführt werden, ist aufgrund der Stöchiometrie

$$\frac{\Delta n_1}{\nu_1} = \ldots = \frac{\Delta n_m}{\nu_m}.$$

Diesen Quotienten nennt man *Reaktionsfortschritt*, *Reaktionslaufzahl* oder *Formelumsatz*:

$$\xi := \frac{\Delta n_i}{\nu_i}.$$

Dann gilt:

$$n_{i,R} = n_{i,0} + \nu_i \cdot \xi.$$

In der Reaktionskinetik interessiert man sich für den zeitlichen Ablauf von Reaktionen. Differenziert man die letzte Gleichung nach der Zeit t, so erhält man

$$\dot{n}_{i,R} = \frac{\mathrm{d}n_i}{\mathrm{d}t} = \nu_i \cdot \frac{\mathrm{d}\xi_i}{\mathrm{d}t}.$$

Als *Reaktionsgeschwindigkeit* bezeichnet man

$$r := \frac{1}{V} \cdot \frac{\mathrm{d}\xi}{\mathrm{d}t},$$

die Anzahl der Formelumsätze pro Zeit- und Volumeneinheit.

Die Kenntnis bzw. Bestimmung der Reaktionsgeschwindigkeit ist wichtig,

- um die Zusammensetzung der Reaktionsmischung berechnen oder vorausberechnen zu können
- und um Hinweise für das theoretische Verständnis des Reaktionsmechanismus zu erhalten.

2.2.2 Geschwindigkeitsgesetze

Für die Reaktionsgeschwindigkeit können häufig Geschwindigkeitsgesetze angegeben werden. Bei vielen Reaktionen hängt die Geschwindigkeit von den Konzentrationen der Edukte ab:

$$r = k \cdot \prod_{Edukte} c_i^{\alpha_i}.$$

Der konzentrationsunabhängige Term k heißt *Geschwindigkeitskonstante*. α_i ist die *Ordnung* der Reaktion bezüglich Edukt i, im allgemeinen ist sie ungleich dem Stöchiometriekoeffizienten $(-\nu_i)$. Ordnungen können 0,1,2,... sein oder auch gebrochen. Die *Gesamtordnung* der Reaktion ist die Summe aller einzelnen Ordnungen.

Nicht alle Geschwindigkeitsgesetze haben jedoch diese Form. Zum Beispiel können auch die folgenden Fälle vorkommen:

- Auch Konzentrationen von Produkten können die Reaktionsgeschwindigkeit beeinflussen.
- Geschwindigkeitsgesetze können durch kompliziertere Gleichungen beschrieben werden, zum Beispiel

  $$r = \frac{k_1 \cdot c_A \cdot c_B^{\frac{1}{2}}}{c_B + k_2 \cdot c_C}.$$

  Dann kann man keine Ordnungen angeben.
- Wenn die Konzentration eines oder mehrerer Edukte sehr groß und daher praktisch konstant ist, kann das Geschwindigkeitsgesetz vereinfacht werden. Man spricht dann von Reaktionen pseudo-erster, pseudo-zweiter, ... Ordnung.

In der Regel kann das Geschwindigkeitsgesetz nicht aus der chemischen Reaktionsgleichung abgeleitet werden, es muß experimentell bestimmt werden.

2.2.3 Elementarreaktionen und Reaktionsmechanismen

Durch eine stöchiometrische Reaktionsgleichung wird meist nur die Bilanzgleichung einer chemischen Reaktion angegeben. In Wirklichkeit finden aber viele einzelne Schritte statt, sogenannte *Elementarreaktionen*. Deren Abfolge und Zusammenwirken bestimmen den *Reaktionsmechanismus* der Gesamtreaktion.

Da sich Bildungs- beziehungsweise Zersetzungsgeschwindigkeiten von Spezies bei mehr als einer Reaktion addieren, läßt sich das Geschwindigkeitsgesetz der Gesamtreaktion aus den Geschwindigkeitsgesetzen der Elementarreaktionen ableiten.

Für Elementarreaktionen kann man Geschwindigkeitsgesetze direkt angeben, sie sind bestimmt durch die *Molekularität* der Elementarreaktion, das heißt durch die Anzahl der Moleküle oder Atome, die an der Reaktion teilnehmen. Folgende Fälle sind zu unterscheiden:

- Unimolekulare Reaktionen: Ohne Beteiligung eines Reaktionspartners zerfällt oder isomerisiert ein Ausgangsmolekül: Die Geschwindigkeit ist proportional zur Konzentration des Ausgangsstoffes:
$$r = k \cdot c_A,$$
es handelt sich um eine Reaktion erster Ordnung.

- Bimolekulare Reaktionen: Zwei Moleküle oder Atome begegnen sich. Die Geschwindigkeit ist proportional zur Konzentration beider Reaktionspartner:
$$r = k \cdot c_A \cdot c_B,$$
die Reaktion ist zweiter Ordnung.

- Tri- und mehr-molekulare Reaktionen: Drei oder mehr Moleküle oder Atome begegnen sich. Da solche Ereignisse sehr unwahrscheinlich sind, kommen diese Elementarreaktionen nur sehr selten vor.

Die Summe der Elementarreaktion ergibt die Gesamtreaktion. Dabei kann zum Beispiel folgendes vorkommen:

- Hin- und Rückreaktionen finden statt.
- Produkte aus einem Schritt sind Edukte in den folgenden Schritten.
- Zwischenprodukte werden gebildet und wieder verbraucht.

Kennt man den Reaktionsmechanismus, kann man das Geschwindigkeitsgesetz der Gesamtreaktion herleiten. Wenn dabei eine Elementarreaktion viel langsamer ist als alle anderen, bestimmt sie die Geschwindigkeit der Gesamtreaktion. Man nennt sie dann *geschwindigkeitsbestimmenden Schritt.*

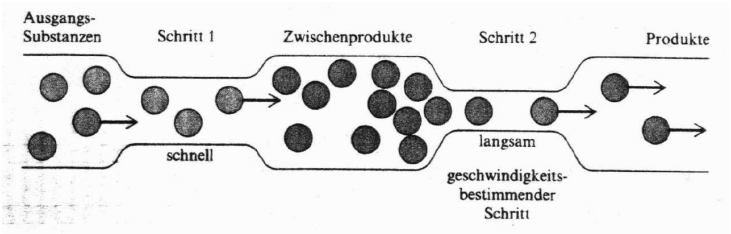


Abbildung 2.1: Geschwindigkeitsbestimmender Schritt im Mechanismus einer chemischen Reaktion

Hat man einen Reaktionsmechanismus aufgestellt und daraus ein Geschwindigkeitsgesetz bestimmt, das sich experimentell bestätigen läßt, heißt das nicht, daß der Mechanismus auf jeden Fall der richtige ist. Eine andere Abfolge von Elementarreaktionen könnte dieselbe Reaktionsgeschwindigkeit ergeben. Kinetische Betrachtungen können einen Reaktionsmechanismus also nur stützen, aber nicht beweisen.

2.2.4 Abhängigkeit der Reaktionsgeschwindigkeit von der Temperatur

Reaktionsgeschwindigkeiten werden beeinflußt durch

- die Konzentrationen der Reaktionspartner,
- die Oberflächen reagierender Körper,
- die Temperatur und den Druck
- und Katalysatoren oder Inhibitoren.

In diesem Abschnitt wollen wir den Temperatureinfluß diskutieren.

Viele Reaktionen, vor allem, aber nicht nur in der Gasphase, zeigen das *Arrhenius-Verhalten*

$$k = A \cdot e^{-\frac{E_a}{R \cdot T}}.$$

$R = 8.31 J \cdot K^{-1} \cdot mol^{-1}$ ist die universelle Gaskonstante und T die Temperatur in Kelvin. A und E_a sind für die Reaktion charakteristische Konstanten, A heißt präexponentieller Faktor oder *Frequenzfaktor* und E_a *Aktivierungsenergie*.

Durch Betrachtung der molekularen Reaktionskinetik kann man Geschwindigkeitsgesetze erklären und Geschwindigkeitskonstanten berechnen. Die Stoßtheorie für Reaktionen in der Gasphase ist ein Ansatz, mit dessen Hilfe man die Arrhenius-Beziehung herleiten kann.

Wir betrachten eine bimolekulare Elementarreaktion

$$A + B \rightarrow C.$$

Die Moleküle können nur dann miteinander reagieren, wenn sie mit so viel kinetischer Energie zusammentreffen, daß dadurch Bindungen aufgebrochen werden.

Die Stoßdichte Z_{AB} ist die Zahl von A-B-Stößen pro Zeit- und Volumeneinheit. Sie wird durch die folgende Formel beschrieben:

$$Z_{AB} = \sigma \cdot \left(\frac{8 \cdot k_B \cdot T}{\pi \cdot \mu} \right)^{\frac{1}{2}} \cdot N_A^2 \cdot c_A \cdot c_B$$

mit dem Stoßquerschnitt $\sigma = \pi^2 \cdot d^2$, wobei $d = \frac{1}{2} \cdot (d_A + d_B)$ der Radius der Trefferfläche ist und d_A und d_B die Durchmesser der Moleküle. $\mu = \frac{m_A \cdot m_B}{m_A + m_B}$ ist die reduzierte Masse, $k_B = 1.38066 \cdot 10^{-23} J \cdot K^{-1}$ die Boltzmann-Konstante, $\pi = 3.141\ldots$ und $N_A = 6.02214 \cdot 10^{23} mol^{-1}$ die Avogadrosche Konstante.

σ ist die Trefferfläche für reaktive und nichtreaktive Stöße. Multipliziert man diese mit dem sterischen Faktor $P \ll 1$, erhält man den Stoßquerschnitt für Teilchen, die an Reaktionen

teilnehmen. P hängt von den speziellen Molekülen ab, dabei gilt, je komplizierter die Moleküle, desto kleiner ist P.

Nach der Stoßtheorie ist die Änderung der molaren Konzentration von A gleich der Stoßdichte multipliziert mit der Wahrscheinlichkeit f, daß ein Stoß energiereich genug ist, um zur Reaktion zu führen:

$$\dot{c}_A = -f \cdot \frac{P \cdot Z_{AB}}{N_A}.$$

Für einen erfolgreichen Stoß sei die kinetische Mindestenergie E_{min}. Aus der Boltzmannverteilung folgt

$$\log f = -\frac{E_{min}}{R \cdot T}.$$

Also beträgt die Reaktionsgeschwindigkeit

$$r = f \cdot \frac{P \cdot Z_{AB}}{N_A} = e^{-\frac{E_{min}}{R \cdot T}} \cdot P \cdot \sigma \cdot \left(\frac{8 \cdot k_B \cdot T}{\pi \cdot \mu}\right)^{\frac{1}{2}} \cdot N_A \cdot c_A \cdot c_B.$$

Die Aktivierungsenergie ist somit gleich E_{min}, und für den Frequenzfaktor gilt

$$A = P \cdot \sigma \cdot \left(\frac{8 \cdot k_B \cdot T}{\pi \cdot \mu}\right)^{\frac{1}{2}} \cdot N_A.$$

Dies ist der Wert, den die Geschwindigkeitskonstante hätte, wenn alle Stöße erfolgreich wären. Der Frequenzfaktor hängt somit doch von der Temperatur ab, meistens kann diese Wurzel-Abhängigkeit gegenüber dem exponentiellen Term aber vernachlässigt werden. Man kann der Arrhenius-Beziehung auch eine Korrektur hinzufügen, dann setzt man für die Geschwindigkeitskonstante

$$k = A \cdot t^{\beta} \cdot e^{-\frac{E_A}{R \cdot T}}$$

an.

Bemerkung 2.2.1 *Die einzelnen Terme der Reaktionsgeschwindigkeit*

$$r = P \cdot \frac{Z_{AB}}{N_A} \cdot f$$

beschreiben verschiedene Kriterien:

- P *modelliert die lokalen Eigenschaften der Reaktanten,*
- $\frac{Z_{AB}}{N_A}$ *das Zusammentreffen der Teilchen und*
- f *das Energiekriterium.*

2.2.5 Katalysatoren und Inhibitoren

Katalysatoren sind Substanzen, deren Zugabe die Geschwindigkeit einer Reaktion erhöhen, ohne daß sie bei der Reaktion verbraucht werden. Ihre Wirkung besteht darin, daß die Reaktion auf einem anderen Reaktionsweg ermöglicht wird, dessen geschwindigkeitsbestimmender Schritt eine kleinere Aktivierungsenergie hat. Man unterscheidet homogene Katalyse und heterogene Katalyse, je nachdem, ob der Katalysator in derselben oder einer anderen Phase als die Reaktionsmischung vorliegt. Katalysatoren beschleunigen auch die Rückreaktionen. Katalysatoren können im Laufe von Reaktionen unwirksam werden, zum Beispiel durch Ablagerungen auf der Katalysatoroberfläche. Dann spricht man von Vergiftung des Katalysators. Enzyme sind — meist sehr wirkungsvolle — Biokatalysatoren. Inhibitoren bremsen chemische Reaktionen.

2.3 Chemische Gleichgewichte

Ein chemisches Reaktionsgemisch ist im Gleichgewicht, wenn sich die Konzentrationen der beteiligten Spezies nicht mehr ändern. Meist handelt es sich um dynamische Gleichgewichte, dann laufen Hin- und Rückreaktionen mit gleicher Geschwindigkeit ab. Der Gleichgewichtszustand ergibt sich als Lösung der Differentialgleichungen für die Reaktionskinetik. Er kann jedoch auch direkt durch das Massenwirkungsgesetz modelliert werden. Dies ist besonders dann sinnvoll, wenn sich das Gleichgewicht sehr schnell oder praktisch instantan einstellt.

2.3.1 Massenwirkungsgesetz

Quantitativ können chemische Gleichgewichte durch das *Massenwirkungsgesetz* beschrieben werden. Es gilt im Gleichgewichtszustand mit $c_i = c_{g,i}$, $i = 1, \ldots, m$:

$$\prod_{i=1}^{m} c_{g,i}^{\nu_i} = K_C.$$

Die (konzentrationsbezogene) *Gleichgewichtskonstante* K_C ist die charakteristische Größe für die Zusammensetzung des Gleichgewichts. Sie hängt nicht von den Konzentrationen ab. Die Exponenten sind die Stöchiometriekoeffizienten ν_i.

Ist $K_C < 1$, liegt das Gleichgewicht auf der Seite der Edukte, für $K_C > 1$ auf der Seite der Produkte, und bei $K_C \gg 1$ läuft die Reaktion vollständig ab. Da $\prod_{i=1}^{m} c_{g,i}^{\nu_i}$ konstant ist, können durch Entfernen oder Hinzufügen gewisser Reaktionsteilnehmer die Gleichgewichtskonzentrationen verschoben werden. Es gilt das *Prinzip von Le Chateliér:* Ein dynamisches Gleichgewicht hat die Tendenz, einer Änderung der Umgebungsbedingungen entgegenzuwirken.

Ist ein chemisches System nicht im Gleichgewicht, so nennt man

$$\prod_{i=1}^{m} c_i^{\nu_i} =: Q_C$$

den Reaktionsquotienten. Er beschreibt die Tendenz der Richtung, in der die Reaktion ablaufen kann: Ist $Q_C < K_C$, besteht die Tendenz zur Bildung von Produkten, bei $Q_C > K_C$ die Tendenz zur Bildung von Edukten, und für $Q_C = K_C$ ist das System im Gleichgewicht. Ob die Reaktion tatsächlich in der entsprechenden Richtung abläuft und wann sich der Endzustand einstellt, hängt von der Reaktionsgeschwindigkeit ab.

Weil Katalysatoren Hin- und Rückreaktionen beeinflussen, haben sie keine Auswirkungen auf die Gleichgewichtszusammensetzung.

Die Gleichgewichtskonstante hängt von der Temperatur ab.

2.3.2 Gleichgewichtskonstante und Reaktionsgeschwindigkeit

Da im chemischen Gleichgewicht Hin- und Rückreaktionen gleichschnell ablaufen, gilt für alle Elementarreaktionen, aus denen sich der Reaktionsmechanismus zusammensetzt

$$k_{hin} \cdot \prod_{Edukte} c_{g,i} = k_{rück} \cdot \prod_{Produkte} c_{g,i}$$

beziehungsweise

$$\frac{k_{hin}}{k_{rück}} = \frac{\prod_{Produkte} c_{g,i}}{\prod_{Edukte} c_{g,i}}.$$

Zusammensetzen für die Gesamtreaktion ergibt für den Gleichgewichtszustand:

$$\frac{k_{hin,1}}{k_{rück,1}} \cdot \frac{k_{hin,2}}{k_{rück,2}} \cdot \ldots = \frac{\prod_{Produkte,1} c_{g,i}}{\prod_{Edukte,1} c_{g,i}} \cdot \frac{\prod_{Produkte,2} c_{g,i}}{\prod_{Edukte,2} c_{g,i}} \cdot \ldots = \prod_{i=1}^{m} c_{g,i}^{\nu_i} = K_C.$$

Obwohl das Geschwindigkeitsgesetz einer Reaktion im allgemeinen nicht aus der Reaktionsgleichung bestimmt werden kann, kann man das Massenwirkungsgesetz daraus ableiten. Das gilt für alle chemischen Reaktionen.

2.4 Grundbegriffe der Thermodynamik

Viele chemische Reaktionen werden isotherm durchgeführt, das heißt die Temperatur des Reaktionsgemischs kann vom Experimentator konstant gehalten werden. Ist dies nicht möglich, zum Beispiel weil eine Reaktion mit starker Wärmeproduktion verbunden ist, muß man die Reaktionstemperatur mit den Gesetzen der Thermodynamik modellieren.

2.4.1 Wärme und Volumenarbeit

Zwei Energieformen spielen in der Thermodynamik eine große Rolle: Wärme und Volumenarbeit.

Unter *Wärme* versteht man Energie, die als Folge einer Temperaturdifferenz zwischen einem System und seiner Umgebung ausgetauscht wird. Der Wärmeinhalt Q eines System der Masse m und der Temperatur T ist

$$Q = m \cdot c_p \cdot T.$$

Dabei ist

$$m = \sum_{i=1}^{m} n_i \cdot M_i$$

die Gesamtmasse (Molmassen M_i) und

$$c_p = \sum_{i=1}^{m} \frac{m_i}{m} \cdot c_{p,i}$$

die *spezifische Wärme* des Systems. Bei einer Temperaturänderung ΔT beträgt also die Wärmeänderung

$$\Delta Q = m \cdot c_p \cdot \Delta T,$$

wenn keine Phasenänderungen auftreten.

Die *Volumenarbeit* wird verrichtet durch die Änderung des Volumens ΔV beim Druck p:

$$W = p \cdot \Delta V.$$

2.4.2 Innere Energie und Enhalpie

Die *innere Energie* ist die Summe aller Energieformen des Systems. Nach dem Energieerhaltungssatz ist diese für abgeschlossene Systeme konstant. Ist das System nicht von der Umgebung isoliert, bewirken Wärmeänderung und Volumenarbeit die Änderung der inneren Energie:

$$\Delta U = U_{Ende} - U_{Anfang} = \Delta Q - p \cdot \Delta V.$$

Viele Reaktionen werden in offenen Gefäßen bei konstantem Druck durchgeführt. Zur Beschreibung dieser Systeme betrachtet man die *Enthalpie*. Sie ist definiert durch

$$H = U + p \cdot V.$$

Für die *Enthalpieänderung* gilt dann

$$\Delta H = \Delta U + p \cdot \Delta V + \Delta p \cdot V = \Delta Q + \Delta p \cdot V$$

und bei konstantem Druck

$$\Delta H = \Delta Q.$$

Die Änderung der Enthalpie ΔH entspricht also bei konstantem Druck der Wärmeänderung.

2.4.3 Reaktionsenthalpie

Zur Beschreibung der Wärmeproduktion oder des Wärmeverbrauchs durch eine chemische Reaktion betrachtet man die *Reaktionsenthalpie*. Damit bezeichnet man die Änderung der Gesamtenthalpie des Systems bei einer chemischen Reaktion. Ist dabei $\Delta H < 0$, fließt Wärme in die Umgebung, die Reaktion heißt *exotherm*. Bei *endothermen* Reaktionen ist $\Delta H > 0$, Wärme fließt in das System hinein.

Die Reaktionsenthalpie bei Standardbedingungen (Druck von $1,01325\,bar$ und Temperatur von $298.15\,K$) ΔH_R^0 läßt sich aus den tabelliert vorliegenden Standardbildungsenthalpien ΔH_b^0 der Reaktionsteilnehmer berechnen:

$$\Delta H_R^0 = \sum_{i=1}^{m} \nu_i \cdot \Delta H_{b,i}^0.$$

Für die Temperaturabhängigkeit der Reaktionsenthalpie gilt

$$\Delta H_R = \Delta H_R^0 + \int_{T^0}^{T} m \cdot \Delta c_p(T) dT.$$

Eine chemische Reaktion mit der Reaktionslaufzahl ξ_i erzeugt bei konstantem Druck die *Reaktionswärme*

$$Q_{R,i} = -\Delta H_{R,i} \cdot \xi_i.$$

Die gesamte zeitliche Änderung der Reaktionswärme bei einem Reaktionssystem mit n_R Reaktionen ist dann

$$\dot{Q}_R(t) = \sum_{i=1}^{n_R} \frac{\mathrm{d}Q_{R,i}}{\mathrm{d}t} = -\sum_{i=1}^{n_R} \Delta H_{R,i} \cdot \frac{\mathrm{d}\xi_i}{\mathrm{d}t} = -\sum_{i=1}^{n_R} \Delta H_{R,i} \cdot V \cdot r_i.$$

Wärmeproduktion bzw. *Wärmeverbrauch* hängen also neben der Reaktionsenthalpie und dem Volumen auch von den Reaktionsgeschwindigkeiten und damit der Kinetik des Prozesses ab.

2.4.4 Wärmeproduktion und Wärmeaustausch

Außer durch die chemische Reaktion kann Wärme auch durch Austausch mit der Umgebung zu- oder abgeführt werden.

Viele Reaktionen sind mit einer Heiz- bzw. Kühlvorrichtung ausgestattet. Die Wärme wird über ein Übertragungsmedium ausgetauscht. Der *Wärmeaustausch* läßt sich dann durch die Gleichung

$$\dot{Q}_W = k_W \cdot A_W \cdot (T_W - T)$$

modellieren. k_W ist dabei der Wärmedurchgangskoeffizient, A_W die Wärmeaustauschfläche und T_W die Temperatur des Wärmeübertragungsmediums.

Werden Stoffe mit der Temperatur T_e und der Masseneintragsrate $\dot{m}_e$ in den Reaktor zugegeben, so bewirkt dies ebenfalls eine Wärmeänderung

$$\dot{Q}_e = \dot{m}_e \cdot c_{p,e} \cdot T_e.$$

2.4.5 Temperaturdifferentialgleichung

Aus diesen thermodynamischen Betrachtungen kann man eine Differentialgleichung für die Reaktionstemperatur ableiten:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\left(m \cdot c_p \cdot T\right) &= \dot{Q}_R + \dot{Q}_W + \dot{Q}_e \\ &= \left(-\sum_{i=1}^{n_R} \Delta H_{R,i} \cdot V \cdot r_i + k_W \cdot A_W \cdot (T_W - T) + \dot{m}_e \cdot c_{p,e} \cdot T_e \right) \end{aligned} \quad (2.1)$$

2.5 Chemische Reaktionstechnik

Chemische Reaktionen zu Produktions- oder Forschungszwecken, die kontrolliert ablaufen sollen, werden im allgemeinen in Reaktoren durchgeführt. Abbildung 2.2 zeigt das Schema eines chemischen Reaktors.

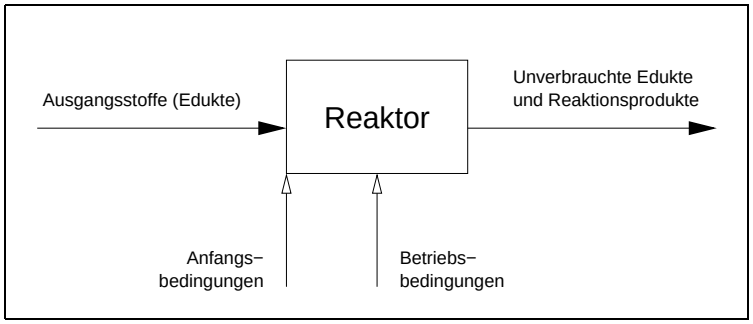


Abbildung 2.2: Schematische Darstellung eines chemischen Reaktors

Die Betriebsweise von Reaktoren wird unter anderem bestimmt durch

- die im Reaktor vorhandenen oder vorgelegten oder während der Reaktion zugeführten oder abgezogenen und an der Reaktion beteiligten *Komponenten* wie Edukte, Produkte, Lösungsmittel oder Katalysatoren,
- die *Phasenverhältnisse* des Reaktionsgemischs,
- die *Temperaturführung* des Prozesses,
- die *Reaktionsführung* und den
- *Reaktortyp*.

Auf Phasenverhältnisse, Temperaturführung, Reaktionsführung und Reaktortyp wollen wir in den folgenden Abschnitten eingehen.

2.5.1 Phasenverhältnisse

Man unterscheidet *homogene* einphasige Reaktionssysteme und *heterogene* Prozesse mit zwei, drei oder mehr Phasen.

Bei mehrphasigen Prozessen müssen die Stoffe zur Reaktion die Phasen wechseln. Dies geschieht an der sogenannten *Phasengrenzfläche* oder *Stoffaustauschfläche*.

Meistens gibt es eine zusammenhängende Phase (kontinuierliche Phase) und darin verteilte Phasen (disperse Phasen). Beispiele sind zerkleinerte Feststoffe in Gas oder Flüssigkeit, Flüssigkeitstropfen in Gas oder in anderer, nicht mischbarer Flüssigkeit und Gasblasen in Flüssigkeit.

2.5.2 Temperaturführung

Ursache für Temperaturänderungen sind Wärmeaustausch und Wärmeproduktion.

Man unterscheidet die folgenden Modi der Temperaturführung:

- *Isotherm:* Die Temperatur der Reaktionsmasse ist räumlich einheitlich und zeitlich konstant, produzierte Wärme wird momentan ausgetauscht.
- *Adiabatisch:* Es findet kein Wärmeaustausch mit der Umgebung statt.
- *Polytrop:* Die Temperaturänderung wird durch zeitliche oder räumliche Unterschiede von Wärmeproduktion und Wärmeaustausch hervorgerufen.

Bei der Modellierung kann im isothermen Fall die Temperatur als Steuerung oder Konstante auftreten. Für adiabatische oder polytrope Reaktionsführung formuliert man dagegen eine Differentialgleichung für die Temperatur, siehe Gleichnug (2.1), Steuerung ist im polytropen Fall gegebenfalls der Wärmeaustausch durch Kühlung oder Heizung.

2.5.3 Reaktionsführung

Das Zeitverhalten eines chemischen Prozesses kann

- *stationär* sein, das heißt alle Zustandsgrößen und Steuerungen des Reaktors (Volumenströme, Konzentrationen, Temperatur, Reaktionsvolumen) sind zeitlich konstant ($\frac{\mathrm{d}}{\mathrm{d}t}. = 0$), oder
- *instationär*, das heißt alle oder ein Teil dieser Größen sind zeitlich veränderlich.

Für stationäre und instationäre Prozesse können nun verschiedene Betriebsweisen angewendet werden:

- Im *kontinuierlich betriebenen Reaktor* wird der Zulaufstrom ständig zugeführt und der Ablaufstrom ständig entnommen.

Stationär und kontinuierlich ist der Standardbetrieb zur Produktion großer Mengenströme bestimmter Produkte in gleichbleibender Qualität. Instationärer Betrieb kann aber notwendig sein

1. zum Anfahren oder Abstellen eines Reaktors,
2. bei Veränderungen der Reaktorleistung,
3. bei Betriebsstörungen
4. oder gezielt eingesetzt werden, zum Beispiel zur Verbesserung von Umsatz oder Ausbeute.

Diskontinuierlich betriebene Reaktoren können auf die folgenden Arten gefahren werden:

- Beim *absatzweisen Betrieb (Batch-Betrieb)* werden alle Komponenten zum selben Zeitpunkt hinzugegeben und während der Reaktionsdauer nicht, das heißt es gibt keinen Zulauf- und Ablaufstrom. Man hat so immer ein instationäres Verhalten. Typische Produktionszyklen sind dabei: Füllen des Reaktors, Aufheizen auf Reaktionstemperatur, Durchführen der Reaktion mit Wärmetausch, Kühlen, Entleeren und Reinigen. Dies setzt sich einerseits zusammen aus der Reaktionsdauer für die chemische Umsetzung und andererseits aus der Rüstzeit für Vor- und Nachbereitung.
- Beim *halbkontinuierlichen Betrieb (Semi-Batch, Feedbatch)* werden gewisse Komponenten zum selbem Zeitpunkt vorgelegt, und während der Reaktionsphase werden zusätzlich gewisse Komponenten zu- oder abgeführt. Diese Reaktionsführung ist immer instationär und kann besonders variabel gestaltet werden.

Bei der Modellierung treten typischerweise Massen oder Molzahlen der Einwaagen als Steuerungen auf. Beim Semi-Batch-Betrieb können außerdem Zulauf- und Ablaufmengen und -profile gesteuert werden.

2.5.4 Reaktoren

Wir betrachten hier elementare Reaktortypen für einphasige Reaktionen. Es sollen nur die sogenannten idealen Reaktoren vorgestellt werden, auf die realen Schwierigkeiten wird hier nicht eingegangen

- Der *Rührkessel*, siehe Abbildung 2.3, ermöglicht eine gute Homogenisierung des Reaktorinhalts und die Intensivierung des Wärmeaustauschs.

  Mehrere Rührkessel können als *Rührkesselkaskade* hintereinandergeschaltet werden, dadurch können auch unterschiedliche Reaktionsbedingungen realisiert werden.

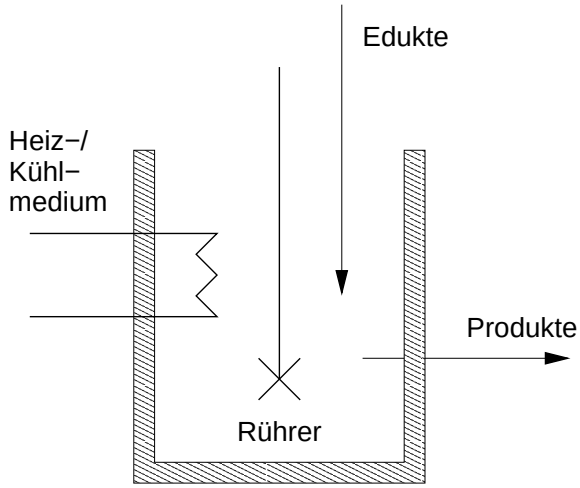


Abbildung 2.3: Schematische Darstellung eines Rührkessels

- In *Strömungsrohren*, siehe Abbildung 2.4, wird ein kontinuierlicher Eduktstrom zu- und ein kontinuierlicher Ausgangsstrom abgeführt. Es besteht ein größeres Verhältnis an Wärmeaustauschfläche zu Volumen, deshalb sind diese Reaktoren gut für hohen Druck und stark exotherme Reaktionen geeignet.

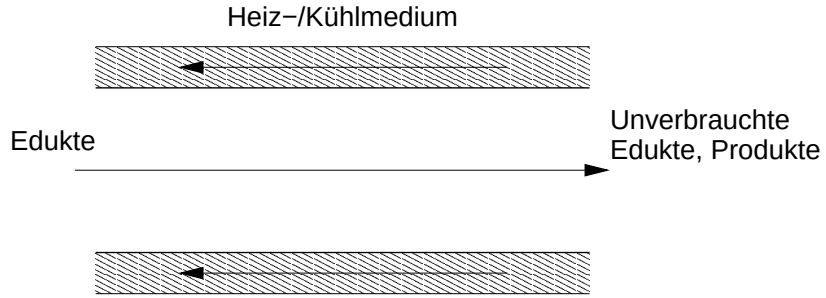


Abbildung 2.4: Schematische Darstellung eines Strömungsrohrs

2.6 Bilanzen

Bei chemischen Reaktionen gelten verschiedene Erhaltungssätze. Diese können verwendet werden, um Bilanzen aufzustellen, die als Differentialgleichungen oder algebraische Gleichungen in die Modellierung eingehen.

Es gilt das folgende *Prinzip der Bilanzierung:* Die zeitliche Änderung der im Bilanzraum vorhandenen Menge wird dadurch verursacht, daß Ströme durch die Oberfläche des Bilanzraums ein- und austreten und daß im Bilanzraum eine Produktion durch chemische Reaktion stattfindet.

Daraus ergibt sich die allgemeine *Bilanzgleichung*

$$\begin{pmatrix} \text{Zeitliche Änderung} \\ \text{der Menge} \end{pmatrix} = \begin{pmatrix} \text{Mengen-} \\ \text{strom ein} \end{pmatrix} - \begin{pmatrix} \text{Mengen-} \\ \text{strom aus} \end{pmatrix} + \begin{pmatrix} \text{Produktion durch} \\ \text{chemische Reaktion} \end{pmatrix}.$$

Genauer wollen wir hier die wichtigsten Bilanzen betrachten, und zwar

- Gesamtmassenbilanz,
- Stoffmengenbilanz für die Spezies A_i, $i = 1, \ldots, m$ und
- Wärmebilanz.

Darüber hinaus müssen in speziellen Fällen noch berücksichtigt werden:

- Die Gesamtenergiebilanz als Verallgemeinerung der Wärmebilanz, sie berücksichtigt zum Beispiel auch Strömungsenergien, potentielle und kinetische Energien, falls diese relevant sind,
- die Bilanz der elektrischen Ladung bei elektrochemischen Prozessen,
- die Impulsbilanz oder
- die Bilanz der Lichtenergie.

2.6.1 Gesamtmassenbilanz

Bei chemischen Reaktionen wird die Masse erhalten, daher gilt:

$$\frac{\mathrm{d}m}{\mathrm{d}t} = \dot{m}_e - \dot{m}_a,$$

das heißt die Änderung der Masse ergibt sich durch die Masseneintragsrate der Zuläufe minus der Massenaustragsrate der Abläufe.

2.6.2 Stoffmengenbilanz

Aufgrund der Teilchenerhaltung kann für jede im Reaktionsgemisch vorhandene Spezies eine Stoffmengenbilanz aufgestellt werden:

$$\frac{\mathrm{d}n_i}{\mathrm{d}t} = \dot{n}_{i,e} - \dot{n}_{i,a} + \dot{n}_{i,R}, \quad i = 1, \ldots, m.$$

Bemerkung 2.6.1 *Die Gesamtmassenbilanz folgt aus den einzelnen Komponentenmengenbilanzen: Multiplikation mit den Molmassen ergibt*

$$M_i \cdot \frac{dn_i}{dt} - M_i \cdot \dot{n}_{i,e} + M_i \cdot \dot{n}_{i,a} = M_i \cdot \dot{n}_{i,R}, \quad i = 1, \ldots, m,$$

aufsummiert folgt

$$\begin{aligned} \frac{dm}{dt} - \dot{m}_e + \dot{m}_a &= \sum_{i=1}^{m} M_i \cdot \frac{dn_i}{dt} - \sum_{i=1}^{m} M_i \cdot \dot{n}_{i,e} + \sum_{i=1}^{m} M_i \cdot \dot{n}_{i,a} \\ &= \sum_{i=1}^{m} M_i \cdot \dot{n}_{i,R} = V \cdot r \cdot \sum_{i=1}^{m} M_i \cdot \nu_i = 0. \end{aligned}$$

Die Summe der Molmassen multipliziert mit den stöchiometrischen Koeffizienten ist dabei gleich null wegen der Erhaltung der chemischen Elemente.

In integrierter Form lautet die Stoffmengenbilanz

$$\Delta n = n - n_0 = \Delta n_R + n_e - n_a.$$

Der Quellterm aufgrund der chemischen Reaktion ist

$$\Delta n_R = n - n_0 - n_e + n_a = \nu^T \xi$$

mit dem Reaktionsfortschrittvektor

$$\xi = \begin{pmatrix} \xi_1 \\ \vdots \\ \xi_n \end{pmatrix}.$$

Sei $r := m - \text{Rang } \nu$ und $\epsilon \in I\!R^{r \times m}$ eine vollrangige Matrix mit

$$\epsilon \cdot \nu^T = 0.$$

Dann nennt man ϵ *Bilanzmatrix*, und es gilt:

$$0 = \epsilon \cdot \nu^T \xi = \epsilon \cdot \Delta n_R.$$

2.6.3 Wärmebilanz

Für die Wärmebilanz gilt

$$\frac{\mathrm{d}}{\mathrm{d}t} (m \cdot c_p \cdot T) = \dot{Q}_e + \dot{Q}_W + \dot{Q}_R.$$

Dabei ist $\dot{Q}_e$ die Wärmeänderung durch die Einträge, $\dot{Q}_W$ der Wärmeaustausch und $\dot{Q}_R$ die Wärmeproduktion oder der Wärmeverbrauch durch die Reaktion, siehe Abschnitt 2.4.

2.6.4 Spezialfälle in den Bilanzgleichungen

Bei *stationären* Prozessen sind alle Zeitableitungen gleich null, daher gilt:

$$\frac{\mathrm{d}m}{\mathrm{d}t} = 0, \qquad \frac{\mathrm{d}n_i}{\mathrm{d}t} = 0, \quad i = 1, \ldots, m, \qquad \frac{\mathrm{d}}{\mathrm{d}t}\left(m \cdot c_p \cdot T\right) = 0.$$

Bei *Batchprozessen* gibt es keinen Ein- und Austrag:

$$\dot{m}_e = \dot{m}_a = 0, \qquad \dot{n}_{i,e} = \dot{n}_{i,a} = 0, \qquad \dot{Q}_e = \dot{Q}_a = 0.$$

Bei *isothermen* Reaktionen ist

$$T = konst.$$

Wird der Prozeß *adiabatisch* durchgeführt, entfällt der Wärmeaustausch:

$$\dot{Q}_W = 0.$$

Und findet schließlich *keine Reaktion* statt, so ist

$$\dot{n}_{i,R} = 0, \quad i = 1, \ldots, m, \qquad \dot{Q}_R = 0.$$

2.7 Zusammenfassung

Die Gesetze der Physikalischen Chemie für Reaktionskinetik, chemische Gleichgewichte, Thermodynamik und Reaktionstechnik ermöglichen es, chemische Prozesse auf mathematische Modelle abzubilden. In dieser Arbeit betrachten wir solche Prozesse, die eine Modellierung mittels Differentiell-Algebraischer Gleichungssysteme (DAEs) ermöglichen. Diese DAEs sind im allgemeinen nichtlinear, sowohl in den Zustandsvariablen als auch in den Modellparametern und in den Steuerungen. Da bei Reaktionssytemen die Geschwindigkeiten der verschiedenen Einzelreaktionen auf sehr unterschiedlichen Zeitskalen liegen können, sind Anfangswertprobleme für solche DAEs in der Regel steif [63].

Im nächsten Kapitel werden wir uns mit Verfahren zur Lösung von steifen DAEs, also zur Simulation chemischer Prozesse, beschäftigen. Darauf aufbauend werden anschließend Methoden behandelt, um die Modellparameter aus experimentellen Daten zu schätzen. Der optimale Entwurf von Experimenten zur möglichst signifikanten Schätzung der Parameter ist schließlich Gegenstand der Optimalen Versuchsplanung.

In Kapitel 10 schließen wir dann den Kreis durch Anwendung der Methoden zur Simulation, Parameterschätzung und Optimalen Versuchsplanung auf Beispiele aus der chemischen Reaktionskinetik und Verfahrenstechnik.

Obwohl in dieser Arbeit der Schwerpunkt auf chemischen Reaktionssystemen liegt, erlauben die hier vorgestellten Methoden aber darüber hinaus die Behandlung beliebiger Prozesse, die durch DAEs modelliert werden können.

Kapitel 3

Numerische Methoden für Differentiell-Algebraische Gleichungen

Die Modellierung naturwissenschaftlicher Prozesse führt oft auf Systeme gewöhnlicher Differentialgleichungen. In vielen Fällen sind dabei die Zustandsvariablen Nebenbedingungen unterworfen, zum Beispiel durch geometrische Einschränkungen, Erhaltungssätze oder Invarianten. Dadurch können die Trajektorien nicht den gesamten Zustandsraum durchlaufen, sondern bewegen sich auf Mannigfaltigkeiten. Ältere Modellierungsansätze versuchen, durch entsprechende Variablenwahl diese Nebenbedingungen zu eliminieren und nur die gewöhnlichen Differentialgleichungen (ODEs) zu untersuchen. Dies führt oft auf Systeme, die wegen starker Nichtlinearität numerisch nur mit sehr hohem Rechenaufwand zu lösen sind. Der Ansatz in diesem Kapitel geht den umgekehrten Weg. Die algebraischen Nebenbedingungen werden als Teil des dynamischen Systems verstanden und gemeinsam mit den Differentialgleichungen behandelt. Diese Systeme nennt man *Differentiell-Algebraische Gleichungen* (DAEs).

Ziel dieses Kapitels ist es, einen Überblick über die Methoden zu geben, die wir verwenden, um Lösungen von Anfangswertproblemen für DAE-Systeme numerisch zu berechnen. Insbesondere soll dargestellt werden, warum die Verfahren der Backward Differentiation Formulae (BDF) besonders geeignet sind.

Die Lösung von DAE-Anfangswertproblemen ist nicht nur zur Simulation der modellierten Prozesse erforderlich, sie tritt auch als der Teilaufgabe in Algorithmen zur Parameterschätzung und Optimalen Versuchsplanung auf, siehe Kapitel 4 und 5. Darüber hinaus werden dort auch Ableitungen der Lösung der Anfangswertprobleme nach Anfangswerten, Parametern und Steuergrößen benötigt. Deren Berechnung wird in Kapitel 6 behandelt.

Über die numerische Behandlung gewöhnlicher Differentialgleichungen gibt es zahlreiche Lehrbücher, zum Beispiel [2], [34], [61], [63] oder [109]. Methoden für Differentiell-Algebraische Gleichungssysteme werden in [31] oder [63] behandelt.

Das weitverbreitetste Programm zur Lösung von steifen Index-1-DAEs ist DASSL [90], [31], ein BDF-Verfahren. Wir verwenden die Software DAESOL aus unserer Arbeits-

gruppe [41], [28], [17]. Sie verwendet ebenfalls ein BDF-Verfahren mit automatischer Schrittweiten- und Ordnungssteuerung. Zum Einsatz in der Optimalen Versuchsplanung wurde von Bauer [15], [14] eine spezielle Variante implementiert, die mittels Interner Numerischer Differentiation alle benötigten ersten und zweiten Ableitungen bereitstellen kann, siehe auch Abschnitt 6.2.7 in Kapitel 6.

3.1 Verschiedene Formen Differentiell-Algebraischer Gleichungen

In der allgemeinsten, sogenannten *voll-impliziten nichtlinearen Form* läßt sich eine Differentiell-Algebraische Gleichung (erster Ordnung) schreiben als

$$F(t, \dot{x}, x) = 0, \tag{3.1}$$

wobei t aus einem reellen Intervall $I = [t_0; t_{end}]$, x aus aus dem Raum der Abbildungen von I nach $\mathbb{R}^n$ und F eine Abbildung von $I \times \mathbb{R}^n \times \mathbb{R}^n$ in den $\mathbb{R}^m$ ist.

Bei der in dieser Arbeit betrachteten Modellierung chemischer Reaktionssysteme tritt typischerweise der Spezialfall der *semi-impliziten quasilinearen DAE* auf:

$$\begin{aligned} A(t, y, z)\, \dot{y} &= f(t, y, z) && (3.2)\\ 0 &= g(t, y, z) && (3.3) \end{aligned}$$

mit $A : I \times \mathbb{R}^n \to \mathbb{R}^{n_y \times n_y}$, $f : I \times \mathbb{R}^n \to \mathbb{R}^{n_y}$, $g : I \times \mathbb{R}^n \to \mathbb{R}^{n_z}$.

Die Matrix A werde dabei stets ohne Beschränkung der Allgemeinheit als regulär angenommen, dann heißen

- (3.2) die *Differentialgleichung* und $y(t) \in \mathbb{R}^{n_y}$ die *differentiellen Variablen* und
- (3.3) die *algebraische Gleichung* und $z(t) \in \mathbb{R}^{n_z}$ die *algebraischen Variablen*.

Die Zustandsvariablen insgesamt werden oft als x zusammengefaßt:

$$x(t) := (y(t), z(t)) \in \mathbb{R}^n, \quad n := n_y + n_z.$$

Ein Spezialfall der semi-impliziten quasilinearen Form ist die *semi-explizite Form*:

$$\begin{aligned} \dot{y} &= f(t, y, z)\\ 0 &= g(t, y, z) \end{aligned}$$

Bemerkung 3.1.1 *[31] Eine voll-implizite lineare DAE mit konstanten Koeffizienten kann man durch konstante Koordinaten-Transformationen stets in semi-explizite lineare Form mit konstanten Koeffizienten bringen.*

3.2 Der Index einer DAE

Bei der Klassifizierung und der Untersuchung des Verhaltens von DAEs spielt der *Index* eine entscheidende Rolle. Er gibt an, wie oft man die DAE differenzieren muß, um eine ODE zu erhalten.

Wenn man die algebraische Gleichung (3.3) einer semi-impliziten DAE nach t differenziert, ergibt sich

$$0 = g_t(t, y, z) + g_y(t, y, z)\dot{y} + g_z(t, y, z)\dot{z}.$$

Wenn g_z regulär ist, kann man nach $\dot{z}$ auflösen

$$\dot{z} = g_z(t, y, z)^{-1} \left(g_t(t, y, z) + g_y(t, y, z)\dot{y}\right)$$

und erhält zusammen mit (3.2) eine gewöhnliche Differentialgleichung (ODE) für $\dot{y}$ und $\dot{z}$. In diesem Fall sagt man, die DAE (3.2)-(3.3) habe den *Index 1*. Allgemeiner definiert man den Index einer DAE durch die minimale Anzahl von Differentiationen, die man braucht, um die DAE in eine ODE überzuführen. Eine ODE hat entsprechend den Index 0.

Definition 3.2.1 *Die minimale Anzahl von Differentiationen von (3.1) nach t, die man benötigt, um $\dot{x}$ als stetige Funktion von (t, x) zu erhalten, nennt man den* Index *(*Differentiationsindex*) von (3.1).*

Bemerkung 3.2.2 *Die Lösung einer Index-r-DAE erfüllt also auch eine gewöhnliche Differentialgleichung*

$$\dot{x} = G(t, x),$$

wobei G partielle Ableitungen von F bis zum Grad r enthalten kann.

3.3 Anfangswertprobleme für Index-1-DAEs, konsistente Anfangswerte

Wir betrachten nun *Anfangswertprobleme* für DAEs, die in der semi-impliziten quasilinearen Form (3.2)-(3.3) gegeben sind.

Weiterhin nehmen wir an, daß der Index der DAE gleich 1 ist, das heißt

$$g_z \text{ ist regulär.} \tag{3.4}$$

Es müssen nur für die differentiellen Variablen Anfangswerte angegeben werden

$$y(t_0) = y_0, \tag{3.5}$$

da auch im Anfangspunkt t_0 die algebraischen Gleichungen (3.3) erfüllt sein müssen:

$$g(t_0, y_0, z(t_0)) = 0. \tag{3.6}$$

Dadurch sind die Werte $z(t_0)$ wegen der Index-1-Bedingung (3.4) (lokal) eindeutig festgelegt. Bedingung (3.6) nennt man *Konsistenz*.

Die Gleichungen (3.2), (3.3) und (3.5) bilden also zusammen ein Anfangswertproblem für eine Index-1-DAE.

3.3.1 Konsistente Anfangswerte

Bestimmung konsistenter Anfangswerte

Zur Bestimmung geeigneter konsistenter Anfangswerte für Index-1-Probleme muß die nichtlineare Gleichung (3.6) gelöst werden. Dies kann prinzipiell mit einem Newtonverfahren erfolgen. Bei stark nichtlinearen Gleichungen ist es aber schwer, geeignete Startwerte für das Newtonverfahren anzugeben. Mit globalisierten Newtonverfahren oder Homotopieverfahren versucht man, dieses Problem in den Griff zu bekommen. Für eine detaillierte Darstellung siehe [28].

Integration mit inkonsistenten Anfangswerten

In dieser Arbeit betrachten wir Optimierungsprobleme, bei denen Index-1-DAE-Systeme als Nebenbedingungen auftreten. Wir lösen diese Probleme mit iterativen Algorithmen, in denen auch immer wieder DAE-Anfangswertprobleme gelöst werden müssen. Es ist dabei nicht sinnvoll, jedesmal vor der Integration die Konsistenz herzustellen. Daher gehen wir gemäß [27] anders vor. Wir betrachten eine relaxierte Variante des Anfangswertproblems

$$\begin{aligned} A(t, y, z)\, \dot{y} &= f(t, y, z) \\ 0 &= g(t, y, z) - \beta(t) \cdot g(t_0, y_0, z_0) \\ & \quad y(t_0) = y_0,\ z(t_0) = z_0. \end{aligned}$$

$\beta(t)$ wählt man so, daß $\beta(t_0) = 1$ ist, zum Beispiel

$$\beta(t) = e^{-\alpha(t-t_0)}, \quad \alpha \geq 0.$$

Das so formulierte Problem ist für jede Wahl von y_0 und z_0 automatisch konsistent.

Die ursprüngliche Konsistenzbedingung

$$g(t_0, y_0, z_0) = 0 \tag{3.7}$$

wird als Nebenbedingung des Optimierungsproblems formuliert. Gemeinsam mit der Lösung des Optimierungsproblems werden dann Werte für z_0 bestimmt, die (3.7) erfüllen.

3.4 Die Zustandsform einer Index-1-DAE

Durch Anwendung des Satzes für implizite Funktionen lassen sich bei semi-impliziten Index-1-DAEs die algebraischen Variablen z eindeutig als Funktion der differentiellen Variablen y schreiben:

$$z = \phi(y).$$

Einsetzen in Gleichung (3.2) liefert die sogenannte *Zustandsform* der DAE

$$A(t, y, \phi(y))\ \dot{y} = f(t, y, \phi(y)), \tag{3.8}$$

die im Prinzip eine gewöhnliche Differentialgleichung ist.

Somit übertragen sich alle Konvergenzresultate für numerische Integrationsverfahren für gewöhnliche Differentialgleichungen auf semi-implizite DAEs vom Index 1, sofern die Diskretisierung des DAE-Systems äquivalent ist zu einer Diskretisierung der Gleichung (3.8) zusammen mit

$$g(t, y, \phi(y)) = 0.$$

Diese Überlegung gestattet es, zur Lösung von Index-1-DAEs Verfahren für gewöhnliche Differentialgleichungen zu verwenden. Der nächste Abschnitt gibt einen kurzen Abriß der Theorie über lineare Mehrschrittverfahren mit dem Ziel der Herleitung der für steife Probleme geeigneten BDF-Verfahren. Im anschließenden Abschnitt werden wir diese Verfahren dann auf Index-1-DAE-Probleme anwenden.

3.5 Lineare Mehrschrittverfahren

3.5.1 Aufgabenstellung

Ziel der hier diskutierten Verfahren ist es, die Lösung $x \in C^1([t_0, t_{end}], I\!R^n)$ eines Anfangswertproblems mit einer gewöhnlichen Differentialgleichung

$$\dot{x} = f(t, x), \qquad x(t_0) = x_0 \tag{3.9}$$

auf einem *Gitter*

$$\Delta = \{t_0, t_1, \ldots, t_N\}$$

durch eine *Gitterfunktion*

$$x_\Delta : \Delta \to I\!R^n : t_j \mapsto x_\Delta(t_j)$$

zu approximieren. Wir schreiben kurz

$$x_j := x_\Delta(t_j) \text{ und } f_j := f_\Delta(t_j) := f(t_j, x_\Delta(t_j)) \text{ für } t_j \in \Delta.$$

Zunächst betrachten wir *äquidistante Gitter*, das heißt

$$t_j = t_0 + jh, \quad j = 0, \ldots, N \text{ mit } h > 0.$$

Das Gitter Δ nennen wir dann auch $\Delta(h)$.

3.5.2 Definition der Verfahren

Wir wollen uns im folgenden mit linearen Mehrschrittverfahren befassen.

Während bei Einschrittverfahren, siehe zum Beispiel [109, 34], nur Informationen aus dem letzten Schritt benutzt werden, um den Wert der Gitterfunktion an einem neuen Punkt zu berechnen, greifen Mehrschrittverfahren weiter in die Vergangenheit zurück. Sie berechnen den aktuellen Wert aus den Werten der letzten k Iterationen. Auch frühere Funktionswerte der rechten Seite können dabei berücksichtigt werden. Dadurch sind Mehrschrittverfahren sehr effizient, was den Aufwand der Funktionsauswertungen angeht.

Ein *lineares Mehrschrittverfahren (k-Schrittverfahren)* auf einem äquidistanten Gitter ist gegeben durch eine Rekursion der Form

$$\sum_{j=0}^{k} \alpha_j \, x_{n+j} = h \sum_{j=0}^{k} \beta_j \, f_{n+j}. \tag{3.10}$$

Das Verfahren heißt *explizit*, wenn $\beta_k = 0$, sonst *implizit*. Sinnvoll ist nur eine Wahl der Koeffizienten mit $|\alpha_0| + |\beta_0| > 0$, da es sich sonst um ein $(k-1)$-Schrittverfahren handelt. α_k wird immer $\neq 0$ gewählt (oder ohne Beschränkung der Allgemeinheit $\alpha_k = 1$), denn explizite Verfahren wären sonst $(k-1)$-Schrittverfahren, für implizite Verfahren ist diese Voraussetzung wichtig, um die Existenz von Gitterlösungen zu garantieren:

Lemma 3.5.1 *[34] Die Abbildung f genüge der Lipschitzbedingung*

$$|f(t,x) - f(t,\bar{x})| \leq L|x - \bar{x}| \quad \forall x, \bar{x} \in I\!R^n, \; t \in I\!R$$

mit $L \geq 0$. Dann existiert für

$$h < \frac{|\alpha_k|}{|\beta_k| L} \tag{3.11}$$

zu beliebigen Startwerten $x_\Delta(t_0)$, ..., $x_\Delta(t_{k-1})$ eine eindeutige Gitterfunktion x_Δ, welche die Rekursion des Mehrschrittverfahrens (3.10) erfüllt.

Beweis. Für explizite Verfahren mit $\beta_k = 0$ ist die Aussage klar. Sei also nun $\beta_k \neq 0$. Zur Berechnung von x_{n+k} ist die folgende Fixpunktgleichung zu lösen:

$$x_{n+k} = h \frac{\beta_k}{\alpha_k} f_{n+k} + R_n, \tag{3.12}$$

wobei R_n nur Terme enthält, die nicht von x_{n+k} abhängen. Die rechte Seite genügt nach Voraussetzung einer Lipschitzbedingung mit der Lipschitzkonstanten

$$\bar{L} = h \frac{|\beta_k|}{|\alpha_k|} L.$$

Nach dem Banachschen Fixpunktsatz konvergiert eine Fixpunktiteration für (3.12), wenn $\bar{L} < 1$. Dies ist äquivalent zu (3.11). Induktiv können dann eindeutige Werte für x_Δ berechnet werden. ∎

Die *charakteristischen Polynome* ρ und σ des Mehrschrittverfahrens sind definiert durch

$$\rho(\xi) := \sum_{j=0}^{k} \alpha_j \, \xi^j, \qquad \sigma(\xi) := \sum_{j=0}^{k} \beta_j \, \xi^j.$$

Man nimmt immer an, daß sie teilerfremd sind, denn hätten sie bei einem k-Schrittverfahren einen größten gemeinsamen Teiler vom Polynomgrad $\mu > 0$, beschrieben sie ein reduziertes $(k - \mu)$-Schrittverfahren.

Man definiert nun einen *Shiftoperator* E durch

$$(E\phi)(t_j) = \phi(t_{j+1}) \text{ bzw. } (E\phi)_j = \phi_{j+1}.$$

Damit kann man das Mehrschrittverfahren (3.10) in einer sehr kompakten Notation schreiben:

$$\rho(E) \, x_\Delta = h \, \sigma(E) \, f_\Delta.$$

3.5.3 Konsistenz, Stabilität und Konvergenz

Um die Qualität eines Diskretisierungsverfahrens für gewöhnliche Differentialgleichungen zu charakterisieren, sind drei Eigenschaften entscheidend: Konsistenz, Stabilität und Konvergenz.

Konsistenz

Der Begriff der *Konsistenz* beschreibt, wie sich das Verfahren verhält, wenn man die Lösung $x(t)$ des Anfangswertproblems statt der Gitterfunktion in das Diskretisierungsschema (3.10) einsetzt.

Definition 3.5.2 *Die* Konsistenzordnung *eines linearen Mehrschrittverfahrens (3.10) ist p, wenn für den* Konsistenzfehler *gilt:*

$$L(x, t, h) := \rho(E) \, x(t) - h \, \sigma(E) \, \dot{x}(t) = O(h^{p+1})$$

für alle $x \in C^\infty([t_0, t_{end}], \mathbb{R}^n)$ gleichmäßig für alle t und h mit $[t, t + kh] \subset [t_0, t_{end}]$.

Der folgende Satz faßt einige Konsistenzkriterien zusammen.

Satz 3.5.3 *[34] Ein lineares Mehrschrittverfahren hat genau dann die Konsistenzordnung p, wenn eine der folgenden äquivalenten Bedingungen erfüllt ist:*

1. *Für beliebiges* $x \in C^\infty([t_0, t_{end}], \mathbb{R}^n)$ *gilt*

$$L(x, t, h) = O(h^{p+1})$$

gleichmäßig in allen t, h *mit* $[t, t + kh] \subset [t_0, t_{end}]$.

2. *Für alle Polynome* P *vom Grad* p *gilt*

$$L(P, 0, h) = 0.$$

3. *Für die Exponentialfunktion gilt:*

$$L(\exp, 0, h) = \rho(e^h) - h\,\sigma(e^h) = O(h^{p+1}).$$

4. *Es gilt*

$$\sum_{j=0}^{k} \alpha_j = 0, \qquad \sum_{j=0}^{k} \alpha_j j^l = l \sum_{j=0}^{k} \beta_j j^{l-1} \text{ für } l = 1, \ldots, p.$$

Stabilität

Bei Mehrschrittverfahren wird die Lösung eines Anfangswertproblems erster Ordnung durch eine Lösungskomponente einer Differenzengleichung höherer Ordnung approximiert. Es gibt aber noch $k-1$ weitere Lösungskomponenten, die sogenannten *Nebenlösungen* oder *parasitären Lösungen*. Es muß sichergestellt werden, daß diese auch für kleine Schrittweiten die Hauptlösung nicht überwuchern.

Stabilität bedeutet, daß sich kleine Störungen der Anfangswerte nicht stark auf die Lösung des Diskretisierungsverfahrens auswirken.

Dies führt bei linearen Mehrschrittverfahren zur folgenden Definition.

Definition 3.5.4 *Ein lineares Mehrschrittverfahren heißt stabil, wenn die lineare homogene Differenzengleichung*

$$\rho(E)\, x_\Delta = 0 \tag{3.13}$$

stabil ist.

Lemma 3.5.5 (Dahlquistsche Wurzelbedingung) *[34] Die Gleichung (3.13) ist genau dann erfüllt, wenn jede Wurzel* ξ *des Polynoms* ρ *der Bedingung* $|\xi| \leq 1$ *genügt mit* $|\xi| = 1$ *nur für einfache Nullstellen.*

Konvergenz

Definition 3.5.6 *Man sagt, ein Mehrschrittverfahren* konvergiert *gegen die Lösung*

$$x \in C^1([t_0, t_{end}], \mathbb{R}^n)$$

eines Anfangswertproblems (3.9), wenn

$$\lim_{h\to 0} x_{\Delta(h)}(t) = x(t) \quad \forall t \in \Delta(h) \cap [t_0, t_{end}],$$

gilt, sobald die Startwerte

$$\lim_{h\to 0} x_{\Delta(h)}(t_0 + jh) = x_0, \quad j = 0, \ldots, k-1.$$

erfüllen. Wenn ein Mehrschrittverfahren für beliebige Anfangswertprobleme konvergiert, heißt es konvergent.

Die beiden folgenden Sätze formulieren den „Hauptsatz der Numerischen Mathematik“[34] für lineare Mehrschrittverfahren:

$$\text{Konsistenz und Stabilität} \iff \text{Konvergenz.}$$

Satz 3.5.7 *Ein konvergentes lineares Mehrschrittverfahren ist stabil und konsistent, insbesondere gilt*

$$\rho'(1) = \sigma(1) \neq 0.$$

Satz 3.5.8 *Ein stabiles und konsistentes lineares Mehrschrittverfahren ist konvergent. Genauer: Sei ein Anfangswertproblem (3.9) mit rechter Seite $f \in C^p$ gegeben. Ist die Konsistenzordnung des linearen Mehrschrittverfahrens p und der Fehler der Startwerte ϵ_0, so gilt für den lokalen Diskretisierungsfehler*

$$\|x(t) - x_{\Delta(h)}(t)\| \leq C(\epsilon_0 + h^p) \quad \forall t \in \Delta(h),$$

falls $h \leq h_*$ und $\epsilon_0 \leq \epsilon_*$ mit Konstanten $C, h_*\epsilon_* > 0$.

Die asymptotische Konvergenzrate hängt also nicht nur von der Konsistenzordnung ab, sondern auch von der Genauigkeit der Startwerte.

3.5.4 Stabilitätsgebiete und Steifheit

Wir betrachten nun eine lineare Differentialgleichung

$$\dot{x} = Ax \tag{3.14}$$

mit $A \in I\!R^{n\times n}$. Wir wollen wissen, wie sich ihr Stabilitätsverhalten auf das des Mehrschrittverfahrens (3.10) überträgt.

Setzt man (3.14) in (3.10) ein, so ergibt sich:

$$\rho(E)\, x_\Delta = h\, \sigma(E)\, A x_\Delta. \tag{3.15}$$

Wir nehmen nun an, daß die Matrix A diagonalisierbar ist. Für nicht diagonalisierbare Matrizen können analoge Untersuchungen durchgeführt werden, siehe [34]. Also:

$$A = U^{-1}\Lambda U \quad \text{mit } \Lambda = \mathrm{diag}(\lambda_i,\ i = 1, \ldots, n), \quad U \in GL(n).$$

Eingesetzt in (3.15)

$$\rho(E)\, U x_\Delta = h\, \sigma(E)\, \Lambda U x_\Delta$$

erhält man entkoppelte Differenzengleichungen ($\bar{x} := Ux$):

$$\rho(E)\, \bar{x}_{\Delta,j} = h\, \sigma(E)\, \lambda_j \bar{x}_{\Delta,j}, \quad j = 1, \ldots, n.$$

Also reicht es aus, die Differenzengleichung für irgendeinen herausgegriffenen Eigenwert $\lambda = \lambda_j$ mit $X = \bar{x}_{\Delta,j}$ zu betrachten:

$$(\rho(E) - h\lambda\, \sigma(E))X = 0$$

bzw. mit $z := h\lambda$ und $\rho_z := \rho - z\sigma$:

$$\rho_z(E)X = 0.$$

Das *Stabilitätsgebiet* $\mathcal{S}$ des Mehrschrittverfahrens wird nun folgendermaßen definiert:

$$\mathcal{S} = \{z \in \mathbb{C} :\ \rho_z(E)X = 0 \text{ ist eine stabile Differenzengleichung}\}.$$

Wegen $\rho_0 = \rho$ und Definition 3.5.4 gilt:

$$0 \in \mathcal{S} \iff \text{Das Mehrschrittverfahren ist stabil.}$$

Man spricht daher auch von *nullstabil*.

Die analytische Lösung des Anfangswertproblems

$$\dot{x} = Ax, \qquad x(0) = x_0 \tag{3.16}$$

ist durch die Exponentialfunktion gegeben:

$$x(t) = \exp(tA)\, x_0.$$

Sie hat das Stabilitätsgebiet

$$\{z \in \mathbb{C} : \ |\exp(z)| \leq 1\} = \mathbb{C}^-.$$

Das Ziel sind nun Mehrschrittverfahren, deren Stabilitätsgebiet dem der Exponentialfunktion möglichst nahe kommt. Diese nennt man A-stabil.

Definition 3.5.9 *Ein lineares Mehrschrittverfahren heißt* A-stabil*, wenn*

$$\mathbb{C}^- \subset \mathcal{S}.$$

Es gilt jedoch:

Satz 3.5.10 (Zweite Dahlquist-Schranke) *[34] Ein A-stabiles lineares Mehrschrittverfahren besitzt notwendigerweise die Konsistenzordnung*

$$p \leq 2.$$

Daher betrachtet man eine Abschwächung der A-Stabilität, und zwar, daß nicht die gesamte negative Halbebene in $\mathcal{S}$ enthalten ist, sondern nur ein (möglichst großer) Winkelbereich:

Definition 3.5.11 *Ein lineares Mehrschrittverfahren heißt* $A(\alpha)$-stabil*, wenn*

$$\mathcal{S}_\alpha := \{z \in \mathbb{C} : \ |\arg(-z)| \leq \alpha\} \subset \mathcal{S}$$

mit $\alpha \in [0; \frac{\pi}{2}]$.

Aus der Gestalt des Stabilitätsgebiets ergeben sich die Schrittweiten, die eine Vererbung der Stabilität gestatten:

Satz 3.5.12 *[34] Die Schrittweite h zur Vererbung der Stabilität des Anfangswertproblems (3.16) an die durch ein stabiles Mehrschrittverfahren gegebene Differenzengleichung genügt der Abschätzung*

$$h^- \leq h \leq h^+$$

mit den Grenzen

$$h^- = \min_{\substack{\lambda \ EW \ von \ A \\ \lambda \neq 0}} \frac{r_{\mathcal{S}}^-(\arg \lambda)}{|\lambda|}, \qquad h^+ = \min_{\substack{\lambda \ EW \ von \ A \\ \lambda \neq 0}} \frac{r_{\mathcal{S}}^+(\arg \lambda)}{|\lambda|}$$

und den Stabilitätsradien (siehe Abbildung 3.1)

$$\begin{aligned} r_{\mathcal{S}}^-(\phi) &= \sup\left\{R : \ re^{i\phi} \in int(\mathcal{S}) \quad \forall\, r \in (0; R]\right\}, \\ r_{\mathcal{S}}^+(\phi) &= \sup\left\{R : \ re^{i\phi} \in \mathcal{S} \quad \forall\, r \in [0; R]\right\}. \end{aligned}$$

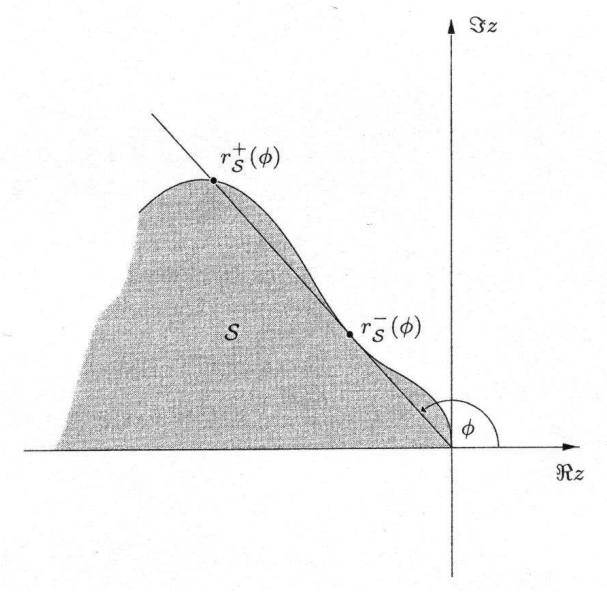


Abbildung 3.1: Veranschaulichung der Stabilitätsradien von linearen Mehrschrittverfahren [34]

Wenn die Differentialgleichung (3.14) *steif* ist, das heißt wenn die Eigenwerte von A mit negativen Realteilen betragsmäßig stark unterschiedliche Größenordnungen haben, bedeutet dies, daß sehr kleine Schrittweiten notwendig sind, wenn das Stabilitätsgebiet beschränkt ist. Für die Behandlung steifer Probleme verlangt man daher:

$$\infty \in \text{int}(\mathcal{S}). \tag{3.17}$$

Man muß dazu implizite lineare Mehrschrittverfahren verwenden, die noch einer speziellen Eigenschaft genügen, wie der folgende Satz zeigt:

Satz 3.5.13 *[34] Explizite lineare Mehrschrittverfahren haben beschränkte Stabilitätsgebiete. Ein implizites lineares Mehrschrittverfahren, welches irreduzibel ist, erfüllt (3.17) genau dann, wenn alle Nullstellen ξ des charakteristischen Polynoms σ der Bedingung*

$$|\xi| < 1$$

genügen.

Zum Beispiel kann man nun σ so wählen, daß die Nullstellen alle gleich null sind. Dies führt, wenn man die Normierung $\beta_k = 1$ vornimmt, zu der Wahl

$$\sigma(\xi) = \xi^k$$

bzw. zu einem Verfahren der Form

$$\sum_{j=0}^{k} \alpha_j \, x_{n+j} = h \, f_{n+k}. \tag{3.18}$$

Zur Bestimmung der Konsistenzordnung p erhält man nach Satz 3.5.3 $p+1$ Gleichungen in den $k+1$ Unbekannten $\alpha_0, \ldots, \alpha_k$. Die Ordnung ist maximal, wenn $p = k$, dann sind die Koeffizienten eindeutig bestimmt.

Das Verfahren, das man auf diese Weise erhält, heißt *BDF-Verfahren*. Wie die Koeffizienten praktisch — auch für variable Schrittweiten — berechnet werden können, wird im folgenden Abschnitt dargestellt.

Für die Stabilitätseigenschaften von BDF-Verfahren gelten folgende Resultate:

Satz 3.5.14 *[62] Das k-Schritt-BDF-Verfahren ist dann und nur dann stabil, wenn*

$$k \leq 6$$

gilt.

Die BDF-Verfahren für $k = 1, \ldots, 6$ sind alle $A(\alpha)$-stabil, siehe Tabelle 3.1 und Abbildung 3.2.

k	1	2	3	4	5	6
α	90°	90°	86.03°	73.35°	51.84°	17.84°

Tabelle 3.1: $A(\alpha)$-Stabilität von BDF-Verfahren

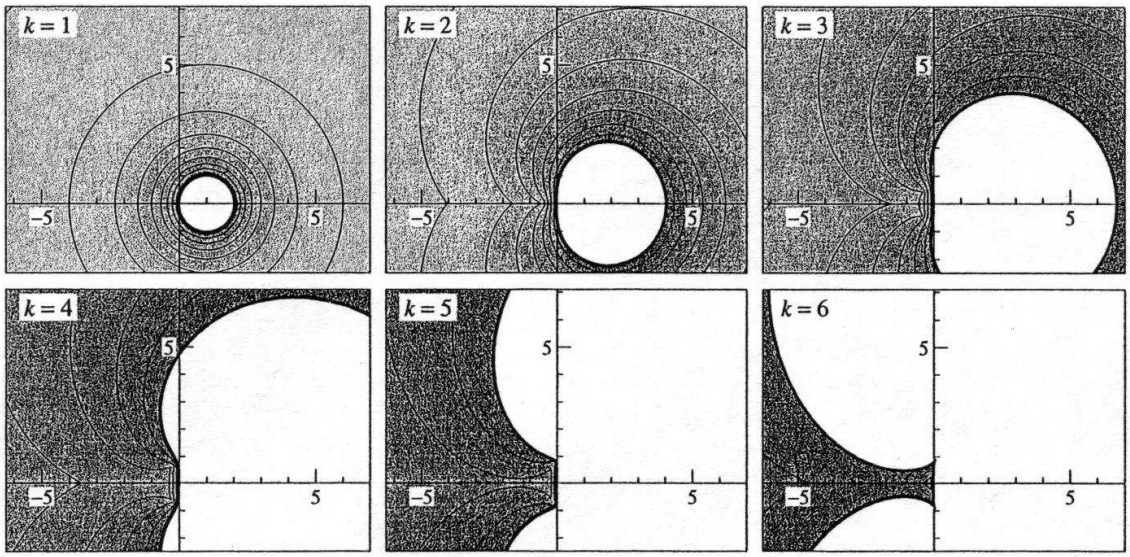


Abbildung 3.2: Stabilitätsgebiete (grau) der BDF-Verfahren [63]

3.6 BDF-Verfahren für Index-1-DAEs

Wir setzen BDF-Verfahren ein, um Anfangswertprobleme für steife Index-1-DAEs zu lösen.

3.6.1 Numerische Realisierung von BDF-Verfahren

Zur Herleitung der Koeffizienten des BDF-Verfahrens faßt man das lineare Mehrschrittverfahren als Konstruktion eines Interpolationspolynoms auf. Man interpoliert dabei den neu zu berechnenden Punkt x_{n+1} und die bereits berechneten Punkte $x_n, \ldots, x_{n-k}$ durch ein Polynom

$$p(t; x_{n+1}, x_n, \ldots, x_{n-k}) = \sum_{i=n-k}^{n+1} L_i(t) \cdot x_i \tag{3.19}$$

mit der Ableitung

$$\dot{p}(t; x_{n+1}, x_n, \ldots, x_{n-k}) = \sum_{i=n-k}^{n+1} \dot{L}_i(t) \cdot x_i. \tag{3.20}$$

L_i sind die Lagrangeschen Basispolynome:

$$L_i(t) = \prod_{\substack{j=n-k \\ j \neq i}}^{n+1} \frac{t - t_j}{t_i - t_j}, \quad i = n-k, \ldots, n+1.$$

$\dot{p}(t_{n+1}; y_{n+1}, y_n, \ldots, y_{n-k})$ soll nun $\dot{y}(t_{n+1})$ approximieren, vergleiche (3.18).

$$\dot{y}(t_{n+1}) \approx \dot{p}(t_{n+1}; y_{n+1}, y_n, \ldots, y_{n-k}) =: \frac{1}{h} \left(\alpha_0 \cdot y_{n+1} + \sum_{i=1}^{k+1} \alpha_i \cdot y_{n+1-i} \right) \tag{3.21}$$

Die Koeffizienten $\alpha_0, \ldots, \alpha_{k+1}$ des Mehrschrittverfahrens ergeben sich somit durch Koeffizientenvergleich von (3.20) und (3.21) als

$$\alpha_i = h \cdot \dot{L}_{n+1-i}(t_{n+1}), \quad i = 0, \ldots, k+1.$$

Abkürzend schreiben wir

$$\beta := \sum_{i=1}^{k+1} \alpha_i \cdot y_{n+1-i}.$$

Die Werte $x_n, \ldots, x_{n-k}$ seien rekursiv in den vorherigen Schritten des Mehrschrittverfahrens berechnet worden, x_{n+1} ist gesucht. Es wird nun dadurch bestimmt, daß das Polynom p im Punkt t_{n+1} die DAE erfüllen soll:

$$\begin{aligned} A(t_{n+1}, y_{n+1}, z_{n+1})\, \dot{p}(t_{n+1}; y_{n+1}, y_n, \ldots, y_{n-k}) &= f(t_{n+1}, y_{n+1}, z_{n+1}) \\ 0 &= g(t_{n+1}, y_{n+1}, z_{n+1}) \end{aligned}$$

bzw.

$$\begin{aligned} A_{n+1} \left(\alpha_0 \cdot y_{n+1} + \sum_{i=1}^{k+1} \alpha_i \cdot y_{n+1-i} \right) - h \cdot f_{n+1} &= 0 \\ g_{n+1} &= 0. \end{aligned}$$

Dies ist ein nichtlineares Gleichungssystem in den Unbekannten $x := (y_{n+1}, z_{n+1})$:

$$\mathcal{F}(x) = 0 \tag{3.22}$$

mit

$$\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}x} = \left(\begin{array}{c|c} \alpha_0 A_{n+1} + \frac{\partial A_{n+1}}{\partial y}(\alpha_0 y_{n+1} + \beta) - h\frac{\partial f_{n+1}}{\partial y} & \frac{\partial A_{n+1}}{\partial z}(\alpha_0 y_{n+1} + \beta) - h\frac{\partial f_{n+1}}{\partial z} \\ \hline \frac{\partial g_{n+1}}{\partial y} & \frac{\partial g_{n+1}}{\partial z} \end{array} \right). \tag{3.23}$$

$\frac{\mathrm{d}\mathcal{F}}{\mathrm{d}x}$ ist regulär wegen der Index-1-Bedingung und wenn f eine Lipschitzbedingung erfüllt und die Schrittweite h hinreichend klein gewählt wird, vergleiche Satz 3.5.1.

Das Gleichungssystem (3.22) kann man zum Beispiel mit einem Newton-ähnlichen Verfahren lösen [28]:

1. Den Startwert x^0 erhält man durch Extrapolation von $x_n, \ldots, x_{n-k-1}$ *(Prädiktor).*

2. Die Iteration

   $$\mathcal{M}_l \Delta x^l = -\mathcal{F}(x^l), \quad x^{l+1} := x^l + \Delta x^l, \quad l = 0, 1, \ldots$$

   wird durchgeführt, bis der Abbruchfehler kleinergleich als der Diskretisierungsfehler ist. In der Regel sind dafür nur 2 bis 3 Iterationen nötig *(Korrektor).*

Bemerkung 3.6.1

- *Der Prädiktor ist zwar ein expliziter Schritt. Er ist aber gerade das implizite Verfahrenspolynom des vorherigen BDF-Schrittes und als solches auch zur Lösung steifer Probleme geeignet.*

- *Die Iterationsmatrix $\mathcal{M}_l$ ist*

  $$\mathcal{M}_l = \frac{d\mathcal{F}}{dx}(x^l)$$

  für das volle Newton-Verfahren. Eine Neuberechnung und Neuzerlegung wird jeweils notwendig durch einen neuen Auswertungspunkt (y, z), eine neue Schrittweite h und neue Koeffizienten α. Da sich diese Größen aber nur langsam ändern, kann man auch mit einem näherungsweisen Newton-Verfahren arbeiten, bei dem

  $$\mathcal{M}_l \approx \frac{d\mathcal{F}}{dx}(x^l)$$

  gewählt wird. Dabei hält man $\mathcal{M}_l^{-1}$ über mehrere Schritte des BDF-Verfahrens fest. Eine Monitorstrategie mittels Fehlerschätzungen für die Newton-Iterierten kontrolliert, wann $\mathcal{M}_l$ neu berechnet und zerlegt wird. Für Einzelheiten siehe [28].

3.6.2 Schrittweiten- und Ordnungssteuerung

Auf adaptive Weise kann die Schrittweite und die Ordnung des BDF-Verfahrens in jedem Integrationsschritt automatisch an das Problem angepaßt werden, um den Diskretisierungsfehler zu kontrollieren.

Die folgenden Eigenschaften des BDF-Verfahrens machen dies möglich:

- Das BDF-Verfahren ist nicht nur auf äquidistanten, sondern auch auf variablen Gittern formulierbar. Man erhält dann statt konstanter gitterabhängige Koeffizienten.
- Aus der Newtonschen Darstellung des Interpolationspolynoms (3.19)

$$p(t; x_{n+1}, x_n, \ldots, x_{n-k}) = \sum_{i=0}^{k+1} x[t_{n-k}, \ldots, t_{n-k+i}] \cdot \prod_{j=0}^{i-1} (t - t_{n-k+j}) \tag{3.24}$$

  mit den dividierten Differenzen

$$\begin{aligned} x[t_i] &:= x_i, \quad i = n-k, \ldots, n+1 \\ x[t_i, \ldots, t_{i+j}] &:= \frac{x[t_{i+1}, \ldots, t_{i+j}] - x[t_i, \ldots, t_{i+j-1}]}{t_{i+j} - t_i} \end{aligned}$$

  erkennt man, daß man durch Hinzufügen weiterer Summanden die Ordnung des Verfahrens (bis 6, vergleiche Satz 3.5.14) erhöhen kann. Eine höhere Konsistenzordnung ist somit beim BDF-Verfahren nicht mit höheren Kosten (Funktionsauswertungen, Lösung nichtlinearer Gleichungssysteme) verbunden.

In DAESOL [17] ist eine simultane Ordnungs- und Schrittweitensteuerung implementiert, die folgendermaßen vorgeht [28]:

Ausgehend von einer Schätzung des lokalen Diskretisierungsfehlers mit Hilfe dividierter Differenzen

$$E_n := \rho\,(t_{n+1} - t_n)^2 (t_{n+1} - t_{n-1}) \cdots (t_{n+1} - t_{n+1-k})\, \|y[t_{n+1}, \ldots, t_{n-k}]\|$$

(mit $\rho < 1$ unabhängig von der Schrittweite) berechnet man für die Ordnungen $k-1$, k und $k+1$ Schrittweiten h^{eq}, mit denen jeweils für den äquidistanten Spezialfall eine vom Benutzer vorgegebene lokale Integrationsgenauigkeit erreicht werden könnte:

$$h^{eq}_{k'} = \sqrt[k'+1]{\frac{TOL'}{\rho\, k'!\, \|y[t_{n+1}, \ldots, t_{n-k'}]\|}}, \quad k' = k-1, k, k+1, \quad k' \le 6.$$

Durch Vergleich dieser Werte wird eine neue und bessere Ordnung ermittelt. Die Zielgenauigkeit TOL' wird hierbei kleiner gewählt als die gewünschte Toleranz TOL , etwa $TOL' = \frac{TOL}{2}$. Im nächsten Schritt wird mit der nicht-äquidistanten Schätzformel

$$E_{n+1} = \rho\,(t_{n+2} - t_{n+1})^2 (t_{n+2} - t_n) \cdots (t_{n+2} - t_{n+2-k})\, \|y[t_{n+2}, \ldots, t_{n+1-k}]\|$$

verglichen. Erfüllt die Schrittweite die Genauigkeitsbedingung

$$E_{n+1} \leq TOL,$$

wird sie akzeptiert, andernfalls wird sie entsprechend reduziert.

Das BDF-Verfahren kann mit $k = 1$ gestartet werden, dem Einschrittverfahren der Familie. Ab dem Zeitpunkt t_k kann man ein k-Schrittverfahren verwenden. Dies geschieht automatisch durch die Ordnungs- und Schrittweitensteuerung.

Als Alternative zum Selbststart des Mehrschrittverfahrens kann man auch Einschrittverfahren höherer Ordnung als Starter verwenden. Dies hat den Vorteil, daß die Integration gleich mit hoher Ordnung beginnen kann. Von Schwerin hat diese Vorgehensweise für Adams-Verfahren entwickelt [114], Bauer für BDF-Verfahren [14].

Bemerkung 3.6.2 *Die Auswertung des Interpolationspolynoms (3.20) bzw. (3.24) erlaubt eine fehlerkontrollierte und asympotisch korrekte kontinuierliche Darstellung der numerischen Lösung des DAE-Anfangswertproblems. Diese Eigenschaft machen wir uns in unserer Software zunutze, um Ausgaben auf beliebigen, benutzerdefinierten Gittern bereitzustellen. Diese können dann zum Beispiel zur Visualisierung der Lösungstrajektorien verwendet werden. Einzelheiten dazu finden sich in Abschnitt 9.2.4 in Kapitel 9.*

Kapitel 4

Parameterschätzung

Oft enthalten Modelle Größen, deren Werte man nicht oder nur sehr ungenau kennt. Zum Beispiel können dies spezifische Konstanten chemischer Reaktionen wie Geschwindigkeitskonstanten (Frequenzfaktoren, Aktivierungeenergien), Gleichgewichtskonstanten oder Reaktionsenthalpien oder Eigenschaften von Stoffen oder Geräten sein. Diese Größen bezeichnen wir als *Parameter*. Um ein Modell für eine realistische Simulation oder Optimierung eines Prozesses verwenden zu können, ist eine genaue Kenntnis der Parameter erforderlich. Zu ihrer Bestimmung werden Experimente durchgeführt und mit den Methoden der *Parameterschätzung* ausgewertet.

Wir diskutieren Parameterschätzprobleme in dieser Arbeit aus zwei Gründen. Zum einen ist ihre Problemformulierung und die statistische Analyse ihrer Lösungen wichtig zur Formulierung von Versuchsplanungsproblemem im nächsten Kapitel 5. Die optimale Versuchsplanung zielt auf die Planung von Experimenten, deren Meßdaten eine möglichst genaue Schätzung der Parameter ermöglichen, das heißt mit einem möglichst kleinen Konfidenzgebiet.

Außerdem schlagen wir in Abschnitt 8.1 in Kapitel 8 eine sequentielle Vorgehensweise vor, bei der Versuchsplanung, Experimente und Parameterschätzung abwechselnd durchgeführt werden. Dadurch will man jeweils optimalen Zugewinn an Informationen über den zu untersuchenden Prozeß gewinnen. Da wir dabei numerische Methoden zur Lösung von nichtlinearen Parameterschätzproblemen einsetzen, geben wir hier einen Abriß darüber.

4.1 Beschränkte Parameterschätzprobleme

Wir betrachten wieder Prozesse, die durch DAE-Modelle beschrieben werden, betonen nun aber, daß die Modellgleichungen von Parametern und Steuerungen abhängen können:

$$\begin{aligned} A(t, y(t), z(t), p, q)\ \dot{y}(t) &= f(t, y(t), z(t), p, q, u(t)) \\ 0 &= g(t, y(t), z(t), p, q, u(t)) \end{aligned}$$

Dabei bezeichne wieder $t \in [t_0, t_{end}] \subseteq \mathbb{R}$ die unabhängige Variable („Zeit“) und der Vektor $x(t) = (y(t), z(t)) \in \mathbb{R}^{n_y+n_z}$ die Zustandsgrößen. $p \in \mathbb{R}^{n_p}$ steht für den Parametervektor sowie $q \in \mathbb{R}^{n_q}$ und $u(t) \in \mathbb{R}^{n_u}$ für Steuergrößen und Steuerfunktionen. Oft lassen wir auch das Argument (t) bei x, y, z oder u weg.

Typischerweise sind Anfangswertprobleme gegeben. Die Anfangsbedingungen

$$y(t_0) = y_0(p, q)$$

können auch von Parametern oder Steuergrößen abhängen.

Außerdem müssen oft Innere-Punkt-Bedingungen der Form

$$0 = \hat{d}(x(t_1), \ldots, x(t_{\hat{K}}), p, q) \tag{4.1}$$

erfüllt werden. Wir fassen die Anfangsbedingungen als spezielle Innere-Punkt-Bedingungen auf und nehmen im folgenden an, daß sie in den Gleichungen (4.1) enthalten sind.

Seien nun Experimente durchgeführt worden, die Meßwerte η_i geliefert haben.

Wir machen folgende Annahmen über das Meßmodell: Die Meßfehler ϵ_i seien unabhängig und additiv

$$\eta_i = h_i(t_i, x^*(t_i), p^*, q) + \epsilon_i, \quad i = 1, \ldots, M \tag{4.2}$$

und normalverteilt mit Mittelwert 0 und bekannten Standardabweichungen σ_i

$$\epsilon_i \sim \mathcal{N}(0, \sigma_i^2), \quad i = 1, \ldots, M.$$

Dabei sind p^* die *wahren Werte* der Parameter und x^* die zugehörigen Lösungen des DAE-Systems. Man kann die Differenz zwischen Meßwert und Modellantwort auch für andere Werte p und x auswerten und erhält dann die *Residuen*

$$\eta_i - h_i(t_i, x(t_i), p, q), \quad i = 1, \ldots, M.$$

Das Parameterschätzproblem besteht nun in der Berechnung von gewichteten Least-Squares auf den Residuen [11, 40, 13, 107, 53]. Die Gewichte sind die Standardabweichungen der Messungen.

$$\min_{p,x} \quad \frac{1}{2} \sum_{i=1}^{M} \left(\frac{\eta_i - h_i(t_i, x(t_i), p, q)}{\sigma_i} \right)^2 \tag{4.3}$$

s. t.

$$A(t, y, z, p, q)\, \dot{y} = f(t, y, z, p, q, u) \tag{4.4}$$

$$0 = g(t, y, z, p, q, u) \tag{4.5}$$

$$0 = \hat{d}(x(t_1), \ldots, x(t_{\hat{K}}), p, q) \tag{4.6}$$

Als Nebenbedingungen treten die DAE-Modellgleichungen und die Innere-Punkt-Bedingungen auf. Optimierungsvariablen sind die Parameter p und die Zustandsvariablen x.

Steuergrößen q und Steuerfunktionen u sind fest, sie beschreiben die (ja bereits erfolgte) Durchführung der Experimente.

Es gibt verschiedene Möglichkeiten, die Lösung der DAE durch endlichdimensionale Variablen $s \in I\!R^{n_s}$ zu parametrisieren. Auf Einzelheiten hierzu wird in Abschnitt 4.4 eingegangen. Man erhält jeweils eine Darstellung der Lösung in Abhängigkeit von s:

$$x(t) = \psi(t, p, q, u, s). \tag{4.7}$$

Eventuell ergeben sich durch die Parametrisierung zusätzliche Nebenbedingungen

$$\tilde{d}(x(t_1), \ldots, x(t_{\tilde{K}}), p, q, s) = 0. \tag{4.8}$$

Für (4.7) schreiben wir nun auch

$$x(t, p, q, u, s) := \psi(t, p, q, u, s).$$

Setzt man dies in (4.3)-(4.6) ein, erhält man ein endlichdimensionales beschränktes Parameterschätzproblem:

$$\min_{p,s} \quad \frac{1}{2} \sum_{i=1}^{M} \left(\frac{\eta_i - h_i(t_i, x(t_i, p, q, u, s), p, q)}{\sigma_i} \right)^2 \tag{4.9}$$

s. t.

$$0 = d(x(t_1, p, q, u, s), \ldots, x(t_K, p, q, u, s), p, q, s) \tag{4.10}$$

Die Nebenbedingungen d ergeben sich dabei aus $\hat{d}$ und den Gleichungen $\tilde{d}$ aus der Parametrisierung der DAE-Lösung.

Wir fassen nun die im Parameterschätzproblem auftretenden Größen als Vektoren und Matrizen zusammen:

$$\begin{aligned} \eta &:= (\eta_1, \ldots, \eta_M)^T, \\ h &:= (h_i(t_i, x(t_i, p, q, u, s), p, q),\ i = 1, \ldots, M)^T, \\ \epsilon &:= (\epsilon_1, \ldots, \epsilon_M)^T, \\ \Sigma &:= \mathrm{diag}(\sigma_i,\ i = 1, \ldots, M). \end{aligned}$$

Dann können wir die Least-Squares-Terme und Nebenbedingungen kompakt formulieren:

$$F_1 := \left(\frac{\eta_i - h_i(t_i, x(t_i, p, q, u, s), p, q)}{\sigma_i} \right)_{i=1,\ldots,M} = \Sigma^{-1} (\eta - h) \tag{4.11}$$

$$F_2 := d \tag{4.12}$$

und

$$F := \begin{pmatrix} F_1 \\ F_2 \end{pmatrix}.$$

Zur Berechnung der Lösung des Parameterschätzproblems und zur statistischen Analyse der Lösung benötigt man die Jacobimatrizen

$$J_1 := \frac{\mathrm{d}F_1}{\mathrm{d}(p,s)} = -\Sigma^{-1}\frac{\mathrm{d}h}{\mathrm{d}(p,s)} \tag{4.13}$$

$$J_2 := \frac{\mathrm{d}F_2}{\mathrm{d}(p,s)} = \frac{\mathrm{d}d}{\mathrm{d}(p,s)} \tag{4.14}$$

und

$$J := \begin{pmatrix} J_1 \\ J_2 \end{pmatrix}. \tag{4.15}$$

Die numerische Berechnung dieser Größen wird in Kapitel 6 behandelt.

Zur bequemen Notation fassen wir die Variablen in einem Variablenvektor zusammen:

$$v := (p,s).$$

Das beschränkte Parameterschätzproblem (4.9)-(4.10) lautet in dieser Notation nun kurz

$$\begin{aligned} \min_v \quad & \frac{1}{2}\|F_1(v)\|_2^2 \\ & F_2(v) = 0. \end{aligned}$$

4.2 Gauß-Newton-Typ-Verfahren

Zur Lösung von Parameterschätzproblemen, wie sie im letzten Abschnitt eingeführt wurden, haben sich Verfahren vom Gauß-Newton-Typ bewährt.

Im Zusammenhang mit Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen wurden diese erstmals von Bock [26] und Schlöder [104] untersucht, siehe auch[106]. Im Rahmen dieser Arbeiten wurden die Programme PARFIT [26] und FIXFIT [104] entwickelt. Die Behandlung von Mehrfachexperimenten, siehe auch Kapitel 7, ist mit Varianten dieser Software möglich, siehe [104], [70] und[113]. Von Gallitzendörfer wurde eine Parallelisierung des Verfahrens implementiert [52, 29]. Die Anwendung auf durch Linienmethode diskretisierte Partiellle Differentialgleichungen wird in [36] behandelt.

4.2.1 Voraussetzungen

In diesem Abschnitt bezeichnen wir die Dimensionen mit

$$v \in I\!R^n, \quad F_1(v) \in I\!R^{n_1}, \quad F_2(v) \in I\!R^{n_2}.$$

Wir nehmen an, daß die folgenden beiden Bedingungen an allen Punkten v gelten, an denen im folgenden F oder J ausgewertet werden muß:

- (CQ) („Constraint Qualification“)

$$\text{Rang} J_2(v) = n_2, \tag{4.16}$$

- (PD) („Positive Definiteness“)

$$\text{Rang} J(v) = n. \tag{4.17}$$

Lemma 4.2.1 *Wenn (CQ) und (PD) in v erfüllt sind, dann gelten auch die folgenden Aussagen:*

1. *$J_1(v)^T J_1(v)$ ist positiv definit auf dem Kern von $J_2(v)$.*
2. *Die Matrix $\begin{pmatrix} J_1(v)^T J_1(v) & J_2(v)^T \\ J_2(v) & 0 \end{pmatrix}$ ist regulär.*

Beweis.

1. Sei $p \neq 0$, $p \in \text{Kern} J_2(v)$, also $J_2(v)p = 0$.
   Wegen (PD) ist Kern $\begin{pmatrix} J_1(v) \\ J_2(v) \end{pmatrix} = \{0\}$.
   Daher ist $J_1(v)p \neq 0$ und somit $p^T J_1(v)^T J_1(v)p > 0$.
2. Wir nehmen an, es gebe $\begin{pmatrix} p \\ q \end{pmatrix} \neq 0$ mit $J_1(v)^T J_1(v)p + J_2(v)^T q = 0$ und $J_2(v)p = 0$.
   Dann ist auch $p^T J_1(v)^T J_1(v)p + p^T J_2(v)^T q = p^T J_1(v)^T J_1(v)p = 0$.
   Wegen Aussage 1 dieses Lemmas muß also $p = 0$ sein.
   Dann gilt aber $J_2(v)^T q = 0$, und weil $J_2(v)$ Vollrang hat (CQ), folgt daraus $q = 0$.
   ↯

■

4.2.2 Algorithmus

Der folgende Algorithmus, das Verallgemeinerte Gauß-Newton-Verfahren, löst eine Folge von linearisierten beschränkten Ausgleichsproblemen.

Algorithmus 4.2.2 (Verallgemeinertes Gauß-Newton-Verfahren)

1. *Starte mit einer Schätzung v_0.*
2. *Für $k = 0, 1, 2, \ldots$:*

(a) Löse das linearisierte Problem

$$\min_{\Delta v_k} \quad \frac{1}{2}\|J_1(v_k)\Delta v_k + F_1(v_k)\|_2^2 \tag{4.18}$$

$$J_2(v_k)\Delta v_k + F_2(v_k) = 0 \tag{4.19}$$

(b) Berechne die neue Iterierte

$$v_{k+1} := v_k + \alpha \cdot \Delta v_k$$

mit $\alpha \in (0;1]$ aus einer Globalisierungsstrategie.

(c) Ist eine Abbruchbedingung erfüllt, dann beende den Algorithmus mit der Lösung $\hat{v} := v_{k+1}$.

Als Abbruchbedingung kann zum Beispiel gewählt werden, daß das Inkrement Δv_k klein ist:

$$\|\Delta v_k\|_2 \leq \epsilon.$$

4.2.3 Darstellung der Lösung der linearisierten Probleme

Wir betrachten das linearisierte Problem (4.18)-(4.19) aus der k-ten Iteration von Algorithmus 4.2.2. Zur Abkürzung schreiben wir in diesem Unterabschnitt:

$$F_1 := F_1(v_k),\ F_2 := F_2(v_k),\ J_1 := J_1(v_k),\ J_2 := J_2(v_k),\ \Delta v := \Delta v_k.$$

Es sei daran erinnert, daß wir die Bedingungen (CQ) und (PD) ((4.16) und (4.17)) voraussetzen.

Die hinreichenden und notwendigen Bedingungen für die Optimalität (Karush-Kuhn-Tucker-Bedingungen) [57] bestehen in der Erfüllung der Nebenbedingungen und darin, daß der Gradient der Lagrangefunktion

$$\begin{aligned} \mathcal{L}(\Delta v, \lambda) &= \frac{1}{2}\Delta v^T J_1^T J_1 \Delta v + F_1^T J_1 \Delta v + \frac{1}{2} F_1^T F_1 - \lambda^T (J_2 \Delta v + F_2) \\ \nabla_{\Delta v}\mathcal{L}(\Delta v, \lambda) &= J_1^T J_1 \Delta v + J_1^T F_1 - J_2^T \lambda \end{aligned}$$

gleich null ist.

Also: Es existiert $\lambda \in \mathbb{R}^{n_2}$, so daß

$$\begin{aligned} J_1^T J_1 \Delta v + J_1^T F_1 - J_2^T \lambda &= 0 \\ J_2 \Delta v + F_2 &= 0 \end{aligned}$$

beziehungsweise

$$\begin{pmatrix} J_1^T J_1 & J_2^T \\ J_1 & 0 \end{pmatrix} \begin{pmatrix} \Delta v \\ -\lambda \end{pmatrix} = - \begin{pmatrix} J_1^T F_1 \\ F_2 \end{pmatrix}.$$

Die Lösung Δv besitzt somit die folgende Darstellung:

$$\Delta v = -\begin{pmatrix} I & 0 \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_1^T & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} = -J^+ \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} \tag{4.20}$$

mit der *Verallgemeinerten Inversen*

$$J^+ = \begin{pmatrix} I & 0 \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_1^T & 0 \\ 0 & I \end{pmatrix}.$$

Die *adjungierten Variablen* λ lassen sich gemäß

$$\lambda = \begin{pmatrix} 0 & I \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_1^T & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} F_1 \\ F_2 \end{pmatrix}$$

darstellen.

Wegen (CQ) und (PD) gilt Existenz und Eindeutigkeit von J^+ (Lemma 4.2.1) und somit auch der Lösung Δv des linearisierten Problems sowie der adjungierten Variablen λ. J^+ erfüllt die Beziehung $J^+ J J^+ = J^+$, diese ist eines der vier Moore-Penrose-Axiome. Im allgemeinen ist J^+ aber nicht die Moore-Penrose-Pseudoinverse [89], nur im unbeschränkten Fall.

Zur Konvergenz des Gauß-Newton-Verfahrens und zur praktischen Implementierung des Algorithmus siehe die Abschnitte 4.4 und 4.5.

4.3 Statistische Analyse der Lösung

Da die Meßdaten Zufallsvariablen sind, ist auch die Lösung des Gauß-Newton-Verfahrens eine Zufallsvariable.

Wir betrachten das linearisierte Problem im Lösungspunkt $\hat{v}$ des Gauß-Newton-Verfahrens. Es gilt: $\Delta v = -J^+ F$ ist multinormalverteilt mit Mittelwert

$$\begin{aligned} E(\Delta v) &= E(-J^+ \begin{pmatrix} F_1 \\ F_2 \end{pmatrix}) \\ &= -J^+ E \begin{pmatrix} F_1 \\ F_2 \end{pmatrix} \\ &= -J^+ \begin{pmatrix} E(\Sigma^{-1}(\eta - h)) \\ E(d) \end{pmatrix} \\ &= -J^+ \begin{pmatrix} 0 \\ 0 \end{pmatrix} \\ &= 0 \end{aligned}$$

und Kovarianzmatrix

$$\begin{aligned} C &= E(\Delta v \Delta v^T) \\ &= E(J^+ \begin{pmatrix} F_1F_1^T & F_1F_2^T \\ F_2F_1^T & F_2F_2^T \end{pmatrix} J^{+T}) \\ &= J^+ \begin{pmatrix} E(F_1F_1^T) & 0 \\ 0 & 0 \end{pmatrix} J^{+T} \\ &= J^+ \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} J^{+T} \\ &= \begin{pmatrix} I & 0 \end{pmatrix} \begin{pmatrix} J_1^TJ_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_1^TJ_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} J_1^TJ_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-T} \begin{pmatrix} I \\ 0 \end{pmatrix}. \end{aligned} \quad (4.21)$$

Dabei haben wir benutzt, daß

$$\begin{aligned} E(F_1F_1^T) &= \Sigma^{-1}\, E(\eta\eta^T)\, \Sigma^{-1} \\ &= \Sigma^{-1}\, \Sigma^2\, \Sigma^{-1} \\ &= I \end{aligned}$$

und $E(F_1F_2^T) = 0$, $E(F_2F_1^T) = 0$, $E(F_2F_2^T) = 0$, weil F_2 keine Zufallsvariable ist.

Das $(100 \cdot \alpha)\%$-*Konfidenzgebiet* ist das Gebiet, in dem die wahren — aber unbekannten — Parameter mit einer Wahrscheinlichkeit von $\alpha \in [0, 1]$ liegen.

Das Parameterschätzproblem (4.9)-(4.10) hat n Variablen und n_2 Gleichungsnebenbedingungen. Sei

$$\gamma^2(\alpha) := \chi^2_{n-n_2}(1-\alpha) \quad (4.22)$$

das Quantil der χ^2-Verteilung zum Wert α mit $n - n_2$ Freiheitsgraden.

Da wir die wahren Parameter nicht kennen, betrachten wir eine Näherung des $(100 \cdot \alpha)\%$-Konfidenzgebiets im Lösungspunkt $\hat{v}$ des Gauß-Newton-Verfahrens:

$$G(\alpha, \hat{v}) := \{v \in I\!R^n :\ F_2(v) = 0,\ \|F_1(v)\|_2^2 - \|F_1(\hat{v})\|_2^2 \le \gamma^2(\alpha)\}.$$

Wegen der Nichtlinearität der Ausgleichsfunktion und der Beschränkungen ist dieses meist nur sehr schwer zu bestimmen. Daher beschränken wir uns auf die linearisierte Näherung

$$\begin{aligned} G_L(\alpha, \hat{v}) \ :=\ & \{v \in I\!R^n :\ F_2(\hat{v}) + J_2(\hat{v})(v - \hat{v}) = 0, \\ & \|F_1(\hat{v}) + J_1(\hat{v})(v - \hat{v})\|_2^2 - \|F_1(\hat{v})\|_2^2 \le \gamma^2(\alpha)\}. \end{aligned} \quad (4.23)$$

Das folgende Lemma sagt, daß sich das linearisierte Konfidenzgebiet auch in der Form

$$\bar{G}_L(\alpha, \hat{v}) := \{v \in I\!R^n :\ v - \hat{v} = -J^+(\hat{v}) \begin{pmatrix} \delta w \\ 0 \end{pmatrix},\ \delta w \in I\!R^{n_1},\ \|\delta w\|_2 \le \gamma(\alpha)\}$$

darstellen läßt:

Lemma 4.3.1 *[26] Es gilt:*

$$G_L(\alpha, \hat{v}) = \bar{G}_L(\alpha, \hat{v}).$$

Beweis. Sei $\hat{v}$ die Lösung von

$$\begin{aligned} \min_v \quad & \frac{1}{2}\|F_1(v)\|_2^2 \\ & F_2(v) = 0. \end{aligned}$$

Aufgrund der KKT-Bedingungen gilt im Lösungspunkt $\hat{v}$ (wir lassen das Argument $\hat{v}$ bei $F_{1,2}$ und $J_{1,2}$) weg): Es existiert λ mit

$$\begin{aligned} 0 &= \frac{\partial}{\partial v}\left(\frac{1}{2}F_1^T F_1 - \lambda^T F_2\right)\delta v = \frac{1}{2}\left(\delta v^T J_1^T F_1 + F_1^T J_1 \delta v\right) - \lambda^T J_2 \delta v \\ &= \frac{1}{2}\left((F_1 + J_1\delta v)^T(F_1 + J_1\delta v) - F_1^T F_1 - (J_1\delta v)^T(J_1\delta v)\right) - \lambda^T J_2\delta v. \end{aligned}$$

Betrachte δv mit $J_2\delta v = 0$. Dann folgt:

$$\|J_1\delta v\|_2^2 = \|F_1 + J_1\delta v\|_2^2 - \|F_1\|_2^2. \tag{4.24}$$

Sei $v \in G_L(\alpha, \hat{v})$. Definiere $\delta v := v - \hat{v}$. Dann ist $J_2\delta v = -F_2 = 0$. Sei $\delta w := -J_1\delta v$. Es gilt:

$$\|\delta w\|_2^2 = \|J_1\delta v\|_2^2 = \|F_1 + J_1\delta v\|_2^2 - \|F_1\|_2^2 \le \gamma(\alpha)^2.$$

Das lineare Ausgleichsproblem

$$\begin{aligned} \min_{\Delta v} \quad & \frac{1}{2}\|\delta w + J_1\Delta v\|_2^2 && (4.25) \\ & J_2\Delta v = 0 && (4.26) \end{aligned}$$

hat die Lösung $\Delta v = \delta v$, wie man leicht direkt ablesen kann, nach (4.20) hat diese aber auch die Darstellung

$$\delta v = -J^+(\hat{v})\begin{pmatrix} \delta w \\ 0 \end{pmatrix}.$$

Daraus folgt $v \in \bar{G}_L(\alpha, \hat{v})$.

Sei umgekehrt $v \in \bar{G}_L(\alpha, \hat{v})$. Also gilt $\|\delta w\|_2 \le \gamma(\alpha)$. $\delta v := v - \hat{v} = -J^+(\hat{v})\begin{pmatrix} \delta w \\ 0 \end{pmatrix}$ löst das lineare Ausgleichsproblem (4.26)-(4.26). Aufgrund der KKT-Bedingungen ist daher $J_2\delta v = 0$, und es existiert λ, so daß

$$\begin{aligned} 0 &= \frac{\partial}{\partial \Delta v}\frac{1}{2}\left(\|\delta w\|_2^2 + 2\cdot\delta w^T J_1\Delta v + \Delta v^T J_1^T J_1\Delta v\right) - \lambda^T J_2\Delta v\Big|_{\Delta v = \delta v}\delta v \\ &= \delta w^T J_1\delta v + \delta v^T J_1^T J_1\delta v - \lambda^T J_2\delta v = \delta w^T J_1\delta v + \|J_1\delta v\|_2^2. \end{aligned}$$

Daraus und aus (4.24) folgt

$$\begin{aligned} 0 &\leq \|\delta w + J_1\delta v\|_2^2 = \|\delta w\|_2^2 + 2\cdot\delta w^T J_1\delta v + \|J_1\delta v\|_2^2 = \|\delta w\|_2^2 - \|J_1\delta v\|_2^2 \\ &\leq \gamma(\alpha)^2 + \|F_1\|_2^2 - \|F_1 + J_1\delta v\|_2^2. \end{aligned}$$

Also ist

$$\|F_1 + J_1\delta v\|_2^2 - \|F_1\|_2^2 \leq \gamma(\alpha)^2$$

und damit $v \in G_L(\alpha, \hat{v})$. ■

Mit diesem Resultat haben wir die Möglichkeit, Abschätzungen für die Konfidenzintervalle der einzelnen Parameter anzugeben:

Satz 4.3.2 *Sei*

$$\theta_i := \gamma(\alpha)\cdot\sqrt{C_{ii}(\hat{v})}, \quad i = 1, \ldots, n,$$

wobei $\gamma(\alpha)$ der in (4.22) definierte Wahrscheinlichkeitsfaktor und $\sqrt{C_{ii}(\hat{v})}$ die Wurzel des i-ten Hauptdiagonalelements der Kovarianzmatrix (4.21) ist. Dann gilt: Das linearisierte Konfidenzgebiet $G_L(\alpha, \hat{v})$ wird durch einen Quader mit den Seitenlängen $2\cdot\theta_i$ eingeschlossen:

$$G_L(\alpha, \hat{v}) \subseteq [\hat{v}_1 - \theta_1, \hat{v}_1 + \theta_1] \times \ldots \times [\hat{v}_n - \theta_n, \hat{v}_n + \theta_n].$$

Beweis. Sei $e_i \in \mathbb{R}^n$ der i-te Einheitsvektor. Nach Lemma 4.3.1 gilt für $v \in G_L(\alpha, \hat{v})$:

$$\begin{aligned} |v_i - \hat{v}_i| &\leq \| \left(J^+(\hat{v}) \begin{pmatrix} I \\ 0 \end{pmatrix} \right)^T e_i\|_2 \cdot \|\delta w\|_2 \\ &= \left(e_i^T J^+(\hat{v}) \begin{pmatrix} I \\ 0 \end{pmatrix} \begin{pmatrix} I & 0 \end{pmatrix} J^+(\hat{v})^T e_i \right)^{\frac{1}{2}} \cdot \|\delta w\|_2 \\ &\leq \sqrt{C_{ii}(\hat{v})}\cdot\gamma(\alpha), \quad i = 1, \ldots, n. \end{aligned}$$

■

Bevor wir im nächsten Kapitel an diese statistische Analyse der Lösung von Parameterschätzproblemen angeknüpfen und darauf die Optimierung von Versuchsplänen begründen, wollen wir uns im Rest dieses Kapitels mit einigen numerischen Aspekten der Parameterschätzung befassen.

4.4 Parametrisierung der Dynamik

In diesem Abschnitt geben wir zwei Techniken zur Parametrisierung der Dynamik an, vergleiche (4.7)-(4.8).

4.4.1 Einfachschießverfahren

Die einfachste Möglichkeit ist das Einfachschießverfahren (Single-Shooting, Black-Box-Ansatz).

Man löst dabei das *relaxierte* DAE-System

$$\begin{aligned} A(t,y,z,p,q)\,\dot{y} &= f(t,y,z,p,q,u) \\ 0 &= g(t,y,z,p,q,u) - \beta(t)\cdot g(t_0,y(t_0),z(t_0),p,q,u) \end{aligned}$$

mit den Anfangsbedingungen

$$y(t_0) = y_0(p,q), \quad z(t_0) = s$$

mit einem Anfangswertproblemlöser auf dem gesamten Integrationsintervall $t \in [t_0, t_{end}]$. Durch die Relaxierung der algebraischen Gleichungen, siehe Abschnitt 3.3.1, sind diese für beliebige s automatisch konsistent.

Es ergibt sich dadurch eine Darstellung der Lösung an jedem Zeitpunkt $t \in [t_0, t_{end}]$:

$$x(t) = \psi(t,p,q,u,s).$$

Die algebraischen Variablen müssen zusätzlich die Konsistenzbedingungen

$$g(t_0,y(t_0),z(t_0),p,q,u) = 0$$

erfüllen. Mit $s := z(t_0)$ ergibt dies eine zusätzliche Nebenbedingung, die wir zu den Gleichungen des Optimierungsproblems hinzunehmen:

$$\tilde{d}(x(t_1),\ldots,x(t_{\tilde{K}}),p,q,s) = 0.$$

Problematisch kann diese Methode im Zusammenhang mit Gauß-Newton-Verfahren (oder anderen Newton-Typ-Verfahren) werden, wenn man mit sehr schlechten Anfangsschätzern für die Parameter (oder die entsprechenden Optimierungsvariablen) startet. Dann kann man sehr weit entfernt sein vom Konvergenzbereich des Newton-Typ-Verfahrens (siehe unten). Es kann auch vorkommen, daß die DAE dann gar keine globale Lösung besitzt, daß also keine Starttrajektorie existiert.

4.4.2 Mehrzielmethode

Um die Probleme vermeiden zu können, die beim Einfachschießverfahren auftreten, wurde die Mehrzielmethode (Multiple Shooting) entwickelt. Multiple Shooting wurde zunächst nur zur Lösung von Randwertproblemen verwendet, siehe zum Beispiel [109]. Bock [26] hat diese Methode erstmals auch zur Parametrisierung der Dynamik bei der Lösung von Parameterschätzproblemen eingesetzt.

Die Idee besteht in der Zerlegung des Integrationsintervalls in Teilintervalle

$$t_0 < t_1 < \ldots < t_{N_{ms}} < t_{N_{ms}+1} = t_{end}$$

und der Integration des relaxierten DAE-Systems

$$\begin{aligned} A(t,y,z,p,q)\,\dot{y} &= f(t,y,z,p,q,u) \\ 0 &= g(t,y,z,p,q,u) - \beta(t)\cdot g(t_i,y(t_i),z(t_i),p,q,u) \end{aligned}$$

nur jeweils auf diesen Teilintervallen $t \in [t_i, t_{i+1})$, $i = 0,\ldots,N_{ms}$ mit den Anfangsbedingungen

$$y(t_i) = s_{y,i}, \quad z(t_i) = s_{z,i}, \quad i = 0,\ldots,N_{ms}$$

und $s_{y,0} = y_0(p,q)$, siehe Abbildung 4.1. Die Größen $s_i = (s_{y,i}, s_{z,i})$, $i = 0,\ldots,N_{ms}$ sind zusätzliche Variablen des Problems.

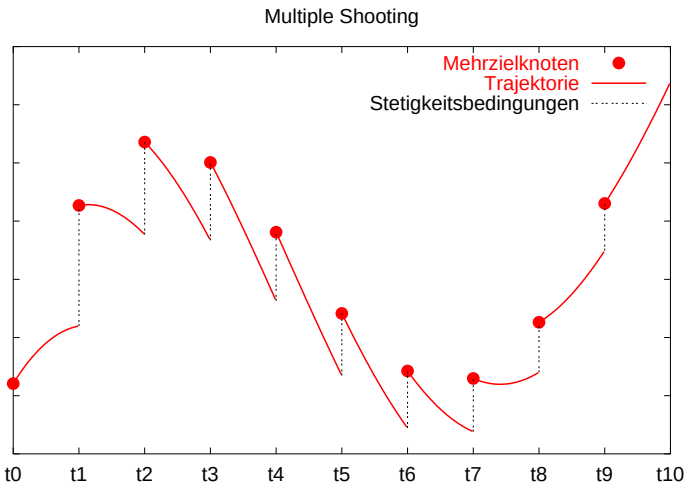


Abbildung 4.1: Veranschaulichung der Mehrzielmethode

Auswertung der Lösungen ergibt eine abschnittsweise Darstellung

$$x(t) = \psi_i(t,p,q,u,s_i), \quad t \in [t_i, t_{i+1}), \quad i = 0,\ldots,N_{ms}.$$

Zusätzlich muß man aber noch verlangen, daß die Lösung stetig ist, das heißt

$$\lim_{t\to t_i} \psi_{y,i-1}(t,p,q,u,s_{i-1}) - s_{y,i} = 0, \quad i = 1,\ldots,N_{ms},$$

und daß die Konsistenzbedingungen

$$g(t_i, s_{y,i}, s_{z,i}, p, q, u) = 0, \quad i = 0,\ldots,N_{ms}$$

erfüllt sind.

Stetigkeitsbedingungen und Konsistenzbedingungen sind also zusätzliche Gleichungen

$$\tilde{d}(x(t_1), \ldots, x(t_{\tilde{K}}), p, q, s) = 0,$$

die zum Parameterschätzproblem hinzugenommen werden müssen.

Durch die Mehrzielmethode wird gewissermaßen das Vorwärtsproblem, also die Lösung der DAE, als implizite Nebenbedingung des Parameterschätzproblems formuliert.

Der wesentliche Vorteil der Mehrzielmethode gegenüber dem Einfachschießverfahren besteht darin, daß man die Variablen des Parameterschätzproblems oft durch Kenntnis des Prozeßverhaltens, zum Beispiel durch Meßdaten, initialisieren kann. Dadurch wird der Einfluß schlechter Parameterstartdaten abgeschwächt, man kann „nahe“ der Lösung starten und viel eher Konvergenz des Gauß-Newton-Verfahrens erwarten als beim Einfachschießverfahren. Die Existenz einer Starttrajektorie ist auf dem ganzen Intervall gesichert, die Integration ist numerisch stabil. Die Wahl der Mehrzielpunkte ist dabei meist unkritisch, eine Strategie zur Bestimmung des Mehrzielgitters ist zum Beispiel in [26] ausgeführt.

Auf den ersten Blick erhält man bei der Mehrzielmethode ein höherdimensionales Optimierungsproblem. Dessen spezielle Struktur kann jedoch bei der numerischen Behandlung effizient ausgenutzt werden.

Die Jacobimatrix der Least-Squares-Terme und der Nebenbedingungen hat bei der Mehrzielmethode die folgende Struktur:

$$J = \left(\frac{J_1}{J_2}\right) = \left(\begin{array}{ccccccc}
D_1^1 & D_1^0 & & \cdots & & D_1^{N_{ms}} & D_1^p \\
\hline
D_2^1 & D_2^0 & & \cdots & & D_2^{N_{ms}} & D_2^p \\
G_0 & (-I,0) & & & & & G_0^p \\
 & G_1 & (-I,0) & & & & G_1^p \\
 & & \ddots & \ddots & & & \vdots \\
 & & & G_{N_{ms}-1} & (-I,0) & & G_{N_{ms}-1}^p \\
H_0 & & & & & & H_1^p \\
 & H_1 & & & & & H_2^p \\
 & & \ddots & & & & \vdots \\
 & & & & & H_{N_{ms}} & H_{N_{ms}}^p
\end{array}\right) \tag{4.27}$$

mit

$$\begin{aligned}
D_1^i &:= \frac{\partial F_1}{\partial s_i}, \quad i = 0, \ldots, N_{ms}, \quad D_1^p := \frac{\partial F_1}{\partial p}, \\
D_2^i &:= \frac{\partial \hat{d}}{\partial s_i}, \quad i = 0, \ldots, N_{ms}, \quad D_2^p := \frac{\partial \hat{d}}{\partial p}, \\
G_i &:= \frac{\partial \psi_{y,i}}{\partial s_i}(t_{i+1}, p, q, u, s_i), \quad i = 0, \ldots, N_{ms} - 1, \\
G_i^p &:= \frac{\partial \psi_{y,i}}{\partial p}(t_{i+1}, p, q, u, s_i)
\end{aligned}$$

$$
\begin{aligned}
H_i &:= \frac{\partial g}{\partial (s_{y,i}, s_{z,i})}(t_i, s_{y,i}, s_{z,i}, p, q, u), \quad i = 0, \ldots, N_{ms}, \\
H_i^p &:= \frac{\partial g}{\partial p}(t_i, s_{y,i}, s_{z,i}, p, q, u).
\end{aligned}
$$

Die Matrizen $(-I, 0)$ haben in (4.27) die Dimensionen $I \in I\!R^{n_y \times n_y}$, $0 \in I\!R^{n_y \times n_z}$.

Der Aufwand zur Integration entspricht ungefähr dem beim Einfachschießverfahren. Die Struktur der Matrix (4.27) kann ausgenutzt werden, um die quadratischen Programme im Gauß-Newton-Verfahren effizient zu lösen. Dadurch ist der Aufwand für die Lineare Algebra im wesentlichen genauso groß wie beim Einfachschießverfahren. Zu Einzelheiten sei an dieser Stelle auf die Arbeiten [26, 104] verwiesen. Darüber hinaus eignet sich die strukturausnutzende Lineare Algebra ebenso wie die entkoppelte Auswertung des DAE-Systems auf den einzelnen Teilintervallen sehr gut zur Parallelisierung, siehe [52, 29].

Eine Alternative zur Mehrzielmethode, die ähnliche Vorteile aufweist, sind Kollokationsverfahren, siehe zum Beispiel [105].

4.5 Konvergenzresultate für Newton-Typ-Verfahren

In Newton-Typ-Verfahren wird das Inkrement gemäß $\Delta v_k = -M(v_k)F(v_k)$ berechnet. Man kann dabei unterscheiden:

- für nichtlineare Gleichungssysteme $F(v) = 0$ das *Newton-Verfahren* mit der Inversen als Iterationsmatrix:
$$M(v_k) = J(v_k)^{-1},$$
- für unbeschränkte Ausgleichsprobleme $\min_v \frac{1}{2}\|F(v)\|_2^2$ *unbeschränkte Gauß-Newton-Verfahren* mit der Moore-Penrose-Pseudoinversen:
$$M(v_k) = J(v_k)^\dagger = \left(J(v_k)^T J(v_k)\right)^{-1} J(v_k)^T$$
- und für beschränkte Ausgleichsprobleme $\min_v \frac{1}{2}\|F_1(v)\|_2^2$ unter $F_2(v) = 0$ *beschränkte Gauß-Newton-Verfahren* mit der verallgemeinerten Inversen:
$$M(v_k) = J(v_k)^+ = \begin{pmatrix} I & 0 \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_1^T & 0 \\ 0 & I \end{pmatrix}.$$

In *Quasi-Newton-Verfahren* werden statt dieser Inversen Näherungen verwendet.

Bemerkung 4.5.1 *Man kann die Gauß-Newton-Verfahren auch als Näherungen für die Newton-Verfahren für die notwendigen Bedingungen der entsprechenden Optimierungsprobleme interpretieren.*

Wir diskutieren hier den unbeschränkten Fall, im beschränkten Fall sind analoge Betrachtungen durchzuführen.

$$\min_v \frac{1}{2}\|F(v)\|_2^2$$

Die notwendige Bedingung zur Minimierung der Zielfunktion

$$\phi(v) = \frac{1}{2}\sum_i F_i(v)^2$$

lautet

$$\frac{d\phi}{dv}(v) = \frac{1}{2}\sum_i 2 \cdot \frac{dF_i}{dv}(v) \cdot F_i(v)^2 = J(v)^T F(v) = 0.$$

Mit der zweiten Ableitung der Zielfunktion

$$\frac{d^2\phi}{dv^2}(v) = \sum_i \left(\frac{dF_i}{dv}(v) \cdot \frac{dF_i}{dv}(v) + \frac{\partial^2 F_i}{\partial v^2}(v) \cdot F_i(v) \right) = J(v)^T J(v) + H(v)F(v)$$

ist beim Newton-Verfahren in jeder Iteration ein lineares Problem

$$J(v_k)^T F(v_k) + \left(J(v_k)^T J(v_k) + H(v_k)F(v_k) \right) \Delta v_k = 0$$

zu lösen. Das Inkrement ist also

$$\Delta v_k = -\left(J(v_k)^T J(v_k) + H(v_k)F(v_k) \right)^{-1} J(v_k)^T F(v_k).$$

Bei der Parameterschätzung ist der Term $H(v)F(v)$ abhängig von den Meßfehlern, also eine Zufallsvariable. Daher ist es besser, nur die Näherung

$$\Delta v_k = -\left(J(v_k)^T J(v_k) \right)^{-1} J(v_k)^T F(v_k)$$

zu verwenden, was dem Gauß-Newton-Verfahren entspricht.

Der folgende Satz charakterisiert das lokale Konvergenzverhalten von Newton-Typ-Verfahren.

Satz 4.5.2 (Lokaler Kontraktionssatz) *[26]*
Betrachte $F : \mathbb{R}^m \supseteq D \to \mathbb{R}^n$, $F \in C^1(D, \mathbb{R}^n)$, $J := \frac{dF}{dv}$.

Es gelte für alle $v, w \in D$, $\theta \in [0,1]$ mit $w - v = -M(v)F(v)$:

1. *Es existiert ein $\omega < \infty$, so daß*

$$\|M(w)\left(J(v + \theta(w - v)) - J(v)\right)(w - v)\| \leq \omega \cdot \theta \cdot \|w - v\|^2 .$$

2. *Es existiert ein* $\kappa(v) \leq \kappa < 1$, *so daß*

$$\|M(w)R(v)\| \leq \kappa(v)\,\|w - v\|$$

mit dem Residuum $R(v) := F(v) - J(v)M(v)F(v)$.

Sei $v_0 \in D$ *gegeben mit*

$$\delta_0 := \kappa + \frac{\omega}{2}\,\|\Delta v_0\| < 1,$$

$$\delta_k := \kappa + \frac{\omega}{2}\,\|\Delta v_k\|\,, \quad \Delta v_k := -M(v_k)F(v_k)$$

und

$$D_0 := \left\{ v : \|v - v_0\| \leq \frac{\|\Delta v_0\|}{1 - \delta_0} \right\} \subseteq D.$$

Dann gilt:

1. *Die Iteration* $v_{k+1} = v_k + \Delta v_k$ *ist wohldefiniert und bleibt in* D_0.
2. *Es existiert ein* $v_* \in D_0$, *so daß* $v_k \to v_*$ *konvergiert für* $k \to \infty$.
3. *Es gilt die Abschätzung*

$$\|v_{k+j} - v_*\| \leq \frac{\delta_k^j}{1 - \delta_k}\,\|\Delta v_k\|$$

4. *und*

$$\|\Delta v_{k+1}\| \leq \delta_k\,\|\Delta v_k\| = \kappa\,\|\Delta v_k\| + \frac{\omega}{2}\,\|\Delta v_k\|^2\,.$$

Beweis. Siehe [26]. ∎

Für Newton-Verfahren folgt aus dem Lokalen Kontraktionssatz die quadratische Konvergenz:

Korollar 4.5.3 *Bei Newton-Verfahren ist* $M = J^{-1}$. *Dann gilt aber*

$$\left\|J^{-1}(w)R(v)\right\| = \left\|J^{-1}(w)\left(I - J(v)J^{-1}(v)\right)F(v)\right\| = 0.$$

Also ist $\kappa = 0$. *Das bedeutet quadratische Konvergenz:*

$$\|v_{k+1} - v_k\| \leq C\|v_k - v_{k-1}\|^2.$$

Bei Gauß-Newton-Verfahren wird das Konvergenzverhalten und die Konvergenzgeschwindigkeit durch die Größen κ und ω bestimmt.

Bemerkung 4.5.4 *Sei $F \in C^2(D, \mathbb{R}^n)$.*

1. *Gelten die Regularitätbedingungen (CQ) und (PD) in einem Punkt v, so gilt für w aus einer Umgebung von v: $M(w)$ existiert und ist stetig differenzierbar, und $M(w)$ ist beschränkt: $\|M(w)\| \leq \beta$. Außerdem gilt*

$$\|(J(v+\theta(w-v)) - J(v))(w-v)\| = \left\| \frac{dJ}{dv}(\phi)\theta(w-v)(w-v) \right\| \leq \gamma \cdot \theta \cdot \|w-v\|^2$$

   in einer kompakten Teilmenge von D. γ ist ein Maß für die Nichtlinearität *des Problems.*

2. *In einer Umgebung von v_* gilt*

$$\|M(w)(J(v+\theta(w-v)) - J(v))(w-v)\| \leq \beta \cdot \gamma \cdot \theta \cdot \|w-v\|^2 .$$

   Also ist $\omega \leq \beta \cdot \gamma < \infty$. ω kann sehr groß werden, falls $\|M\|$ sehr groß ist (β), das heißt J_2 oder $\begin{pmatrix} J_1 \\ J_2 \end{pmatrix}$ ist nahezu singulär, oder falls die 1. Ableitung von J sehr groß ist (γ), das heißt das Problem ist sehr nichtlinear.

3. *Wenn M die Verallgemeinerte Inverse ist, gilt $M = MJM$ und*

$$M(v)R(v) = M(v)(F(v) - J(v)M(v)F(v)) = (M(v) - M(v)J(v)M(v))F(v) = 0.$$

   Daraus folgt

$$\|M(w)R(v)\| = \|(M(w) - M(v))R(v)\| \leq L \cdot \rho \cdot \|w-v\|,$$

   da M stetig differenzierbar, daher $\|M(w) - M(v)\| \leq L \cdot \|w-v\|$, und $\|R(v)\| \leq \rho$. Also ist $\kappa \leq L \cdot \rho$. $\kappa < 1$ kann man nur erwarten, wenn die erste Ableitung von M klein ist (L) und wenn das Residuum klein ist (ρ). κ ist ein Maß für die Kompatibilität *zwischen Modell und Daten.*

Kapitel 5

Nichtlineare Optimale Versuchsplanung

5.1 Formulierung von Versuchsplanungsproblemen

Im letzten Kapitel haben wir gesehen, daß die Güte einer Parameterschätzung durch die Kovarianzmatrix im Lösungspunkt des Gauß-Newton-Verfahrens ausgedrückt werden kann. Diese hängt von den Größen q und u ab, die die Durchführung der Experimente beschreiben. Wir unterscheiden zwischen *Steuergrößen* q, die zeitlich konstant einen bestimmten Wert haben, und den zeitlich variablen *Steuerfunktionen* u. Durch geschickte Wahl dieser Steuerungen innerhalb vorgegebener Beschränkungen und eine geeignete Plazierung der Meßpunkte kann man Experimente entwerfen, deren Meßdaten eine möglichst signifikante Schätzung erlauben. Dies kann entweder intuitiv durch Ausprobieren erfolgen oder durch eine mathematische Optimierung berechnet werden. In diesem Abschnitt werden wir Nichtlineare Optimale Versuchsplanungsprobleme formulieren. Die folgenden Abschnitte werden sich dann mit Methoden zu deren numerischer Lösung befassen.

5.1.1 Steuergrößen und Steuerfunktionen

Durch die Steuerungen werden die Versuchsbedingungen zur Durchführung der Experimente modelliert.

Wir unterscheiden zwischen

- Steuergrößen $q \in I\!R^{n_q}$
- und zeitlich variablen Steuerfunktionen $u : [t_0, t_{end}] \to I\!R^{n_u}$.

Für die Steuerungen können Grenzen gegeben sein

$$\begin{aligned} q_i &\in [lo_{q_i}, up_{q_i}], \quad i = 1, \ldots, n_q \\ u_i(t) &\in [lo_{u_i}, up_{u_i}], \quad i = 1, \ldots, n_u \end{aligned}$$

oder (nichtlineare) Steuerbeschränkungen

$$c(q) \in [lo_c, up_c]$$

mit $c : I\!R^{n_q} \to I\!R^{n_c}$.

$lo_\cdot$ und $up_\cdot$ stehen hier für die jeweils gegebenen unteren und oberen Schranken in den geeigneten Dimensionen.

5.1.2 Planung von Messungen

Neben der Wahl der experimentellen Bedingungen hängt die Güte einer Parameterschätzung auch in großem Maße von der Planung der Messungen ab. Aus Kosten- oder Kapazitätsgründen können meistens nicht beliebig viele Daten gemessen werden. Daher ist es notwendig, die Plazierung der Probenahmen für die verschiedenen verfügbaren Meßverfahren optimal zu bestimmen.

Zur Optimierung eines Meßentwurfs geben wir zunächst *mögliche Meßpunkte* vor. Wir betrachten M aufsteigend geordnete Zeitpunkte $t_0 \le t_1 \le \ldots \le t_M \le t_{end}$, an denen Messungen des jeweiligen Meßverfahrens mit der Modellantwort $h_i(t_i, x(t_i), p, q)$, $i = 1, \ldots, M$ durchgeführt werden können. Dabei können Meßzeitpunkte auch gleich sein. Die Meßfunktionen an verschiedenen Meßzeitpunkten können natürlich auch die gleichen Meßverfahren beschreiben.

Dann geben wir jeder möglichen Messung ein *Gewicht*

$$w_i \in \{0, 1\}, \quad i = 1, \ldots, M. \tag{5.1}$$

Messungen, die tatsächlich durchgeführt werden sollen, erhalten durch die Optimierung das Gewicht $w_i = 1$, Messungen, die nicht ausgewählt werden, haben das Gewicht $w_i = 0$.

Durch Beschränkungen an die Gewichte können nun Bedingungen wie zum Beispiel die maximale Anzahl von Messungen pro Zeitpunkt, pro Meßverfahren, pro Experiment usw. modelliert werden:

$$lo_{J_j} \le \sum_{i \in J_j} w_i \le up_{J_j}, \quad j = 1, \ldots, N_J$$

für geeignete Indexmengen $J_j \subseteq \{1, \ldots, M\},\ j = 1, \ldots, N_J$.

Die Kosten der Messungen können unter Annahme eines linearen Kostenmodells mit Kostenkoeffizienten c_i^w für Messung i zum Beispiel durch

$$\sum_{i=1}^{M} c_i^w \cdot w_i \tag{5.2}$$

beschrieben werden.

Oft werden wir Relaxierungen der Ganzzahligkeitsbedingungen (5.1) betrachten, zum Beispiel $w_i \in [0, 1]$. Diese können wir dadurch interpretieren, daß Messung i mit Gewicht w_i

mit w_i-fachem „Aufwand“ durchgeführt wird, das heißt mit w_i^{-1}-facher Varianz und w_i-fachen Kosten. Auf diese Weise können auch Gewichte mit $w_i > 1$ verwendet werden.

Durch die Einführung der Gewichte modifizieren wir die Verteilung der Meßfehler, siehe Gleichung (4.2), somit durch

$$\epsilon_i \sim \mathcal{N}\left(0, \frac{\sigma_i^2}{w_i}\right), \quad i = 1, \ldots, M.$$

Dabei gilt für nicht durchgeführte Messungen: geht das Gewicht $w_i \to 0$, geht die Varianz gegen unendlich.

Die Diagonalmatrix der Gewichte bezeichnen wir mit

$$W := \mathrm{diag}(w_i,\ i = 1, \ldots, M).$$

5.1.3 Das der Versuchsplanung zugrundeliegende Parameterschätzproblem

Würde man die geplanten Experimente durchführen und anhand der experimentellen Daten die Parameter schätzen, ergäbe sich nach Parametrisierung der Dynamik entsprechend (4.9)-(4.9) das folgende nichtlineare Ausgleichsproblem:

$$\begin{aligned} \min_{p,s} \quad & \frac{1}{2}\sum_{i=1}^{M} w_i \cdot \frac{(\eta_i - h_i(t_i, x(t_i, p, q, u, s), p, q))^2}{\sigma_i^2} \\ \text{s. t.} & \\ 0 \;&=\; d(x(t_1, p, q, u, s), \ldots, x(t_K, p, q, u, s), p, q, s). \end{aligned}$$

Man beachte, daß wir die Varianzen σ_i^2 durch die gewichteten Varianzen $\frac{\sigma_i^2}{w_i}$ ersetzt haben.

Wegen dieser Modifikation ändert sich die Definition von F_1 und J_1 gegenüber (4.11) und (4.13) in

$$\begin{aligned} F_1 \;&:=\; \left(\sqrt{w_i} \cdot \frac{\eta_i - h_i(t_i, x(t_i, p, q, u, s), p, q)}{\sigma_i}\right)_{i=1,\ldots,M} = \sqrt{W}\,\Sigma^{-1}\,(\eta - h) \\ J_1 \;&:=\; \frac{\mathrm{d}F_1}{\mathrm{d}(p,s)} = -\sqrt{W}\,\Sigma^{-1}\,\frac{\mathrm{d}h}{\mathrm{d}(p,s)}. \end{aligned}$$

Die Kovarianzmatrix im Lösungspunkt hat die Darstellung

$$C = \begin{pmatrix} I & 0 \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_1^T J_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-T} \begin{pmatrix} I \\ 0 \end{pmatrix}.$$

Sie ist — bei festen $v = (p, s)$ — eine Funktion der *Versuchsplanungsgrößen* q, u und w.

5.1.4 Gütekriterien

Wie in der klassischen Versuchsplanung benutzen wir Informationsfunktionen ϕ auf der Kovarianzmatrix C zur Beurteilung der Güte einer Parameterschätzung, siehe [93]. Die üblichsten Funktionen sind dabei die klassischen Gütekriterien:

- Das Spur-Kriterium (A-Kriterium):

$$\phi_A(C) = \frac{1}{n} \cdot \text{Spur}C.$$

- Das Determinanten-Kriterium (D-Kriterium):

$$\phi_D(C) = \left(\det(K^T C K)\right)^{\frac{1}{n}}.$$

  Da die Kovarianzmatrix im allgemeinen nicht regulär ist, betrachtet man die Projektion auf einen n_K-dimensionalen Unterraum des Parameterraums mittels einer vollrangigen Projektionsmatrix $K \in I\!R^{n \times n_K}$.

- Das Größter-Eigenwert-Kriterium (E-Kriterium):

$$\phi_E(C) = \max\{\lambda : \lambda \text{ Eigenwert von } C\}.$$

In Satz 4.3.2 haben wir eine Abschätzung für die Konfidenzintervalle der einzelnen Parameter angegeben. Diese können wir ebenfalls als Gütekriterium verwenden:

- Das Konfidenzintervall-Kriterium (M-Kriterium):

$$\phi_M(C) = \max\{\sqrt{C_{ii}},\ i = 1, \ldots, n\}.$$

Diese Gütekriterien lassen sich auch geometrisch anhand des Konfidenzellipsoids $G_L(\alpha, v)$ (4.23) interpretieren. Das A-Kriterium ist proportional zur durchschnittlichen Halbachsenlänge des Konfidenzellipsoids, das D-Kriterium zu seinem Volumen, das E-Kriterium zur größten Halbachsenlänge und das M-Kriterium zur größten Seitenlänge einer Box um das Konfidenzellipsoid, siehe Abbildung 5.1.

Bemerkung 5.1.1 *Die Kovarianzmatrix und damit auch die Gütekriterien hängen stark von der Skalierung der Parameter im zugrundeliegenden Prozeßmodell ab. Die Kovarianzmatrix bezieht sich auf die absoluten Parameterwerte. Deshalb werden in allen Gütekriterien die Varianzen der Parameter mit absolut großen Werten viel stärker berücksichtigt als die der Parameter mit absolut kleinen Werten. Will man alle Parameter gleich gewichten, so muß man sie auf den gleichen Wert skalieren. Man kann die Skalierung aber auch nutzen, um bei der Versuchsplanung gezielt bestimmte Parameter stärker ins Gewicht fallen zu lassen.*

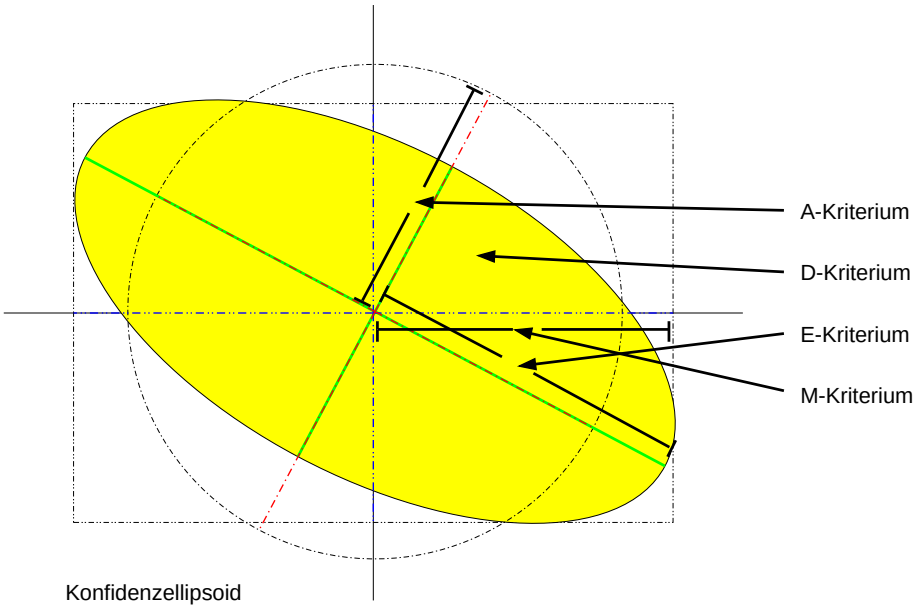


Abbildung 5.1: Geometrische Veranschaulichung der Gütekriterien

Die Skalierung der Parameter und s-Variablen $v = (p, s)$ zu $\tilde{v} = (\tilde{p}, \tilde{s})$ kann man schreiben als $v = S\tilde{v}$ mit einer regulären Matrix S; meistens wird S dabei als Diagonalmatrix gewählt. Dann ist

$$F_{1,2}(v) = F_{1,2}(S\tilde{v}) =: \tilde{F}_{1,2}(\tilde{v})$$

und

$$\tilde{J}_{1,2}(\tilde{v}) := \frac{\mathrm{d}\tilde{F}_{1,2}(\tilde{v})}{\mathrm{d}\tilde{v}} = \frac{\mathrm{d}F_{1,2}(S\tilde{v})}{\mathrm{d}\tilde{v}} = \frac{\mathrm{d}F_{1,2}(v)}{\mathrm{d}v} \cdot \frac{\mathrm{d}v}{\mathrm{d}\tilde{v}} = \frac{\mathrm{d}F_{1,2}(v)}{\mathrm{d}v} \cdot S = J_{1,2} \cdot S.$$

Also geht die Kovarianzmatrix für die skalierten Werte $\tilde{v}$ aus der Kovarianzmatrix für v durch Multipliaktion von links und rechts mit S hervor:

$$\tilde{C} = S^T \cdot C \cdot S.$$

5.1.5 Abhängigkeit der Standardabweichungen der Messungen von den Versuchsbedingungen

In der Parameterschätzung sind nach unseren Annahmen die Standardabweichungen σ_i durch die Eigenschaft der Meßdaten gegeben und fest. In der Versuchsplanung wollen wir aber auch berücksichtigen, daß Änderungen der experimentellen Bedingungen auch Änderungen der Meßgenauigkeiten bewirken können. Deshalb ersetzen wir die Konstanten σ_i durch Funktionen $\varsigma_i(x, q)$, $i = 1, \ldots, M$, in denen die Standardabweichungen also von Zuständen und Steuergrößen abhängen können. In der Parameterschätzung soll die Standardabweichung aber weiterhin fest bleiben, das heißt $\frac{\mathrm{d}\varsigma}{\mathrm{d}v} = 0$.

5.1.6 Kosten für die Prozeßsteuerung

Wir modellieren neben den Kosten für Messungen, siehe (5.2), auch Kosten für die Verwendung von Steuerungen. Eine Möglichkeit der Modellierung bei positiven Steuerungen ist ein lineares Kostenmodell mit Kosten c_i^q für eine Einheit von Steuergröße q_i bzw. Kosten $c_i^u(t)$ für eine Einheit von Steuerfunktion $u_i(t)dt$. Die Gesamtkosten des Versuchs sind dann

$$\sum_{i=1}^{M} c_i^w w_i + \sum_{i=1}^{n_q} c_i^q q_i + \sum_{i=1}^{n_u} \int_{t_0}^{t_{end}} c_i^u(t) u_i(t) dt. \tag{5.3}$$

5.1.7 Versuchsraum

Der Raum der Versuchsplanungsgrößen wird durch diverse Nebenbedingungen eingeschränkt. Wir unterscheiden

- zeitabhängige Zustandsbeschränkungen

$$lo_b \leq b(t, x, p, q, u) \leq up_b,$$

- Steuerbeschränkungen

$$lo_c \leq c(q) \leq up_c,$$

- Gewichtebedingungen

$$lo_{J_j} \leq \sum_{i \in J_j} w_i \leq up_{J_j},\ j = 1, \ldots, N_J,$$

- Kostenbeschränkungen

$$lo_{Kosten} \leq \sum_{i=1}^{M} c_i^w w_i + \sum_{i=1}^{n_q} c_i^q q_i + \sum_{i=1}^{n_u} \int_{t_0}^{t_{end}} c_i^u(t) u_i(t) dt \leq up_{Kosten},$$

- Grenzen an die Variablen

$$lo_{Bounds} \leq (q, u, w) \leq up_{Bounds}.$$

Alle diese Beschränkungen fassen wir nun formal zusammen zu Ungleichungsbedingungen

$$lo \leq \hat{\psi}(t, x, p, q, u, w) \leq up$$

und, wenn untere und obere Schranken gleich sind, Gleichungsbedingungen

$$0 = \hat{\chi}(t, x, p, q, u, w).$$

Daneben treten noch die 0/1-Bedingungen für die Gewichte auf:

$$w_i \in \{0, 1\},\ i = 1, \ldots, M.$$

5.1.8 Das Versuchsplanungs-Optimierungsproblem

Die Berechnung von Steuerungen, Steuerfunktionen und Meßgewichten zur Optimierung der Güte einer Parameterschätzung führt somit auf das folgende nichtlineare Optimierungsproblem:

$$
\left.
\begin{array}{l}
\min\limits_{q,u,w,x} \qquad \phi(C) \\
\\
\text{Dabei berechnet sich die Kovarianzmatrix} \\
C = \begin{pmatrix} I & 0 \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_1^T J_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-T} \begin{pmatrix} I \\ 0 \end{pmatrix} \\
\text{aus der Jacobimatrix } \begin{pmatrix} J_1^T & J_2^T \end{pmatrix}^T \text{ im Lösungspunkt} \\
\text{des beschränkten Parameterschätzproblems} \\
\qquad \begin{array}{rcl} \min\limits_{p,s} & \frac{1}{2} \sum\limits_{i=1}^{M} w_i \cdot \frac{(\eta_i - h_i(t_i, x(t_i,p,q,u,s),p,q))^2}{\varsigma_i(x(t_i,p,q,u,s),q)^2} \\ 0 & = & d(x(t_1,p,q,u,s),\ldots,x(t_K,p,q,u,s),p,q,s). \end{array} \\
\\
\text{Nebenbedingungen sind das DAE-System} \\
\begin{array}{rcl} A(t,y,z,p,q)\,\dot{y} & = & f(t,y,z,p,q,u) \\ 0 & = & g(t,y,z,p,q,u) \end{array} \\
\text{sowie} \\
\begin{array}{rcl} lo & \leq & \hat{\psi}(t,x,p,q,u,w) \leq up, \\ 0 & = & \hat{\chi}(t,x,p,q,u,w) \end{array} \\
\text{und} \\
\qquad w \in \{0,1\}^M.
\end{array}
\right\} \tag{5.4}
$$

Das zugrundeliegende DAE-Modell des Prozesses wird wieder als Nebenbedingung formuliert. Dadurch und durch die Optimierung der Steuerfunktionen ist der Variablenraum unendlichdimensional. Durch die 0/1-Bedingungen an die w-Variablen ist das Problem gemischt-ganzzahlig.

5.2 Transformationen des Problems zur numerischen Behandlung

Wir geben nun Techniken an, wie wir Problem (5.4) modifizieren können, um es mit Methoden der Nichtlinearen Optimierung behandeln zu können.

5.2.1 Relaxierung der 0-1-Bedingungen

Die Ganzzahligkeitsbedingungen

$$w_i \in \{0,1\}, \quad i = 1,\ldots,M$$

relaxieren wir, das heißt wir ersetzen sie durch kontinuierliche Formulierungen. Möglich sind dabei zum Beispiel die Varianten

$$w_i \in [0,1], \quad i = 1, \ldots, M \tag{5.5}$$

oder

$$w_i \in \mathbb{R}_0^+, \quad i = 1, \ldots, M. \tag{5.6}$$

Wichtig dabei ist, daß auch die Lösungen des relaxierten Problems noch eine sinnvolle Bedeutung haben. Für (5.5) und (5.6) haben wir das in Abschnitt 5.1.2 bereits diskutiert. In Abschnitt 5.4 werden wir Möglichkeiten angeben, wie aus den Lösungen der relaxierten Probleme ganzzahlige Designs erzeugt werden können.

Die relaxierten Ganzzahligkeitsbedingungen nehmen wir nun zu den Ungleichungsnebenbedingungen $lo \leq \hat{\psi}(t, x, p, q, u, w) \leq up$ hinzu.

5.2.2 M-Kriterium

Wenn als Zielfunktion das Konfidenzintervall-Kriterium auf der Kovarianzmatrix

$$\phi_M(C) = \max\{\sqrt{C_{ii}},\ i = 1, \ldots, n\}$$

gewählt wird, formulieren wir das min-max-Optimierungsproblem um, indem wir eine zusätzliche Variable $q_0 \in \mathbb{R}$ einführen und die Maximumsbildung durch Ungleichungsbedingungen ausdrücken:

$$\begin{array}{lcl} \min & & q_0 \\ \text{s. t.} & & \\ \quad q_0 & \geq & \sqrt{C_{ii}}, \quad i = 1, \ldots, n \\ \quad \ldots & & \end{array}$$

Diese zusätzlichen Ungleichungen nehmen wir ebenfalls zu den Ungleichungsnebenbedingungen $lo \leq \hat{\psi}(t, x, p, q, u, w) \leq up$ des Optimierungsproblems hinzu und q_0 zum Vektor q.

5.2.3 Parametrisierung der Steuerfunktionen

Zur Behandlung der Steuerfunktionen verwenden wir den *direkten Ansatz*, das heißt wir ersetzen eine Steuerfunktion u durch eine Funktion $\hat{u}$, die durch endlich viele Freiheitsgrade beschrieben werden kann.

Konkret betrachten wir hier ein Gitter $t_0 < t_1 < \ldots < t_{\hat{N}} = t_{end}$ und ersetzen u darauf durch stückweise Gitterfunktionen $\hat{u}_i$, $i = 1, \ldots, \hat{N}$. Die Variablen zur Beschreibung dieser Parametrisierung nennen wir $\hat{q} \in \mathbb{R}^{\hat{q}}$.

Als Gitterfunktionen können wir zum Beispiel Polynome verwenden:

- stückweise konstant: $\hat{u}_i(t) = \hat{q}_i$ für $t \in [t_{i-1}, t_i)$, $i = 1, \ldots, \hat{N}$
- stückweise linear: $\hat{u}_i(t) = \hat{q}_{2i} + \hat{q}_{2i+1} \cdot (t - t_{i-1})$ für $t \in [t_{i-1}, t_i)$, $i = 1, \ldots, \hat{N}$
- usw.

Die Wahl des Gitters und der Gitterfunktionen kann oft an der praktischen Realisierung orientiert werden. Für infinitesimal feine Gitter konvergieren die Lösungen der parametrisierten Probleme gegen die Lösung des kontinuierlichen Steuerproblems, siehe [50].

Nebenbedingungen an die Steuerfunktionen werden bei der Parametrisierung zu Nebenbedingungen an die Parametrisierungsvariablen $\hat{q}$, zum Beispiel wird

$$u(t) \in [lo_u, up_u]$$

für $\hat{u}_i(t) = \hat{q}_{2i} + \hat{q}_{2i+1} \cdot (t - t_{i-1})$ auf $t \in [t_{i-1}, t_i)$ zu

$$\hat{q}_{2i} \in [lo_{q_{2i}}, up_{q_{2i}}] \text{ und } \hat{q}_{2i} + \hat{q}_{2i+1} \cdot (t_i - t_{i-1}) \in [lo_{q_{2i+1}}, up_{q_{2i+1}}].$$

Soll die Steuerfunktion stetig sein, führen die Stetigkeitsbedingungen

$$\lim_{t \to t_i} \hat{u}_i(t) = \hat{u}_{i+1}(t_i)$$

ebenfalls zu Nebenbedingungen an die Parametrisierungsvariablen, zum Beispiel

$$\hat{q}_{2i} + \hat{q}_{2i+1} \cdot (t_i - t_{i-1}) = \hat{q}_{2i+2}.$$

Die Kosten der Steuerfunktion transformieren sich ebenfalls auf die Parametrisierungsvariablen, so wird

$$\int_{t_0}^{t_{end}} c^u(t) u(t) dt$$

zum Beispiel für stückweise lineare Gitterfunktionen zu

$$\begin{aligned}
\sum_{i=1}^{\hat{N}} \int_{t_{i-1}}^{t_i} c^u(t) \hat{u}_i(t) dt &= \sum_{i=1}^{\hat{N}} \int_{t_{i-1}}^{t_i} c^u(t) \left(\hat{q}_{2i} + \hat{q}_{2i+1} \cdot (t - t_{i-1}) \right) dt \\
&= \sum_{i=1}^{\hat{N}} \left(\int_{t_{i-1}}^{t_i} c^u(t) dt \cdot \hat{q}_{2i} + \int_{t_{i-1}}^{t_i} c^u(t) \cdot (t - t_{i-1}) dt \cdot \hat{q}_{2i+1} \right) \\
&= \sum_{i=1}^{\hat{N}} \left(c^{\hat{q}}_{2i} \hat{q}_{2i} + c^{\hat{q}}_{2i+1} \hat{q}_{2i+1} \right),
\end{aligned}$$

also zu Kosten, die ebenfalls linear in den Parametrisierungsvariablen sind mit den Koeffizienten

$$c^{\hat{q}}_{2i} = \int_{t_{i-1}}^{t_i} c^u(t) dt \quad \text{und} \quad c^{\hat{q}}_{2i+1} = \int_{t_{i-1}}^{t_i} c^u(t) \cdot (t - t_{i-1}) dt.$$

Wir nehmen die zusätzlichen Nebenbedingungen zum System $lo \leq \hat{\psi} \leq up$, $0 = \hat{\chi}$ und die Variablen $\hat{q}$ zum Vektor q hinzu. Diesen erweiterten Steuergrößenvektor bezeichnen wir von nun an mit q.

5.2.4 Behandlung der Dynamik und der Zustandsbeschränkungen

Zur Behandlung der Zustandsbeschränkungen diskretisieren wir diese auf einem Gitter. Wir betrachten $t_0 < t_1 < \ldots < t_{end}$ und verlangen, daß die zeitabhängigen Nebenbedingungen an diesen endlich vielen Zeitpunkten erfüllt werden. Auch die Wahl dieses Gitters kann den praktischen Gegebenheiten angepaßt werden.

Die DAE-Nebenbedingung im Versuchsplanungs-Optimierungsproblem parametrisieren wir wieder wie bei der Parameterschätzung, siehe Abschnitt 4.1, durch Einführung von s-Variablen und eventuell zusätzlichen Nebenbedingungen.

Die Variablen des Optimierungsproblems, die *Versuchsplanungsgrößen*, fassen wir im Vektor ξ zusammen:

$$\xi := (q, s, w)\,.$$

Aus den Nebenbedingungen

$$\begin{aligned} lo &\leq \hat{\psi}(t, x, p, q, u, w) \leq up \\ 0 &= \hat{\chi}(t, x, p, q, u, w) \end{aligned}$$

wird durch diese Parametrisierungen und Diskretisierungen das endlich-dimensionale System von Beschränkungen

$$\begin{aligned} lo &\leq \psi(\xi) \leq up \\ 0 &= \chi(\xi), \end{aligned}$$

die Zielfunktion schreiben wir nun allgemein als

$$\min_{\xi} \phi(\xi).$$

Wir haben das Versuchsplanungs-Optimierungsproblem (5.4) damit in das folgende endlich-dimensionale nichtlineare beschränkte Optimierungsproblem transformiert:

$$\begin{aligned} \min_{\xi} \quad & \phi(\xi) \\ lo &\leq \psi(\xi) \leq up \\ 0 &= \chi(\xi). \end{aligned}$$

Im nächsten Abschnitt stellen wir Methoden vor, mit deren Hilfe wir Probleme dieser Art lösen können.

5.3 Lösungsverfahren: SQP-Verfahren

Wir betrachten nichtlineare Optimierungsprobleme der allgemeinen Form

$$\min_{\xi \in \mathbb{R}^n} \quad \phi(\xi) \tag{5.7}$$

$$0 = \chi(\xi) \tag{5.8}$$

$$lo \leq \psi(\xi) \leq up. \tag{5.9}$$

Wir setzen voraus, daß die Abbildungen

$$\phi : I\!R^{n_\xi} \to I\!R, \quad \chi : I\!R^{n_\xi} \to I\!R^{n_\chi}, \quad \psi : I\!R^{n_\xi} \to I\!R^{n_\psi}$$

zweimal stetig differenzierbar sind.

Die *Lagrangefunktion* von Problem (5.7)-(5.9) ist definiert durch

$$\mathcal{L}(\xi, \lambda, \mu) = \phi(\xi) - \lambda^T \chi(\xi) - \mu^T \psi(\xi)$$

mit den *Lagrangemultiplikatoren* $\lambda \in I\!R^{n_\chi}$ und $\mu \in I\!R^{n_\psi}$.

5.3.1 Einführung

SQP-Verfahren (Sequentielle Quadratische Programmierung) haben sich bei der Lösung von beschränkten nichtlinearen Optimierungsproblemen als sehr leistungsfähig erwiesen.

In der Literatur werden SQP-Verfahren in den gängigen Lehrbüchern über nichtlineare Optimierung behandelt, siehe zum Beispiel [77, 47, 56, 108, 83]. Außerdem sind einige Implementierungen verfügbar. Wir verwenden die Software SNOPT [54, 55].

Die Idee bei SQP-Verfahren besteht in der iterativen Lösung des nichtlinearen Optimierungsproblems. In jeder Iteration wird ein beschränktes quadratisches Problem gelöst. Dessen Zielfunktion erhält man durch quadratische Entwicklung mit dem Gradienten der nichtlinearen Zielfunktion und der Hessematrix der Lagrangefunktion, die Nebenbedingungen durch Linearisierung der nichtlinearen Nebenbedingungen. Die Hessematrix der Lagrangefunktion wird dabei meistens nicht exakt berechnet, sondern als Näherung effizient bereitgestellt.

Unser Anliegen in diesem Abschnitt ist vor allem die Diskussion der bei SQP-Verfahren benötigten Ableitungsinformationen. Wie wir bereits gesehen haben, hängt bei Versuchsplanungsproblemen die Zielfunktion von ersten Ableitungen der Lösung des zugrundeliegenden DAE-Systems nach den Parametern p und den Variablen s ab. Wenn wir zur Optimierung dieser Probleme SQP-Verfahren mit Update-Techniken für die Hessematrizen einsetzen, müssen wir zur Berechnung der Gradienten also noch eine weitere Ableitungsordnung bereitstellen. Die Konvergenztheorie besagt, daß wir dann superlinear konvergente Verfahren erhalten können. Wie wir die benötigten Ableitungen konkret berechnen, wird in Kapitel 6 behandelt.

5.3.2 Das Verfahren

Der folgende Algorithmus beschreibt die Grundform von SQP-Verfahren.

Algorithmus 5.3.1 (SQP-Verfahren)

1. *Starte mit den Startwerten ξ^0, λ^0 und μ^0. Setze $k := 0$.*

2. *Berechne die Funktionswerte*

$$\phi^k := \phi(\xi^k), \quad \chi^k := \chi(\xi^k), \quad \psi^k := \psi(\xi^k),$$

die Gradienten

$$\nabla\phi^k := \nabla\phi(\xi^k), \quad \nabla\chi^k := \nabla\chi(\xi^k), \quad \nabla\psi^k := \nabla\psi(\xi^k)$$

und eine Näherung der Hessematrix der Lagrangefunktion

$$H^k \approx \nabla^2_\xi \mathcal{L}(\xi^k, \lambda^k, \mu^k).$$

3. *Bestimme die Suchrichtung $\Delta\xi^k$ als Lösung des Quadratischen Problems*

$$\begin{aligned} \min_{\Delta\xi} \quad & \frac{1}{2}\Delta\xi^T H^k \Delta\xi + {\nabla\phi^k}^T \Delta\xi \\ 0 \;&=\; \chi^k + {\nabla\chi^k}^T \Delta\xi \\ lo \;&\leq\; \psi^k + {\nabla\psi^k}^T \Delta\xi \leq up. \end{aligned}$$

Berechne die zugehörigen Lagrange-Multiplikatoren $\tilde{\lambda}^k$ und $\tilde{\mu}^k$.

4. *Bestimme die Schrittweite α^k. Iteriere:*

$$\begin{aligned} \xi^{k+1} &:= \xi^k + \alpha^k \cdot \Delta\xi^k \\ \lambda^{k+1} &:= \lambda^k + \alpha^k(\tilde{\lambda}^k - \lambda^k) \\ \mu^{k+1} &:= \mu^k + \alpha^k(\tilde{\mu}^k - \mu^k). \end{aligned}$$

5. *Wenn ein Konvergenztest nicht erfüllt ist, weiter bei Schritt 2 mit $k := k + 1$.*

Bemerkung 5.3.2

- *Die Wahl der Schrittweite $\alpha^k \in [0, 1]$ dient der Globalisierung der Konvergenz, zum Beispiel durch eine Line-Search-Strategie. Alternativ können auch Trust-Region-Methoden verwendet werden. Auf Globalisierungsstrategien soll in dieser Arbeit nicht eingegangen werden, siehe zum Beispiel [83].*

- *Die Quadratischen Probleme in Schritt 3 des SQP-Verfahrens können mit direkten oder iterativen Verfahren gelöst werden, wobei die Behandlung der Ungleichungen durch Active-Set- oder Interior-Point-Ansätze erfolgen kann. Sind Strukturen der zu behandelnden Probleme bekannt, können diese dabei sehr effizient ausgenutzt werden. Auch hierfür sei auf die einschlägige Literatur verwiesen, zum Beispiel [57, 119]. Die Verknüpfung von Active-Set- und Interior-Point-Ansätzen in einem Crossover-Verfahren wurde von Huber [67] untersucht.*

- *Die Lagrange-Multiplikatoren λ^k und μ^k werden unter anderem zur Bestimmung von α^k und bei der Active-Set-Strategie benötigt. Bei Vollschrittverfahren (das heißt $\alpha^k \equiv 1$) hängen λ^{k+1} und μ^{k+1} nicht von λ^k und μ^k ab.*

- *Als Abbruchkriterium kann zum Beispiel gewählt werden, daß die KKT-Bedingungen für (5.7)-(5.9) mit einer gewissen Toleranz erfüllt sind, das heißt im Rahmen der Toleranz ist die Lösung zulässig und der Gradient der Lagrangefunktion gleich null. Eine andere Möglichkeit ist, zu überprüfen, ob das Inkrement $\Delta\xi$ in Suchrichtung klein ist: $\|\nabla_\xi \mathcal{L}\, \Delta\xi\| < \epsilon$. Oder man testet, ob die Gütefunktion der Line-Search keine Änderungen mehr aufweist.*

5.3.3 Verhalten des Verfahrens in der Nähe des Minimums

Wenn man sich in einem lokalen Minimum des nichtlinearen Optimierungsproblems (5.7)-(5.9) befindet, dann bleiben SQP-Verfahren auch in diesem Minimum. Dies wird im folgenden Lemma bewiesen.

Dazu sind die folgenden Begriffsbildungen hilfreich:

- Für die Werte der Ungleichungen können folgende Fälle auftreten:

$$\psi_i(\xi) \begin{cases} \in (lo_i, up_i), & \text{die Ungleichung ist inaktiv,} \\ \leq lo_i, & \text{die untere Grenze ist aktiv,} \\ \geq up_i, & \text{die obere Grenze ist aktiv.} \end{cases}$$

  Die *Indexmenge der aktiven Ungleichungen* sei

$$I(\xi) := \{\bar{i}_1, \ldots, \bar{i}_{n_{aktiv}}\}$$

  mit

$$\bar{i}_j = \begin{cases} i_j, & \text{falls die untere Grenze aktiv ist} \\ -i_j, & \text{falls die obere Grenze aktiv ist} \end{cases}, \quad j = 1, \ldots, n_{aktiv}.$$

- Es gilt die sogenannte *Komplementarität*: Die Ungleichung $lo_i \leq \psi_i(\xi) \leq up_i$ ist aktiv, oder der zugehörige Lagrangemultiplikator μ ist null. Ist die untere Grenze aktiv, ist $\mu_i \geq 0$, ist die obere Grenze aktiv, ist $\mu_i \leq 0$

- Gilt zusätzlich die stärkere Eigenschaft $lo_i < \psi_i(\xi) < up_i$ oder $\mu_i \neq 0$, das heißt wenn eine Ungleichung aktiv ist, dann ist der zugehörige Lagrangemultiplikator ungleich null, so spricht man von *strikter Komplementarität.*

- Unter *Regularität* bzw. *Constraint Qualification* versteht man, daß die Matrix

$$\begin{pmatrix} \nabla\chi \\ \nabla\tilde{\psi} \end{pmatrix},$$

  $\tilde{\psi}$ aktive Ungleichungen, regulär ist.

Wir setzen von nun an bis zum Ende dieses Kapitels folgendes voraus:

Voraussetzung 5.3.3 *Das nichtlineare Optimierungsproblem (5.7)-(5.9) habe ein lokales Minimum ξ^* mit zugehörigen Lagrangemultiplikatoren λ^*, μ^*. Sei $I(\xi^*)$ die Indexmenge der aktiven Ungleichungen.*

In ξ^* gelte Constraint Qualification, strikte Komplementarität und die hinreichende Bedingung zweiter Ordnung, das heißt $\nabla^2_\xi \mathcal{L}(\xi^*, \lambda^*, \mu^*)$ ist positiv definit auf dem Tangentialraum $T(\xi^*)$ des Lösungsraums der Gleichungs- und aktiven Ungleichungsbedingungen in ξ^*

$$T(\xi^*) = \{\theta : \ \nabla\chi^T\theta = 0, \ \nabla\tilde{\psi}_i{}^T\theta = 0, \ i \in I(\xi^*)\}.$$

Lemma 5.3.4 (Verhalten von SQP-Verfahren in der Nähe der Lösung)
Sei H^* eine auf $T(\xi^*)$ positiv definite Matrix. Dann gilt: Das quadratische Problem QP(ξ^*)

$$\begin{aligned} \min_{\Delta\xi} \quad & \frac{1}{2}\Delta\xi^T H^* \Delta\xi + \nabla\phi(\xi^*)^T\Delta\xi \\ 0 \ &= \ \chi(\xi^*) + \nabla\chi(\xi^*)^T\Delta\xi \\ lo \ &\leq \ \psi(\xi^*) + \nabla\psi(\xi^*)^T\Delta\xi \leq up \end{aligned}$$

hat die Lösung $\Delta\xi = 0$ und die Lagrangemultiplikatoren $\tilde{\lambda} = \lambda^*$, $\tilde{\mu} = \mu^*$. Die Indexmenge der aktiven Ungleichungen von QP(ξ^*) im Lösungspunkt ist $I(\xi^*)$.

Beweis. Wir überprüfen die hinreichenden Bedingungen für die Optimalität von $\Delta\xi = 0$ für QP(ξ^*):

- Die Kuhn-Tucker-Bedingungen [57] von QP(ξ^*) sind erfüllt:

  $$\nabla_{\Delta\xi}\mathcal{L}_{QP} = H^*\Delta\xi + \nabla\phi(\xi^*) - \nabla\chi(\xi^*)\tilde{\lambda} - \nabla\psi(\xi^*)\tilde{\mu} = 0$$

  für $\Delta\xi = 0$, $\tilde{\lambda} = \lambda^*$, $\tilde{\mu} = \mu^*$ wegen der Minimalität von ξ^* für (5.7)-(5.9).

- Die Gleichungs- und Ungleichungsbedingungen von QP(ξ^*) sind erfüllt wegen der Zulässigkeit von ξ^* für (5.7)-(5.9)

- Es gelten nach Voraussetzung Regularität, strikte Komplementarität, und auf $T(\xi^*)$ ist $\nabla^2_{\Delta\xi}\mathcal{L}_{QP} = H^*$ positiv definit.

■

Aus Stetigkeitsgründen folgt für kleine Störungen von (QP) in der Nähe von ξ^*:

Korollar 5.3.5 *Es existiert ein $\epsilon > 0$, so daß für alle $\bar{\xi} \in U(\xi^*, \epsilon)$ gilt: Das quadratische Problem QP($\bar{\xi}$) mit positiv definitem $\bar{H}$ hat die Lösung $\Delta\xi \approx 0$ mit den Lagrangemultiplikatoren $\tilde{\lambda} \approx \lambda^*$, $\tilde{\mu} \approx \mu^*$ sowie die gleichen aktiven Ungleichungen wie QP(ξ^*): $I(\bar{\xi}) = I(\xi^*)$.*

Also konvergieren SQP-Verfahren nach Identifizierung der aktiven Indizes I in der Nähe der Lösung wie für gleichungsbeschränkte Probleme:

$$\begin{aligned} \min_{\xi} \quad & \phi(\xi) \\ 0 \quad &= \quad \hat{\chi}(\xi). \end{aligned}$$

$\hat{\chi}$ besteht dabei aus Gleichungen und aktiven Ungleichungen:

$$\hat{\chi} = \begin{pmatrix} \chi \\ (\psi_i - lo_i)_{i \in I, i>0} \\ (\psi_i - up_i)_{i \in I, i<0} \end{pmatrix}.$$

5.3.4 SQP- und Newton-Typ-Verfahren

In diesem Abschnitt wird die Verwandtschaft zwischen SQP-Verfahren und Newton-Typ-Verfahren diskutiert. Es wird sich zeigen, daß ein SQP-Verfahren mit exakter Hessematrix nichts anderes ist als ein Newton-Verfahren für die KKT-Bedingungen des nichtlinearen Problems.

Aufgrund der Überlegungen des vorherigen Abschnitts lassen wir nun hier die Ungleichungsbedingungen weg und bezeichnen der besseren Lesbarkeit halber $\hat{\chi}$ wieder als χ:

$$\min_{\xi} \quad \phi(\xi) \tag{5.10}$$

$$0 \quad = \quad \chi(\xi) \tag{5.11}$$

Mit der Lagrangefunktion $\mathcal{L}(\xi, \lambda) = \phi(\xi) - \lambda^T \chi(\xi)$ lauten dafür die Kuhn-Tucker-Bedingungen

$$\begin{aligned} \nabla_\xi \mathcal{L}(\xi, \lambda) = \nabla\phi(\xi) - \nabla\chi(\xi)\lambda \quad &= \quad 0 \\ \chi(\xi) \quad &= \quad 0. \end{aligned}$$

Dieses System schreiben wir kurz als $F(\xi, \lambda) = 0$.

Um dieses nichtlineare Gleichungssystem mit einem *Newton-Verfahren* zu lösen, starten wir mit $\begin{pmatrix} \xi^0 \\ \lambda^0 \end{pmatrix}$ und berechnen iterativ für $k = 0, 1, \ldots$:

$$\begin{pmatrix} \xi^{k+1} \\ \lambda^{k+1} \end{pmatrix} := \begin{pmatrix} \xi^k \\ \lambda^k \end{pmatrix} + \alpha^k \cdot \begin{pmatrix} \Delta\xi^k \\ \Delta\lambda^k \end{pmatrix}.$$

Dabei ist das Inkrement $\begin{pmatrix} \Delta\xi^k \\ \Delta\lambda^k \end{pmatrix}$ durch die Lösung des linearisierten Systems

$$F(\xi^k, \lambda^k) + F'(\xi^k, \lambda^k) \begin{pmatrix} \Delta\xi^k \\ \Delta\lambda^k \end{pmatrix} = 0$$

bestimmt, das heißt

$$\begin{pmatrix} \nabla_\xi \mathcal{L}(\xi^k, \lambda^k) \\ \chi(\xi^k) \end{pmatrix} + \begin{pmatrix} \nabla^2_\xi \mathcal{L}(\xi^k, \lambda^k) & \nabla\chi(\xi^k) \\ \nabla\chi(\xi^k)^T & 0 \end{pmatrix} \begin{pmatrix} \Delta\xi^k \\ -\Delta\lambda^k \end{pmatrix} = 0. \tag{5.12}$$

Wenn wir nun

$$\tilde{\lambda}^k := \lambda^k + \Delta\lambda^k$$

definieren, ist (5.12) wegen

$$\nabla_\xi \mathcal{L}(\xi^k, \lambda^k) - \nabla\chi(\xi^k)\Delta\lambda^k = \nabla\phi(\xi^k) - \nabla\chi(\xi^k)\lambda^k - \nabla\chi(\xi^k)\Delta\lambda^k = \nabla\phi(\xi^k) - \nabla\chi(\xi^k)\tilde{\lambda}^k$$

äquivalent zu

$$\begin{pmatrix} \nabla\phi(\xi^k) \\ \chi(\xi^k) \end{pmatrix} + \begin{pmatrix} \nabla^2_\xi \mathcal{L}(\xi^k, \lambda^k) & \nabla\chi(\xi^k) \\ \nabla\chi(\xi^k)^T & 0 \end{pmatrix} \begin{pmatrix} \Delta\xi^k \\ -\tilde{\lambda}^k \end{pmatrix} = 0.$$

Das sind aber genau die Kuhn-Tucker-Bedingungen für das Quadratische Problem in der k-ten Iteration des SQP-Verfahrens für Problem (5.10)-(5.11)

$$\begin{aligned} \min_{\Delta\xi^k} \quad & \frac{1}{2}\Delta{\xi^k}^T H^k \Delta\xi^k + \nabla\phi(\xi^k)^T \Delta\xi^k \\ 0 \quad = \quad & \chi(\xi^k) + \nabla\chi(\xi^k)^T \Delta\xi^k, \end{aligned}$$

wenn man $H^k = \nabla^2_\xi \mathcal{L}(\xi^k, \lambda^k)$ wählt.

Damit ist folgendes Lemma bewiesen:

Lemma 5.3.6 *Wählt man $H^k = \nabla^2_\xi \mathcal{L}(\xi^k, \lambda^k)$, ist das SQP-Verfahren äquivalent zum Newton-Verfahren für die Kuhn-Tucker-Bedingungen des nichtlinearen Optimierungsproblems.*

Damit erhalten wir das folgende Resultat über die Konvergenzrate von SQP-Verfahren mit exakten Hessematrizen:

Korollar 5.3.7 *SQP-Verfahren mit $H^k = \nabla^2_\xi \mathcal{L}(\xi^k, \lambda^k)$ konvergieren lokal quadratisch.*

Im allgemeinen wird man aber nur $H^k \approx \nabla^2_\xi \mathcal{L}(\xi^k, \lambda^k)$ wählen. Wie man dafür noch superlineare Konvergenz erzielen kann, wird im nächsten Abschnitt diskutiert.

5.3.5 Update-Formeln für die Hessematrix

Die vorigen Überlegungen zeigen also, daß in der Nähe der Lösung die exakte Hessematrix der Lagrangefunktion $\nabla^2_\xi \mathcal{L}(\xi^k, \lambda^k)$ die ideale Wahl für H^k ist.

Weiter weg von der Lösung kann $\nabla^2_\xi \mathcal{L}(\xi^k, \lambda^k)$ jedoch auch nicht positiv definit sein, was Schwierigkeiten bei der Behandlung der quadratischen Probleme bedeutet. (SQP-Verfahren mit indefiniten Hessematrizen werden von Schäfer untersucht, siehe [102].)

Da die Berechnung von exakten Hessematrizen außerdem einen erheblichen numerischen Aufwand mit sich bringt, begnügt man sich mit Näherungen von $\nabla^2_\xi \mathcal{L}$. Update-Formeln liefern eine Möglichkeit, diese zur Verfügung zu stellen. Unter Update versteht man die Berechnung der Näherung H^{k+1} aus der Näherung H^k. Initialisiert werden kann H zu Beginn oder auch nochmals in einer späteren Iteration zum Beispiel mit der exakten Hessematrix oder auch einfach mit der Einheitsmatrix.

Wir betrachten die Taylorentwicklung des Gradienten der Lagrangefunktion

$$\nabla_\xi \mathcal{L}(\xi^k, \lambda^{k+1}) = \nabla_\xi \mathcal{L}(\xi^{k+1}, \lambda^{k+1}) + \nabla^2_\xi \mathcal{L}(\xi^{k+1}, \lambda^{k+1})(\xi^k - \xi^{k+1}) + O(\|\xi^{k+1} - \xi^k\|^2)$$

und verlangen, daß die Beziehung

$$\nabla^2_\xi \mathcal{L}(\xi^{k+1}, \lambda^{k+1})(\xi^{k+1} - \xi^k) = \nabla_\xi \mathcal{L}(\xi^{k+1}, \lambda^{k+1}) - \nabla_\xi \mathcal{L}(\xi^k, \lambda^{k+1}) + O(\|\xi^{k+1} - \xi^k\|^2)$$

bis auf die Terme höherer Ordnung auch von der Approximation H^{k+1} erfüllt wird:

$$H^{k+1} s^k = y^k \tag{5.13}$$

mit

$$\begin{aligned} s^k &:= \xi^{k+1} - \xi^k = \alpha_k \Delta\xi^k \\ y^k &:= \nabla_\xi \mathcal{L}(\xi^{k+1}, \lambda^{k+1}) - \nabla_\xi \mathcal{L}(\xi^k, \lambda^{k+1}). \end{aligned}$$

Gleichung (5.13) heißt *Sekantenbedingung*. Sie ergibt aber erst n Gleichungen für die n^2 Freiheitsgrade in H^{k+1}. Folgende Varianten können nun betrachtet werden:

- Symmetrie: ergibt weitere: $n(n-1)/2$ Bedingungen.
- H^k positiv definit $\rightsquigarrow$ H^{k+1} positiv definit: weitere n Ungleichungen (Hauptminoren positiv).

Notwendig für positive Definitheit ist die *Krümmungsbedingung*:

$$0 < s^{k^T} H^{k+1} s^k = y^{k^T} s^k. \tag{5.14}$$

Ein weiteres Kriterium ist die Übernahme von Information aus H^k:

$$H^{k+1} \approx H^k.$$

Bemerkung 5.3.8 *[83] Sucht man unter allen symmetrischen Matrizen, die die Sekantenbedingung erfüllen, als H^{k+1} diejenige, die am nächsten zur aktuellen Matrix H^k im Sinne einer Norm liegt, und wählt man dabei die gewichtete Frobenius-Norm, so erhält man als eindeutige Lösung die DFP-Update-Formel (5.16). Eine analoge Betrachtung für die Inverse von H^{k+1} führt als eindeutige Lösung auf die BFGS-Update-Formel (5.15).*

Rang-1-Updates

Bemerkung 5.3.9 *Jede $n \times n$-Matrix vom Rang 1 hat eine Darstellung*

$$a \cdot b^T, \quad a, b \in I\!R^n.$$

Die einfachsten Updates bestehen im Aufaddieren von Rang-1-Matrizen auf H^k.

Dabei soll die Sekantenbedingung gelten. Deshalb definieren wir

$$w^k := y^k - H^k s^k$$

und betrachten Updates der Form

$$H^{k+1} := H^k + \frac{w^k z^T}{z^T s^k}.$$

Diese erfüllen die Sekantenbedingung:

$$H^{k+1} s^k = H^k s^k + \frac{w^k z^T s^k}{z^T s^k} = H^k s^k + w^k = y^k.$$

Die Wahl von z ist noch frei.

Die Update-Formel von Broyden benutzt $z = s^k$:

$$H^{k+1} := H^k + \frac{w^k z^T}{{s^k}^T s^k}.$$

Sie hat die Eigenschaft, daß

$$H^{k+1} p = H^k p \quad \forall p \perp s^k$$

H^{k+1} ist aber im allgemeinen nicht symmetrisch.

In der Powell-symmetric-Broyden-Formel (PSB) wird $z = w^k$ gewählt:

$$H^{k+1} := H^k + \frac{w^k z^T}{{w^k}^T s^k}.$$

Dieses Update ist symmetrisch, erhält aber im allgemeinen keine Positiv-Definitheit, nur dann mit Sicherheit, wenn ${w^k}^T s^k > 0$ ist.

Rang-2-Updates

Bemerkung 5.3.10 *Jede $n \times n$-Matrix vom Rang 2 hat eine Darstellung*

$$a \cdot b^T + c \cdot d^T, \quad a, b, c, d \in \mathbb{R}^n.$$

Rang-2-Updates erlauben nun, Sekantenbedingung und Symmetrie zu erfüllen. Die gebräuchlichsten sind dabei

BFGS-Update (Broyden, Fletcher, Goldfarb, Shanno)

$$H^{k+1} = H^k + \frac{y^k y^{k^T}}{y^{k^T} s^k} - \frac{(H^k s^k)(H^k s^k)^T}{(H^k s^k)^T s^k} \tag{5.15}$$

und DFP-Update (Davidon, Fletcher, Powell)

$$H^{k+1} = H^k + \frac{w^k y^{k^T} + y^k w^{k^T}}{y^{k^T} s^k} - \frac{s^{k^T} w^k}{y^{k^T} s^k} \cdot \frac{y^k y^{k^T}}{y^{k^T} s^k}. \tag{5.16}$$

Beide Formeln erfüllen die Sekantenbedingung. Für beide Formeln gilt: H^{k+1} ist positiv definit, wenn H^k positiv definit ist und die Krümmungsbedingung erfüllt ist.

Mit der *Powell-Modifikation*, siehe zum Beispiel [83] kann Positiv-Definitheit erzwungen werden:

Falls $y^{k^T} s^k \leq 0$, setze

$$\tilde{y}^k := \alpha\, y^k + (1 - \alpha)\, H^k s^k, \quad \alpha \in [0, 1],$$

so daß

$$(\tilde{y}^k)^T s^k = \underbrace{\alpha\, y^{k^T} s^k}_{\leq 0} + \underbrace{(1 - \alpha)\, s^{k^T} H^k s^k}_{>0} > 0.$$

Wähle α so, daß

$$(\tilde{y}^k)^T s^k = 0.2 \cdot s^{k^T} H^k s^k > 0,$$

das heißt

$$\alpha = 0.8 \cdot \frac{s^{k^T} H^k s^k}{s^{k^T} H^k s^k - y^{k^T} s^k}.$$

Dann ist allerdings die Sekantenbedingung nicht mehr erfüllt.

Updates der Inversen

Bei vielen Implementierungen von QP-Lösern wird nicht nur die Hessematrix, sondern vor allem deren Inverse benötigt. Die Rang-1- bzw. Rang-2-Update-Formeln ermöglichen die einfache Berechnung davon, ebenfalls durch Rang-1- bzw. Rang-2-Update-Formeln.

Das folgende Lemma zeigt, daß sich die Inverse eines Rang-1-Updates durch die Sherman-Morrison-Formel darstellen läßt.

Lemma 5.3.11 (Sherman-Morrison-Formel) *Seien $A \in \mathbb{R}^{n\times n}$ und $x, y \in \mathbb{R}^n$ gegeben mit A regulär und $y^T A^{-1} x \neq -1$. Dann ist auch $A + xy^T$ regulär, und es gilt:*

$$\left(A + xy^T\right)^{-1} = A^{-1} - \frac{A^{-1}xy^TA^{-1}}{1 + y^TA^{-1}x}.$$

Beweis. Nachrechnen. ∎

Die Inverse des BFGS-Updates

$$H^{k+1} = H^k + \frac{y^k y^{kT}}{y^{kT} s^k} - \frac{(H^k s^k)(H^k s^k)^T}{(H^k s^k)^T s^k}$$

kann ebenfalls durch

$$\left(H^{k+1}\right)^{-1} = \left(H^k\right)^{-1} + \frac{u^k s^{kT} + s^k u^{kT}}{y^{kT} s^k} - \frac{y^{kT} u^k}{y^{kT} s^k} \cdot \frac{s^k s^{kT}}{y^{kT} s^k}$$

mit $u^k := s^k - \left(H^k\right)^{-1} y^k$ explizit angegeben werden.

Man beobachtet, daß dies gerade die DFP-Update-Formel ist, angewendet auf $\left(H^k\right)^{-1}$ mit s^k und y^k vertauscht, so daß $\left(H^{k+1}\right)^{-1} y^k = s^k$.

Diese Tatsache bezeichnet man damit, daß die Updates BFGS und DFP *zueinander dual* sind.

Mit den Abkürzungen

$$H^{k+1} = H^k + U_{BFGS}\left(s^k, y^k, H^k\right)$$

und

$$\left(H^{k+1}\right)^{-1} = \left(H^k\right)^{-1} + U_{DFP}\left(y^k, s^k, \left(H^k\right)^{-1}\right)$$

kann man nun allgemeiner die *(eingeschränkte) Broyden-Familie* von Rang-2-Updates definieren:

$$U_\theta\left(s^k, y^k, H^k\right) := \theta \cdot U_{BFGS}\left(s^k, y^k, H^k\right) + (1-\theta) \cdot U_{DFP}\left(s^k, y^k, H^k\right), \quad \theta \in [0,1].$$

Diese Updates erfüllen alle die Sekantenbedingung und erhalten die Positiv-Definitheit, wenn die Krümmungsbedingung erfüllt ist.

Außerdem gilt: Für jedes $\theta \in [0,1]$ gibt es ein $\theta' \in [0,1]$, so daß $U^{\theta'}$ zu U^{θ} dual ist:

$$\theta' = \frac{\theta - 1}{\theta - 1 - \theta \cdot \mu}, \quad \mu = \frac{s^{k^T} H^k s^k \cdot y^{k^T} \left(H^k\right)^{-1} y^k}{\left(s^{k^T} y^k\right)^2} \geq 1.$$

Es gilt: Alle Updates der Broyden-Familie liefern (bei exakter Line-Search) dieselbe Folge von Iterierten.

Eine detaillierte Untersuchung der Broyden-Familie findet sich in [83].

Konvergenzresultate für Rang-2-Updates

Seien die Matrizen H^k symmetrisch, nichtsingulär und gleichmäßig beschränkt. Außerdem seien die H^k gleichmäßig positiv definit auf dem Kern von $\nabla\chi(\xi^k)^T$, das heißt es gebe ein $\beta > 0$, so daß für $\xi \neq 0$, $\nabla\chi(\xi^k)^T\xi = 0$ für alle k die Beziehung $\xi^T H^k \xi \geq \beta\|\xi\|^2$ gilt. In anderen Worten soll H^k also die notwendigen Bedingungen 2. Ordnung erfüllen, die auch für die Hessematrix $\nabla^2_\xi \mathcal{L}(\xi^*, \lambda^*)$ der Lagrangefunktion im Lösungspunkt gelten sollen, siehe Voraussetzung 5.3.3.

Wir betrachten nun die Matrix, die Vektoren auf den Tangentialraum der Niveaumengen der Nebenbedingungen projiziert. Sie hat die Darstellung

$$P^k := P(\xi^k) := \left(I - \nabla\chi(\xi^k)(\nabla\chi(\xi^k)^T\nabla\chi(\xi^k))^{-1}\nabla\chi(\xi^k)^T\right).$$

Die Konvergenz der ξ^k gegen ξ^* heißt *Q-superlinear*, wenn gilt: Es existiert eine Folge $\{\rho^k\} \to 0$ für $k \to \infty$, so daß

$$\left\|\xi^{k+1} - \xi^k\right\| \leq \rho^k \left\|\xi^k - \xi^{k-1}\right\|. \tag{5.17}$$

Boggs, Tolle und Wang [30] haben das folgende Konvergenzresultat bewiesen:

Satz 5.3.12 *Gelten für die Matrizen H^k die obigen Bedingungen, und die Folge der ξ^k konvergiere linear gegen ξ^*. Diese Konvergenz ist Q-superlinear genau dann, wenn*

$$\lim_{k\to\infty} \frac{\left\|P^k\left(H^k - \nabla^2_\xi \mathcal{L}(\xi^*, \lambda^*)\right)\Delta\xi^k\right\|}{\|\Delta\xi^k\|} = 0. \tag{5.18}$$

Und weiter:

Korollar 5.3.13 *Sei $\nabla^2_\xi \mathcal{L}(\xi^*, \lambda^*)$ positiv definit. Wenn die Matrizen H^k entweder durch die BFGS- oder die DFP-Update-Formeln definiert werden und wenn die Folge der ξ^k linear gegen ξ^* konvergiert, dann ist die Konvergenz Q-superlinear.*

Die Annahme der linearen Konvergenz, die hier jeweils gemacht wurde, kann gesichert werden, wenn die Inversen der Hessematrizen gleichmäßig beschränkt sind:

Satz 5.3.14 *Existiere $\eta \geq 0$, so daß $\|H^{k^{-1}}\| \leq \eta$ für alle k. Dann gibt es positive Konstanten $\epsilon > 0$ und $\lambda > 0$, so daß wenn*

1. $\|\xi^0 - \xi^*\| < \lambda$
2. $\left\|P(x^*)\left(H^k - \nabla^2_\xi \mathcal{L}(\xi^*, \lambda^*)\right)\right\| < \epsilon$ *für alle* $k \geq 0$,

die Folge der ξ^k im Vollschritt-SQP-Verfahren linear gegen ξ^* konvergiert.

Für die Beweise dieser Resultate sei auf [30] verwiesen.

5.4 Erzeugung gemischt-ganzzahliger Lösungen

Die ganzzahligen Aspekte des Versuchsplanungs-Optimierungsproblems (5.4) resultieren aus der Modellierung der Auswahl der Messungen, siehe Abschnitt 5.1.2. Gegeben seien M mögliche Meßpunkte

$$t_0 \leq t_1 \leq \ldots \leq t_M \leq t_{end}.$$

Die Gewichte der Messungen sind ganzzahlige Variablen

$$w_i \in \{0, 1\}, \quad i = 1, \ldots, M.$$

Messungen, die tatsächlich durchgeführt werden sollen, erhalten das Gewicht $w_i = 1$, Messungen, die nicht ausgewählt werden, haben das Gewicht $w_i = 0$.

Diese Ganzzahligkeitsbedingung kann man relaxieren, zum Beispiel durch

$$w_i \in [0, 1], \quad i = 1, \ldots, M$$

oder

$$w_i \in \mathbb{R}_0^+, \quad i = 1, \ldots, M.$$

Auch dies erlaubt eine Interpretation: Eine Messung i mit Gewicht w_i wird mit w_i-fachem „Aufwand“ durchgeführt, das heißt mit w_i^{-1}-facher Varianz und w_i-fachen Kosten.

An die Meßgewichte können Nebenbedingungen gestellt sein. Wir nehmen an, daß nur lineare Nebenbedingungen vorkommen, die die Maximalzahlen von Messungen für Teilmengen von $1, \ldots, M$ beschreiben. Im allgemeinen haben diese die Form

$$lo_{J_j} \leq \sum_{i \in J_j} w_i \leq up_{J_j}, \quad j = 1, \ldots, N_J$$

für Indexmengen $J_j \subseteq \{1, \ldots, M\},\ j = 1, \ldots, N_J$.

Solche Nebenbedingungen durch 0-1-Variablen zu erfüllen, ist im allgemeinen NP-vollständig [81] und führt auf Aufgaben der gemischt-ganzzahligen nichtlinearen Optimierung. Einen Überblick über Methoden zur Lösung solcher Probleme gibt zum Beispiel Floudas [48]. Eine typische Schwierigkeit dabei ist neben der kombinatorischen Natur der Ganzzahligkeitsbedingungen, daß die Nichtlinearitäten im allgemeinen nichtkonvex sind. Daher gibt es möglicherweise mehrere lokale Lösungen, die Bestimmung der globalen Lösung eines nichtkonvexen Problems ist im allgemeinen ebenfalls NP-schwer [48]. Da in typischen Algorithmen für gemischt-ganzzahlige nichtlineare Probleme die Lösungen nichtlinearer Teilprobleme als Schranken dienen, sind dabei aber globale Optima zu bestimmen.

Wir begnügen uns in dieser Arbeit daher mit der heuristischen Erzeugung ganzzahliger Lösungen, die die Nebenbedingungen erfüllen.

Bei der Anwendung von Heuristiken nutzen wir die Strukturen aus, die die Nebenbedingungen in typischen Anwendungsproblemen aufweisen. Beispiele dafür sind

- Mehrfachexperimente, siehe Kapitel 7. In der Regel haben die einzelnen Experimente getrennte Beschränkungen an die Gewichte.
- Meßverfahren. Durch den experimentellen Aufbau und die vorhandenen Apparaturen können verschiedene Meßmethoden vorgegeben sein. Während einige Verfahren nur selten eingesetzt werden können, da sie aufwendig und kostenintensiv sind, fallen andere Daten sehr einfach an. Typischerweise werden Beschränkungen an die Anzahl der teuren Messungen formuliert.

  Sei $\{1, \ldots, M\}$ die Indexmenge aller Messungen. Seien n_V verschiedene Meßverfahren gegeben, V_j sei die Indexmenge der Messungen für Verfahren $j = 1, \ldots, n_V$. Dann sind die
  $$V_j \subseteq \{1, \ldots, M\},\ j = 1, \ldots, n_V \tag{5.19}$$
  disjunkte Teilmengen:
  $$\bigcup_{j=1,\ldots,n_V} V_j = \{1, \ldots, M\}, \quad V_i \cap V_j = \{\},\ i \neq j. \tag{5.20}$$
  Die Beschränkung der Anzahl der Messungen pro Verfahren wird beschrieben durch
  $$0 \leq \sum_{i \in V_j} w_i \leq up_j, \quad j = 1, \ldots, n_V. \tag{5.21}$$

In beiden Beispielen haben wir eine Entkopplung der Gewichtenebenbedingungen. Eine getrennte Behandlung macht die Anwendung von Heuristiken einfacher.

Neben den Gewichtebedingungen muß eventuell auch die Kostenbeschränkung 5.3 berücksichtigt werden. Außerdem muß gewährleistet sein, daß die Identifizierbarkeit aller Parameter, also die Bedingung (PD) (4.17), erfüllt bleibt.

Wir schlagen für Nebenbedingungen der Form (5.19)-(5.21) die folgenden Heuristiken vor [16], siehe auch Abbildung 5.2:

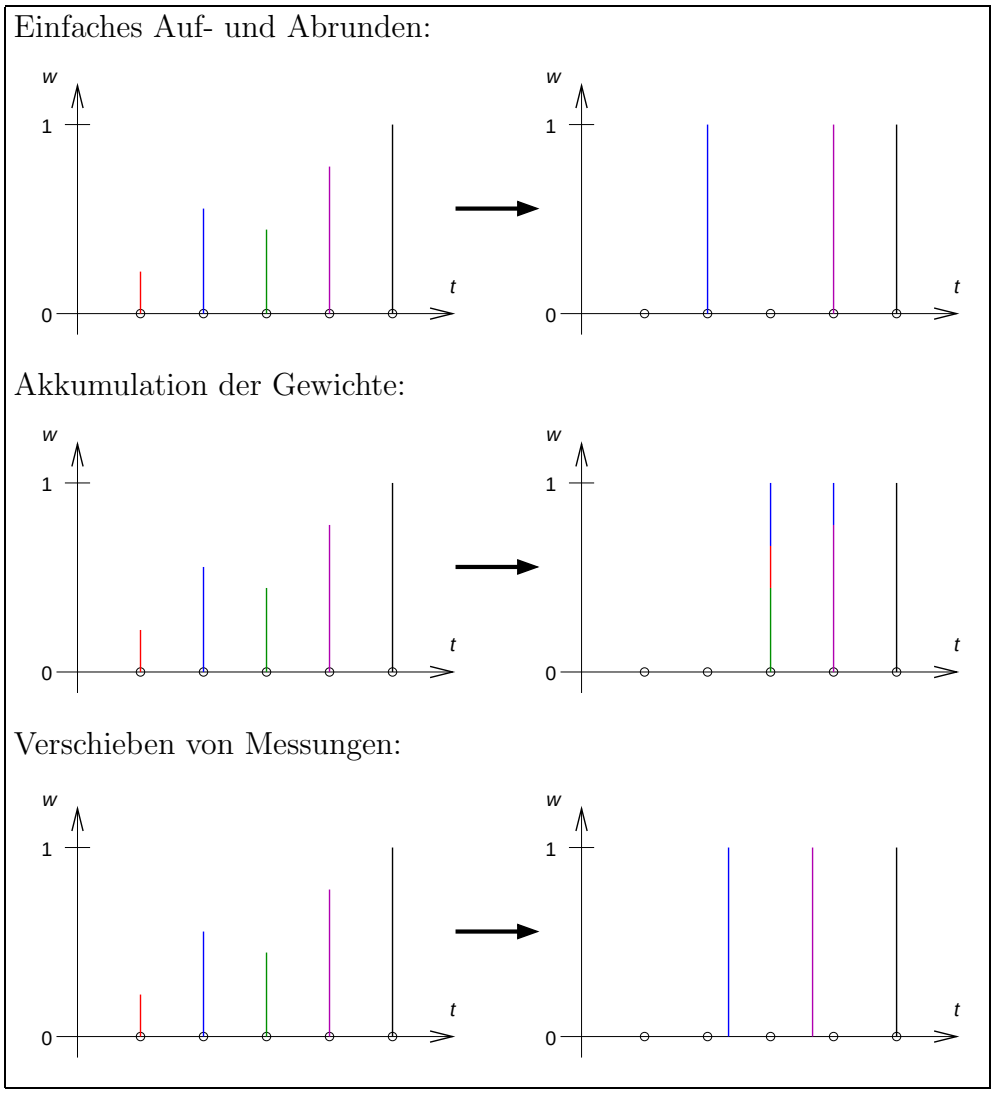


Abbildung 5.2: Illustration der Rundungsheuristiken

- Einfaches Auf- oder Abrunden: Die up_j (Maximalzahl von Messungen für Verfahren j) größten Gewichte werden auf-, die anderen abgerundet.
- Akkumulation der Gewichte: Man betrachtet sukzessive die Gewichte der Messungen für Verfahren j und bildet jeweils die Summe aller bisher betrachteten Gewichte. Wird diese Summe größer oder gleich 1, plaziert man eine Messung, zieht 1 von der akkumulierten Summe ab und fährt dann fort. Auf diese Weise sammelt man also so lange Gewichte, bis jeweils eine volle Messung komplett ist.
- Zusammenfassen von Messungen: Man plaziert volle Messungen so, daß sie im Schwerpunkt von Gruppen von Messungen mit gebrochenen Gewichten liegen und als Gewicht die Summe deren Gewichte erhalten. Dabei können auch Zeitpunkte gewählt werden, die nicht zu den ursprünglich gegebenen Meßzeitpunkten gehören.

Alle Heuristiken sind interaktiv vom Benutzer anzuwenden. Mit der Simulationsumgebung, siehe Kapitel 9, stellen wir ein machtvolles Softwarewerkzeug zur Verfügung, das eine schnelle Beurteilung der gewählten Strategien hinsichtlich Güte des Versuchsplans, Kosten und Erfüllung der Nebenbedingungen erlaubt.

Unsere numerischen Ergebnisse zeigen, daß die Ganzzahligkeitsbedingungen oft auch für die Lösungen der relaxierten Probleme schon beinahe erfüllt sind. Anwendung der Rundungsheuristiken ändert dann den Zielfunktionswert nur noch unwesentlich. Für Beispiele sei auf die Abschnitte 10.1 und 10.2 in Kapitel 10 verwiesen.

Diese Beobachtung wird durch die Resultate von Pukelsheim [93] für (lineare) Polynom-Fit-Modelle gestützt:

Gegeben sei ein Polynom-Fit-Modell vom Grad d:

$$\eta_{ij} = \sum_{k=0}^{d} p_k \cdot q_i^k + \epsilon_{ij}, \quad j = 1, \dots n_i, \quad i = 1, \dots, M.$$

Am Meßpunkt i, der durch die Steuerung q_i charakterisiert wird, werden dabei $n_i \in \mathbb{N}$ Proben genommen, $i = 1, \dots, M$. Die q_i seien paarweise verschieden. Die Gesamtzahl der Messungen sei $\sum_{i=1}^{M} n_i = n$.

Die Ganzzahligkeit der Meßhäufigkeiten wird nun dadurch relaxiert, daß die n_i ersetzt werden durch $n \cdot w_i$ mit Gewichten $w_i \in [0; 1]$, $i = 1, \dots, M$.

Ein Design besteht aus den Steuerungen q_i und den Gewichten w_i, $i = 1, \dots, M$. Der Träger des Designs ist die Menge $\{q_i : \ w_i > 0\}$.

Für diese Situation zeigt Pukelsheim [93] die folgenden Resultate:

Satz 5.4.1 *[93] Für eine Klasse von Gütekriterien, zu der auch das A-, D- und E-Kriterium gehören, gilt: Ein Design für ein Polynom-Fit-Modell vom Grad d ist optimal genau dann, wenn sein Träger aus $d + 1$ Punkten besteht. Für das D-Kriterium gilt zusätzlich: Für das optimale Design sind die Gewichte alle gleich.*

Pukelsheim gibt weiter [94, 93] eine „Effiziente Rundungstechnik“ für Probleme der linearen Versuchsplanung an, die beim Runden der Gewichte möglichst wenig Verlust an Güte bewirkt. Dafür kann er zeigen:

Satz 5.4.2 *[93] Durch die „Effiziente Rundungstechnik“ wird ein D-optimales Design mit gebrochenen Gewichten auf ein D-optimales Design mit ganzzahligen Meßhäufigkeiten gerundet.*

Kapitel 6

Ableitungserzeugung

Zur Lösung von Versuchsplanungsoptimierungsproblemen verwenden wir die in Abschnitt 5.3 vorgestellten gradientenbasierten SQP-Verfahren. Dazu müssen neben den Funktionsauswertungen der Zielfunktion und der Nebenbedingungen auch Gradienten nach den Optimierungsvariablen, also den Versuchsplanungsgrößen $\xi = (q, s, w)$, bereitgestellt werden. Bei Versuchsplanungsproblemen haben wir es mit einer sehr komplexen Zielfunktion zu tun, die auf den Sensitivitäten der Lösung des zugrundeliegenden DAE-Systems definiert ist. Um dafür effektiv und effizient Ableitungen berechnen zu können, benutzen wir die Techniken der Automatischen Numerischen Differentiation und der Internen Numerischen Differentiation. Dies wird in diesem Kapitel dargestellt. Die ebenfalls von SQP-Verfahren benötigte Näherung der Hessematrix der Lagrangefunktion des Optimierungsproblems wird mittels Updates berechnet, siehe Abschnitt 5.3.5.

6.1 Berechnung der Ableitung der Nebenbedingungen nach den Versuchsplanungsgrößen

Zunächst betrachten wir die Berechnung der Ableitungen der nichtlinearen Nebenbedingungen. Bei den Zustandsbeschänkungen

$$lo \leq b(t, x(t), p, q) \leq up$$

wenden wir die Kettenregel an, um nach den Zustandsvariablen nachzudifferenzieren:

$$\frac{\mathrm{d}b}{\mathrm{d}q} = \frac{\partial b}{\partial x}\frac{\partial x}{\partial q} + \frac{\partial b}{\partial q}$$

sowie

$$\frac{\mathrm{d}b}{\mathrm{d}s} = \frac{\partial b}{\partial x}\frac{\partial x}{\partial s}.$$

Mit der Berechnung von $\frac{\partial x}{\partial q}$ und $\frac{\partial x}{\partial s}$ beschäftigt sich Abschnitt 6.2.7, siehe unten.

Die Steuerbeschränkungen

$$lo \leq c(q) \leq up$$

hängen nur von den Steuergrößen ab:

$$\frac{\mathrm{d}c}{\mathrm{d}q} = \frac{\partial c}{\partial q}.$$

Daneben gibt es auch lineare Nebenbedingungen, nämlich Gewichtebedingungen und Kostenbeschränkungen. Für diese liegen die Ableitungen direkt vor.

Die Ableitungen der Nebenbedingungen sind also einfach bereitzustellen. Dagegen hängt die Zielfunktion in äußerst komplexer Weise von den Versuchsplanungsgrößen ab. Wie davon die im SQP-Verfahren benötigten Funktionsauswertungen und Gradienten berechnet werden können, wird im nächsten Abschnitt dargestellt.

6.2 Berechnung der Ableitung der Zielfunktion nach den Versuchsplanungsgrößen

Die Zielfunktion der Versuchsplanungsoptimierung besteht aus einem der Gütekriterien, das auf der Kovarianzmatrix operiert. Die folgenden Formeln zeigen den gesamten Berechnungsweg.

Die Gütekriterien sind auf der Kovarianzmatrix definiert:

$$\phi = \begin{cases} \frac{1}{n} \cdot \mathrm{Spur} C \\ \left(\det(K^T C K)\right)^{\frac{1}{n}} \\ \max\{\lambda : \lambda \text{ Eigenwert von } C\} \\ \max\{\sqrt{C_{ii}}, i = 1, \ldots, n\}. \end{cases}$$

Die Kovarianzmatrix hängt von der Jacobimatrix ab:

$$C = \begin{pmatrix} I & 0 \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_1^T J_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-T} \begin{pmatrix} I \\ 0 \end{pmatrix}.$$

Und in die Jacobimatrix gehen, direkt und indirekt über die Lösung x des DAE-Systems und ihre Ableitung $\frac{\partial x}{\partial v}$, die Versuchsplanungsgrößen $\xi = (q, s, w)$ ein:

$$J = \begin{pmatrix} J_1 \\ J_2 \end{pmatrix} = \begin{pmatrix} -\sqrt{W}\,\Sigma^{-1}\,\frac{\mathrm{d}h}{\mathrm{d}v} \\ \frac{\mathrm{d}d}{\mathrm{d}v} \end{pmatrix} = \begin{pmatrix} -\left(\frac{\sqrt{w_i}}{\varsigma_i} \cdot \left(\frac{\partial h_i}{\partial x}\frac{\partial x}{\partial v} + \frac{\partial h_i}{\partial v}\right)\right)_{i=1,\ldots,M} \\ \frac{\partial d}{\partial x}\frac{\partial x}{\partial v} + \frac{\partial d}{\partial v} \end{pmatrix}$$

$$x = x(t, p, q, u, s).$$

Bei Verwendung eines SQP-Verfahrens müssen wir Ableitungen der Zielfunktion nach den Optimierungsvariablen $\xi = (q, s, w)$ bereitstellen.

Wir betrachten Richtungsableitungen. Sei eine Richtung $\Delta\xi$ gegeben. Die Richtungsableitung von ϕ in Richtung $\Delta\xi$ ist

$$\frac{\mathrm{d}\phi}{\mathrm{d}\xi}\,\Delta\xi := \lim_{\epsilon\to 0}\frac{\phi(\xi+\epsilon\cdot\Delta\xi)-\phi(\xi)}{\epsilon}.$$

Nach der Kettenregel können wir die Ableitung schrittweise berechnen:

$$\Delta\phi := \frac{\mathrm{d}\phi}{\mathrm{d}\xi}\,\Delta\xi = \frac{\mathrm{d}\phi}{\mathrm{d}C}\,\frac{\mathrm{d}C}{\mathrm{d}J}\,\frac{\mathrm{d}J}{\mathrm{d}\xi}\,\Delta\xi$$

bzw.

$$\Delta\phi = \frac{\mathrm{d}\phi}{\mathrm{d}C}\,\Delta C \tag{6.1}$$

$$\Delta C := \frac{\mathrm{d}C}{\mathrm{d}J}\,\Delta J \tag{6.2}$$

$$\Delta J := \frac{\mathrm{d}J}{\mathrm{d}\xi}\,\Delta\xi. \tag{6.3}$$

Die folgenden Abschnitte befassen sich mit der Berechnung dieser einzelnen Schritte (6.1)-(6.3).

6.2.1 Ableitungen von Matrizen nach Matrizen

In den nächsten zwei Lemmata sind einige elementare Formeln zur Ableitung von Matrix-wertigen Abbildungen zusammengestellt, die wir später benutzen werden.

Lemma 6.2.1 *Seien $A, \Delta A \in \mathbb{R}^{m\times n}$, $U \in \mathbb{R}^{m\times m}$ und $V \in \mathbb{R}^{n\times n}$ Matrizen. Dann gilt für die Richtungsableitungen der folgenden Matrix-wertigen Abbildungen nach A in der Richtung ΔA:*

$$\frac{d}{dA}(UAV)\,\Delta A = U\,\Delta A\,V \tag{6.4}$$

$$\frac{d}{dA}(A^T)\,\Delta A = \Delta A^T \tag{6.5}$$

$$\frac{d}{dA}(A^TA)\,\Delta A = \Delta A^T\,A + A^T\,\Delta A. \tag{6.6}$$

Sei $A \in \mathbb{R}^{n\times n}$ regulär, $\Delta A \in \mathbb{R}^{n\times n}$. Dann gilt:

$$\frac{d}{dA}(A^{-1})\,\Delta A = -A^{-1}\,\Delta A\,A^{-1}. \tag{6.7}$$

Beweis. Zunächst rechnen wir (6.4) und (6.5) elementar nach:

$$\begin{aligned}\left(\frac{\mathrm{d}}{\mathrm{d}A}(U\,A\,V)\Delta A\right)_{ij} &= \sum_{r,s}\frac{\mathrm{d}}{\mathrm{d}A_{rs}}(U\,A\,V)_{ij}\Delta A_{rs} = \sum_{r,s}\frac{\mathrm{d}}{\mathrm{d}A_{rs}}\sum_{k,l}U_{ik}A_{kl}V_{lj}\Delta A_{rs}\\ &= \sum_{r,s}\sum_{k,l}U_{ik}\delta_{kr}\delta_{ls}V_{lj}\Delta A_{rs} = \sum_{r,s}U_{ir}\Delta A_{rs}V_{sj}\\ &= (U\,\Delta A\,V)_{ij},\end{aligned}$$

$$\begin{aligned}\left(\frac{\mathrm{d}}{\mathrm{d}A}A^T\Delta A\right)_{ij} &= \sum_{r,s}\frac{\mathrm{d}}{\mathrm{d}A_{rs}}A^T_{ij}\Delta A_{rs} = \sum_{r,s}\frac{\mathrm{d}}{\mathrm{d}A_{rs}}A_{ji}\Delta A_{rs}\\ &= \sum_{r,s}\delta_{jr}\delta_{is}\Delta A_{rs} = \Delta A_{ji}\\ &= \Delta A^T_{ij}.\end{aligned}$$

Hier bezeichne jeweils $\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$ das Kronecker-Delta.

Aus (6.4) und (6.5) können wir mit der Produkt- und Kettenregel (6.6) folgern:

$$\frac{\mathrm{d}}{\mathrm{d}A}(A^TA)\Delta A = \left(\frac{\mathrm{d}}{\mathrm{d}A}A^T\Delta A\right)A + A^T\frac{\mathrm{d}}{\mathrm{d}A}A\Delta A = \Delta A^TA + A^T\Delta A.$$

Sei schließlich

$$f(X, A) := AX - I$$

mit der Einheitsmatrix I. Dann ist

$$\frac{\partial f}{\partial X}\Delta X = A\Delta X, \quad \frac{\partial f}{\partial A}\Delta A = \Delta AX.$$

Weil nach Voraussetzung $\frac{\partial f}{\partial X} = A$ regulär ist, läßt sich $f(X, A) = 0$ nach dem Satz für implizite Funktionen nach X auflösen:

$$X = X(A) = A^{-1}, \quad \Delta X = \frac{\mathrm{d}X}{\mathrm{d}A}\Delta A = \frac{\mathrm{d}A^{-1}}{\mathrm{d}A}\Delta A,$$

und es gilt:

$$0 = \frac{\partial f}{\partial X}\Delta X + \frac{\partial f}{\partial A}\Delta A = A\Delta X + \Delta AX = A\frac{\mathrm{d}A^{-1}}{\mathrm{d}A}\Delta A + \Delta AA^{-1}.$$

Daraus folgt:

$$\frac{\mathrm{d}A^{-1}}{\mathrm{d}A}\Delta A = -A^{-1}\Delta AA^{-1}.$$

■

Lemma 6.2.2 *Sei $W = diag(w_i,\ i = 1, \ldots, n)$, $w = (w_i)_{i=1,\ldots,n}$. Dann ist*

$$\frac{dW}{dw}\,\Delta w = diag(\Delta w_i,\ i = 1, \ldots, n).$$

Beweis. Aus

$$\frac{\mathrm{d}W_{ij}}{\mathrm{d}w}\,\Delta w = \sum_k \frac{\mathrm{d}W_{ij}}{\mathrm{d}w_k}\Delta w_k = \sum_k \delta_{ij}\delta_{jk}\Delta w_k = \delta_{ik}\Delta w_k$$

folgt die Behauptung

$$\frac{\mathrm{d}W}{\mathrm{d}w}\,\Delta w = \mathrm{diag}(\Delta w_i,\ i = 1, \ldots, n).$$

■

6.2.2 Ableitungen von Funktionen auf Matrizen

Die Gütekriterien der Versuchsplanung sind Funktionen auf der Kovarianzmatrix, und zwar Spur, Determinante und Eigenwert. Formeln zur Berechnung dieser Ableitungen werden in den nächsten drei Lemmata bewiesen.

Lemma 6.2.3 *Für die Ableitung der Spur einer Matrix $A \in \mathbb{R}^{n\times n}$ gilt:*

$$\frac{d}{dA}(Spur A)\,\Delta A = Spur\,\Delta A.$$

Beweis.

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}A}(\mathrm{Spur}A)\,\Delta A &= \sum_{i,j}\frac{\mathrm{d}\,\mathrm{Spur}A}{\mathrm{d}A_{ij}}\,\Delta A_{ij} = \sum_{i,j,k}\frac{\mathrm{d}A_{kk}}{\mathrm{d}A_{ij}}\,\Delta A_{ij}\\ &= \sum_{i,j,k}\delta_{ki}\,\delta_{kj}\,\Delta A_{ij} = \sum_k \Delta A_{kk} = \mathrm{Spur}\,\Delta A.\end{aligned}$$

■

Lemma 6.2.4 *Für die Ableitung der Determinante einer symmetrischen, positiv definiten Matrix $A \in \mathbb{R}^{n\times n}$ gilt:*

$$\frac{d}{dA}(\det A)\,\Delta A = \sum_{i,j}\det A\,(A^{-1})_{ij}\,\Delta A_{ij}.$$

Beweis. Sei A'_{ij} die Matrix, die aus A durch Streichen der i-ten Zeile und der j-ten Spalte entsteht, und $\tilde{A}$ die zu A komplementäre Matrix $\tilde{A}_{ij} = (-1)^{i+j}\det A'_{ji}$. Nach dem Laplaceschen Entwicklungssatz gilt für $i = 1, \ldots, n$:

$$\det A = \sum_{j=1}^{n}(-1)^{i+j}A_{ij}\det A'_{ij} = \sum_{j=1}^{n}A_{ij}\tilde{A}_{ji} = (A\,\tilde{A})_{ii}.$$

Somit ist

$$I \det A = A \tilde{A},$$

und weil A positiv definit und damit regulär ist,

$$\tilde{A} = \det A \, A^{-1}.$$

Weiter gilt, da A'_{ij} nicht von A_{ik}, $k = 1, \ldots, n$, abhängt, für $i, k = 1, \ldots, n$:

$$\frac{\mathrm{d}}{\mathrm{d}A_{ik}} \det A = \sum_{j=1}^{n} (-1)^{i+j} \det A'_{ij} \frac{\mathrm{d}}{\mathrm{d}A_{ik}} A_{ij} = \sum_{j=1}^{n} (-1)^{i+j} \det A'_{ij} \delta_{jk} = (-1)^{i+k} \det A'_{ik}$$

bzw.

$$\frac{\mathrm{d}}{\mathrm{d}A} (\det A) \Delta A = \sum_{i,k} (-1)^{i+k} \det A'_{ik} \Delta A_{ik} = \sum_{i,k} \tilde{A}_{ki} \Delta A_{ik} = \sum_{i,k} \det A \, (A^{-1})_{ki} \, \Delta A_{ik}.$$

Aus der Symmetrie von A und A^{-1} folgt:

$$\frac{\mathrm{d}}{\mathrm{d}A} (\det A) \Delta A = \sum_{i,k} \det A \, (A^{-1})_{ik} \, \Delta A_{ik}.$$

■

Lemma 6.2.5 *Sei $\lambda(A)$ einfacher Eigenwert der symmetrischen Matrix $A \in \mathbb{R}^{n \times n}$. Sei z auf 1 normierter Eigenvektor von A zum Eigenwert $\lambda(A)$. Dann gilt:*

$$\frac{d}{dA} (\lambda(A)) \, \Delta A = z^T \, \Delta A \, z.$$

Beweis. Seien

$$\lambda_1 \leq \ldots \leq \lambda_{k-1} < \lambda = \lambda_k < \lambda_{k+1} \leq \ldots \leq \lambda_n$$

die (reellen) Eigenwerte von A und

$$\{z_1, \ldots, z_{k-1}, z = z_k, z_{k+1}, \ldots, z_n\}$$

eine zugehörige Orthonormalbasis von Eigenvektoren. Definiere die orthogonale Matrix

$$U := (z, Z) := (z, z_1, \ldots, z_{k-1}, z_{k+1}, \ldots, z_n) \, .$$

Wir stören nun das Element A_{ij} (und natürlich auch A_{ji}) der Matrix A mit einem $\epsilon \in \mathbb{R}$:

$$A(\epsilon) := A + \epsilon \cdot E_{ij} \qquad \text{mit} \qquad E_{ij} := \frac{1}{2} \left(e_i e_j^T + e_j e_i^T \right) .$$

Sei $\lambda_k(A(\epsilon))$ der k-te Eigenwert der gestörten Matrix. Für kleine ϵ bleibt dieser Eigenwert einfach, und die Reihenfolge der Eigenwerte ändert sich nicht. Definiere

$$\tilde{A}(\epsilon) := A(\epsilon) - \lambda_k(A(\epsilon)) \cdot I \qquad \text{und} \qquad \delta(\epsilon) := \lambda_k(A(\epsilon)) - \lambda_k.$$

Das charakteristische Polynom von $A(\epsilon)$ ist an der Stelle $\lambda_k(A(\epsilon))$ gleich null:

$$0 = \det(\tilde{A}(\epsilon)) = \det(U^T \tilde{A}(\epsilon) U) = \det \begin{pmatrix} z^T \tilde{A}(\epsilon) z & z^T \tilde{A}(\epsilon) Z \\ Z^T \tilde{A}(\epsilon) z & Z^T \tilde{A}(\epsilon) Z \end{pmatrix}$$

Für $\epsilon = 0$ ist $\lambda_k(0) = \lambda_k$ einfacher Eigenwert von $A(0) = A$. Daher ist

$$Z^T \tilde{A}(0) Z = Z^T (A - \lambda_k \cdot I) Z = \operatorname{diag}(\lambda_i - \lambda_k,\ i \neq k)$$

invertierbar. Das gilt auch für kleine Störungen ϵ für $Z^T \tilde{A}(\epsilon) Z$. Aus der Blockzerlegung

$$\left(\begin{array}{c|c} z^T \tilde{A}(\epsilon) z & z^T \tilde{A}(\epsilon) Z \\ \hline Z^T \tilde{A}(\epsilon) z & Z^T \tilde{A}(\epsilon) Z \end{array}\right) = \left(\begin{array}{c|c} z^T \tilde{A}(\epsilon) z - z^T \tilde{A}(\epsilon) Z (Z^T \tilde{A}(\epsilon) Z)^{-1} Z^T \tilde{A}(\epsilon) z & z^T \tilde{A}(\epsilon) Z \\ \hline 0 & Z^T \tilde{A}(\epsilon) Z \end{array}\right) \cdot \left(\begin{array}{c|c} I & 0 \\ \hline (Z^T \tilde{A}(\epsilon) Z)^{-1} Z^T \tilde{A}(\epsilon) z & I \end{array}\right)$$

folgt

$$0 = \det(\tilde{A}(\epsilon)) = \left(z^T \tilde{A}(\epsilon) z - z^T \tilde{A}(\epsilon) Z (Z^T \tilde{A}(\epsilon) Z)^{-1} Z^T \tilde{A}(\epsilon) z \right) \det \left(Z^T \tilde{A}(\epsilon) Z \right)$$

und somit nach Einsetzen der Definition von $\tilde{A}(\epsilon)$:

$$\begin{aligned} 0 &= z^T \tilde{A}(\epsilon) z - z^T \tilde{A}(\epsilon) Z (Z^T \tilde{A}(\epsilon) Z)^{-1} Z^T \tilde{A}(\epsilon) z \\ &= -\delta(\epsilon) + \epsilon \cdot z^T E_{ij} z - \epsilon^2 (z^T E_{ij} Z (Z^T \tilde{A}(\epsilon) Z)^{-1} Z^T E_{ij} z). \end{aligned}$$

Also ist

$$\frac{\delta(\epsilon)}{\epsilon} = z^T E_{ij} z + O(\epsilon)$$

und damit

$$\frac{\mathrm{d}\lambda_k}{\mathrm{d}A_{ij}} = \lim_{\epsilon \to 0} \frac{\delta(\epsilon)}{\epsilon} = z^T E_{ij} z = z_i \cdot z_j.$$

Daraus folgt:

$$\frac{\mathrm{d}}{\mathrm{d}A}(\lambda_k)\, \Delta A = \sum_{i,j} \frac{\mathrm{d}\lambda_k}{\mathrm{d}A_{ij}} \cdot \Delta A_{ij} = \sum_{i,j} z_i \cdot z_j \cdot \Delta A_{ij} = z^T\, \Delta A\, z.$$

■

6.2.3 Ableitungen der Gütekriterien nach der Kovarianzmatrix

Mit diesen Resultaten können wir nun die Ableitungen der Gütekriterien nach der Kovarianzmatrix berechnen.

Satz 6.2.6 *Sei $C \in \mathbb{R}^{n\times n}$ eine symmetrische, positiv semidefinite Matrix. Die in Abschnitt 5.1.4 definierten Gütekriterien auf C haben in der Richtung $\Delta C \in \mathbb{R}^{n\times n}$ die Richtungsableitungen*

$$\begin{aligned}
\frac{d}{dC}(\phi_A(C))\,\Delta C &= \frac{d}{dC}\left(\frac{1}{n}\cdot Spur C\right)\,\Delta C \\
&= \frac{1}{n}\cdot Spur \Delta C \\
\frac{d}{dC}(\phi_D(C))\,\Delta C &= \frac{d}{dC}\left((\det(K^T C K))^{\frac{1}{n}}\right)\,\Delta C \\
&= \frac{1}{n_K}\cdot(\det(K^T C K))^{\frac{1}{n_K}}\cdot\sum_{i,j=1}^{n_K}((K^T C K)^{-1})_{ij}\cdot(K^T\Delta C K)_{ij}.
\end{aligned}$$

Sei der größte Eigenwert von C einfach und z der zugehörige, auf 1 normierte Eigenvektor. Dann ist

$$\begin{aligned}
\frac{d}{dC}(\phi_E(C))\,\Delta C &= \frac{d}{dC}\left(\max\{\lambda : \lambda \text{ Eigenwert von } C\}\right)\,\Delta C \\
&= z^T\,\Delta C\, z.
\end{aligned}$$

Beweis. Dies folgt direkt aus den Lemmata 6.2.3, 6.2.4 und 6.2.5. ∎

Bemerkung 6.2.7 *Das Konfindenzintervall-Kriterium*

$$\phi_M(C) = \max\{\sqrt{C_{ii}},\ i = 1,\ldots,n\}$$

behandeln wir, wie in Abschnitt 5.2.2 beschrieben, durch Umformulierung in Ungleichungsnebenbedingungen

$$q_0 \geq \sqrt{C_{ii}}, \quad i = 1,\ldots,n.$$

Davon lautet die Ableitung

$$\frac{d}{dC}\left(\sqrt{C_{ii}}\right)\,\Delta C = \frac{1}{2\cdot\sqrt{C_{ii}}}\,\Delta C.$$

6.2.4 Ableitung der Kovarianzmatrix nach der Jacobimatrix

Für die Darstellung der Kovarianzmatrix führen wir eine abkürzende Schreibweise ein:

$$\begin{aligned} C &= \begin{pmatrix} I & 0 \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_1^T J_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-T} \begin{pmatrix} I \\ 0 \end{pmatrix} \\ &= I_1 \cdot M^{-1} \cdot S \cdot M^{-1} \cdot I_1^T \end{aligned}$$

mit

$$M := \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}, \quad S := \begin{pmatrix} J_1^T J_1 & 0 \\ 0 & 0 \end{pmatrix}, \quad I_1 := \begin{pmatrix} I & 0 \end{pmatrix}.$$

Zur Berechnung der Ableitung von C nach $J = \begin{pmatrix} J_1 \\ J_2 \end{pmatrix}$ benutzen wir die Formeln aus Lemma 6.2.1. Sei $\Delta J = \begin{pmatrix} \Delta J_1 \\ \Delta J_2 \end{pmatrix}$. Es gilt

$$\begin{aligned} \Delta M &:= \frac{\mathrm{d}M}{\mathrm{d}J}\Delta J = \begin{pmatrix} \Delta J_1^T J_1 + J_1^T \Delta J_1 & \Delta J_2^T \\ \Delta J_2 & 0 \end{pmatrix}, \\ \Delta S &:= \frac{\mathrm{d}S}{\mathrm{d}J}\Delta J = \begin{pmatrix} \Delta J_1^T J_1 + J_1^T \Delta J_1 & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} \Delta J_1^T J_1 + J_1^T \Delta J_1 \\ 0 \end{pmatrix} \cdot I_1. \end{aligned}$$

Die Richtungsableitung von C nach J in der Richtung ΔJ ist dann

$$\begin{aligned} \Delta C &= \frac{\mathrm{d}C}{\mathrm{d}J}\Delta J \\ &= -I_1 \cdot (M^{-1} \cdot \Delta M \cdot M^{-1}) \cdot S \cdot M^{-1} \cdot I_1^T \\ &\quad + I_1 \cdot M^{-1} \cdot \Delta S \cdot M^{-1} \cdot I_1^T \\ &\quad - I_1 \cdot M^{-1} \cdot S \cdot (M^{-1} \cdot \Delta M \cdot M^{-1}) \cdot I_1^T \\ &= I_1 \cdot M^{-1} \cdot \big(-\Delta M \cdot M^{-1} \cdot S + \Delta S - S \cdot M^{-1} \cdot \Delta M\big) \cdot M^{-1} \cdot I_1^T \\ &= I_1 \cdot M^{-1} \cdot \big(-(S \cdot M^{-1} \cdot \Delta M)^T + \Delta S - (S \cdot M^{-1} \cdot \Delta M)\big) \cdot M^{-1} \cdot I_1^T. \end{aligned}$$

Um C und ΔC effizient berechnen zu können, formen wir diese Formeln noch etwas um. Dabei schreiben wir

$$S \cdot M^{-1} \cdot \Delta M = H \cdot I_1 \cdot M^{-1} \cdot \Delta M \quad \text{mit} \quad H := \begin{pmatrix} J_1^T J_1 \\ 0 \end{pmatrix}.$$

Wir erhalten

$$\begin{aligned} C &= I_1 \cdot M^{-1} \cdot S \cdot M^{-1} \cdot I_1^T \\ &= (I_1 \cdot M^{-1}) \cdot ((I_1 \cdot M^{-1}) \cdot S)^T \qquad (6.8) \\ \Delta C &= I_1 \cdot M^{-1} \cdot \big(-(S \cdot M^{-1} \cdot \Delta M)^T + \Delta S - (S \cdot M^{-1} \cdot \Delta M)\big) \cdot M^{-1} \cdot I_1^T \\ &= (I_1 \cdot M^{-1}) \cdot ((I_1 \cdot M^{-1}) \cdot \\ &\quad (-(H \cdot (I_1 \cdot M^{-1}) \cdot \Delta M)^T + \Delta S - (H \cdot (I_1 \cdot M^{-1}) \cdot \Delta M))^T)^T. \qquad (6.9) \end{aligned}$$

Hier kommt wiederholt die (formale) Multiplikation einer Matrix von links mit $(I_1 \cdot M^{-1})$ vor, was im wesentlichen den numerischen Berechnungsaufwand ausmacht. Diese Operation kann effizient durch die Anwendung einer *Nullraummethode* realisiert werden, wie im folgenden beschrieben wird.

6.2.5 Nullraummethode

Algorithmus 6.2.8

Voraussetzung: Seien (CQ) und (PD) erfüllt.

Ziel: Wende $I_1 \cdot M^{-1}$ auf eine Matrix F an.

Eingabe:

$$M = \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}, \qquad F = \begin{pmatrix} F_1 \\ F_2 \end{pmatrix}.$$

Rechnung:

1. *Berechne die QR-Zerlegung von J_2^T:*

$$J_2^T = \begin{pmatrix} Q_1^T & Q_2^T \end{pmatrix} \begin{pmatrix} L^T \\ 0 \end{pmatrix}.$$

2. *Berechne*

$$Y_1 := L^{-1} F_2.$$

3. *Berechne die QR-Zerlegung von $J_1 Q_2^T$:*

$$J_1 Q_2^T = \hat{Q}\hat{R}.$$

4. *Berechne*

$$G := Q_2(F_1 - J_1^T J_1 Q_1^T Y_1)$$

   und löse

$$\hat{R}^T \hat{R} Y_2 = G$$

   nach Y_2.

5. *Berechne*

$$X := \begin{pmatrix} Q_1^T & Q_2^T \end{pmatrix} \begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix}.$$

Ausgabe:

$$X = I_1 \cdot M^{-1} \cdot F.$$

Bemerkung 6.2.9 *Wenn die Nullraummethode mehrmals hintereinander für dieselbe Matrix M ausgeführt wird, müssen die Zerlegungen in den Schritten 1 und 3 natürlich nur beim ersten Mal berechnet werden, die Faktoren können zwischengespeichert und wiederverwendet werden. Dies wird bei der Berechnung der Kovarianzmatrix und ihrer Ableitung gemäß (6.8) und (6.9) ausgenutzt.*

Satz 6.2.10 *Wenn (CQ) und (PD) erfüllt sind, liefert der Algorithmus das korrekte Resultat.*

Beweis. Wegen (CQ) ist $\text{Rang}L = \text{Rang}J_2 = n_2$. Daher existiert Y_1.

Sei $z \in I\!R^{n_1-n_2}$ beliebig. Dann ist $Q_2^T z \in \text{Kern}J_2$: $J_2 Q_2^T z = LQ_1 Q_2^T z = 0$, da $\begin{pmatrix} Q_1^T & Q_2^T \end{pmatrix}$ eine orthogonale Matrix ist. Also ist $Q_2 J_1^T J_1 Q_2^T$ wegen (PD) positiv definit, und $\hat{R}$ hat vollen Rang $n_1 - n_2$. Daher existiert Y_2.

Das Verhalten des Algorithmus ist also wohldefiniert. Zu zeigen bleibt:

$$X \stackrel{!}{=} I_1 \cdot M^{-1} \cdot F \iff \exists Z: \; M \begin{pmatrix} X \\ Z \end{pmatrix} = \begin{pmatrix} J_1^T J_1 X + J_2^T Z \\ J_2 X \end{pmatrix} \stackrel{!}{=} \begin{pmatrix} F_1 \\ F_2 \end{pmatrix}.$$

Man rechnet nach, daß

$$\begin{aligned} J_2 X &= LQ_1 Q_1^T Y_1 + LQ_1 Q_2^T Y_2 = LY_1 + 0 = F_2 \\ Q_2(J_1^T J_1 X + J_2^T Z) &= Q_2 J_1^T J_1 Q_1^T Y_1 + Q_2 J_1^T J_1 Q_2^T Y_2 + Q_2 J_2^T Z \\ &= Q_2 F_1 - G + \hat{R}^T \hat{R} Y_2 + Q_2 Q_1^T L^T Z \\ &= Q_2 F_1 - G + G + 0 = Q_2 F_1 \\ Q_1(J_1^T J_1 X + J_2^T Z) &= Q_1 J_1^T J_1 X + Q_1 J_2^T Z = Q_1 J_1^T J_1 X + L^T Z = Q_1 F_1, \end{aligned}$$

wenn man definiert

$$Z := L^{-T} Q_1 (F_1 - J_1^T J_1 X).$$

Also gilt

$$\begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} (J_1^T J_1 X + J_2^T Z) = \begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix} F_1$$

für die reguläre Matrix $\begin{pmatrix} Q_1 \\ Q_2 \end{pmatrix}$ und damit auch

$$J_1^T J_1 X + J_2^T Z = F_1.$$

■

Bemerkung 6.2.11 *In der Praxis werden die QR-Zerlegungen aus Stabilitätsgründen mit Spalten-Pivotierung durchgeführt. Um die Darstellung übersichtlich zu halten, wird dies hier weggelassen.*

Bemerkung 6.2.12 *Statt der QR-Zerlegung der Matrix $J_1 Q_2^T$ in Schritt 3 kann auch eine Cholesky-Zerlegung der Matrix $Q_2 J_1^T J_1 Q_2^T$ berechnet werden:*

$$Q_2 J_1^T J_1 Q_2^T = \hat{L}\hat{L}^T.$$

In Schritt 4 hat man dann

$$\hat{L}\hat{L}^T Y_2 = G$$

zu lösen.

Bemerkung 6.2.13 *Wir nennen den Algorithmus* Nullraummethode*, weil in den Schritten 1 und 2 zunächst der Lösungsraum (Nullraum) der zweiten Gleichung dargestellt wird. Auf diesem wird dann in den Schritten 3 und 4 die erste Gleichung gelöst.*

6.2.6 Ableitung der Jacobimatrix nach den Versuchsplanungsgrößen

Als nächsten Schritt leiten wir nun die Jacobimatrix des zugrundeliegenden Parameterschätzproblems ab. Gemäß (4.15) hat diese die folgende Darstellung:

$$J = \begin{pmatrix} J_1 \\ J_2 \end{pmatrix} = \begin{pmatrix} -\sqrt{W}\,\Sigma^{-1}\,\frac{\mathrm{d}h}{\mathrm{d}v} \\ \frac{\mathrm{d}d}{\mathrm{d}v} \end{pmatrix}.$$

Zunächst betrachten wir die Ableitungen von J_1 nach den Versuchsplanungsgrößen. J_1 ist eine $M \times n$-Matrix (M: Anzahl der Meßpunkte, n: Dimension von v). Um übersichtlichere Schreibweisen zu bekommen, benutzen wir die folgenden Abkürzungen:

$$h_i := h_i(t_i, x(t_i, p, q, s), p, q), \quad \varsigma_i = \varsigma_i(x(t_i, p, q, s), q), \quad x_i := x(t_i, p, q, s)$$

für alle Meßpunkte $i = 1, \ldots, M$.

Dann gilt:

$$\frac{\mathrm{d}h}{\mathrm{d}v} = \left(\frac{\partial h_i}{\partial x}\frac{\partial x_i}{\partial v} + \frac{\partial h_i}{\partial v} \right)_{i=1,\ldots,M}.$$

Dabei ist

$$\frac{\partial h_i}{\partial v} = \left(\frac{\partial h_i}{\partial p},\ 0 \right).$$

Außerdem ist

$$\sqrt{W}\,\Sigma^{-1} = \mathrm{diag}\left(\frac{\sqrt{w_i}}{\varsigma_i},\ i = 1, \ldots, M \right).$$

Die Richtungsableitungen von J_1 nach den Versuchsplanungsgrößen q, s und w in Richtung $\begin{pmatrix} \Delta q^T & \Delta s^T & \Delta w^T \end{pmatrix}^T$ lauten im einzelnen:

$$\begin{aligned}
\frac{\mathrm{d}J_1}{\mathrm{d}q}\Delta q &= -\sqrt{W}\,\frac{\mathrm{d}}{\mathrm{d}q}\Sigma^{-1}\left(\frac{\partial h_i}{\partial x}\frac{\partial x_i}{\partial v}+\frac{\partial h_i}{\partial v}\right)_{i=1,\ldots,M}\Delta q \\
&= -\sqrt{W}\,\frac{\mathrm{d}}{\mathrm{d}q}\left(\frac{1}{\varsigma_i}\frac{\partial h_i}{\partial x}\frac{\partial x_i}{\partial v}+\frac{1}{\varsigma_i}\frac{\partial h_i}{\partial v}\right)_{i=1,\ldots,M}\Delta q \\
&= -\sqrt{W}\left(\frac{\mathrm{d}}{\mathrm{d}q}\left(\frac{1}{\varsigma_i}\frac{\partial h_i}{\partial x}\frac{\partial x_i}{\partial v}\right)\Delta q+\frac{\mathrm{d}}{\mathrm{d}q}\left(\frac{1}{\varsigma_i}\frac{\partial h_i}{\partial v}\right)\Delta q\right)_{i=1,\ldots,M} \\
&= -\sqrt{W}\left(\left(\frac{1}{\varsigma_i}\frac{\partial h_i}{\partial x}\right)\frac{\partial^2 x_i}{\partial q\partial v}\Delta q\right. \\
&\quad +\frac{\partial}{\partial x}\left(\frac{1}{\varsigma_i}\frac{\partial h_i}{\partial x}\right)\frac{\partial x_i}{\partial q}\Delta q\frac{\partial x_i}{\partial v}+\frac{\partial}{\partial q}\left(\frac{1}{\varsigma_i}\frac{\partial h_i}{\partial x}\right)\Delta q\frac{\partial x_i}{\partial v} \\
&\quad \left.+\frac{\partial}{\partial x}\left(\frac{1}{\varsigma_i}\frac{\partial h_i}{\partial v}\right)\frac{\partial x_i}{\partial q}\Delta q+\frac{\partial}{\partial q}\left(\frac{1}{\varsigma_i}\frac{\partial h_i}{\partial v}\right)\Delta q\right)_{i=1,\ldots,M},
\end{aligned}$$

$$\begin{aligned}
\frac{\mathrm{d}J_1}{\mathrm{d}s}\Delta s &= -\sqrt{W}\,\frac{\mathrm{d}}{\mathrm{d}s}\Sigma^{-1}\left(\frac{\partial h_i}{\partial x}\frac{\partial x_i}{\partial v}+\frac{\partial h_i}{\partial v}\right)_{i=1,\ldots,M}\Delta s \\
&= -\sqrt{W}\,\frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{1}{\varsigma_i}\frac{\partial h_i}{\partial x}\frac{\partial x_i}{\partial v}+\frac{1}{\varsigma_i}\frac{\partial h_i}{\partial v}\right)_{i=1,\ldots,M}\Delta s \\
&= -\sqrt{W}\left(\frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{1}{\varsigma_i}\frac{\partial h_i}{\partial x}\frac{\partial x_i}{\partial v}\right)\Delta s+\frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{1}{\varsigma_i}\frac{\partial h_i}{\partial v}\right)\Delta s\right)_{i=1,\ldots,M} \\
&= -\sqrt{W}\left(\left(\frac{1}{\varsigma_i}\frac{\partial h_i}{\partial x}\right)\frac{\partial^2 x_i}{\partial s\partial v}\Delta s\right. \\
&\quad \left.+\frac{\partial}{\partial x}\left(\frac{1}{\varsigma_i}\frac{\partial h_i}{\partial x}\right)\frac{\partial x_i}{\partial s}\Delta s\frac{\partial x_i}{\partial v}+\frac{\partial}{\partial x}\left(\frac{1}{\varsigma_i}\frac{\partial h_i}{\partial v}\right)\frac{\partial x_i}{\partial s}\Delta s\right)_{i=1,\ldots,M}
\end{aligned}$$

und

$$\frac{\mathrm{d}J_1}{\mathrm{d}w}\Delta w = -\mathrm{diag}\left(\frac{\mathrm{d}}{\mathrm{d}w_i}\frac{\sqrt{w_i}}{\varsigma_i}\,\Delta w_i\right)\frac{\mathrm{d}h}{\mathrm{d}v} = -\frac{1}{2}\cdot\mathrm{diag}\left(\frac{\Delta w_i}{\sqrt{w_i}\cdot\varsigma_i}\right)\frac{\mathrm{d}h}{\mathrm{d}v}.$$

Statt $\frac{\mathrm{d}J_1}{\mathrm{d}w}\Delta w$ benötigen wir eigentlich nur $\frac{\mathrm{d}J_1^T J_1}{\mathrm{d}w}\Delta w$, siehe Abschnitt 6.2.4. Dann hebt sich auch die Singularität bei $w_i = 0$ weg:

$$\frac{\mathrm{d}J_1^T J_1}{\mathrm{d}w}\Delta w = \left(\frac{\mathrm{d}J_1}{\mathrm{d}w}\Delta w\right)^T J_1 + J_1^T\left(\frac{\mathrm{d}J_1}{\mathrm{d}w}\Delta w\right) = \frac{\mathrm{d}h}{\mathrm{d}v}^T\mathrm{diag}\left(\frac{\Delta w_i}{\varsigma_i^2}\right)\frac{\mathrm{d}h}{\mathrm{d}v}.$$

Die Ableitungen von

$$J_2 = \frac{\mathrm{d}d}{\mathrm{d}v} = \frac{\partial d}{\partial x}\frac{\partial x}{\partial v}+\frac{\partial d}{\partial v}$$

mit

$$\frac{\partial d}{\partial v} = \left(\frac{\partial d}{\partial p},\ \frac{\partial d}{\partial s}\right)$$

ergeben sich analog:

$$\begin{aligned}
\frac{\mathrm{d}J_2}{\mathrm{d}q}\Delta q &= \frac{\mathrm{d}}{\mathrm{d}q}\left(\frac{\partial d}{\partial x}\frac{\partial x}{\partial v}+\frac{\partial d}{\partial v}\right)\Delta q \\
&= \frac{\partial d}{\partial x}\frac{\partial^2 x}{\partial q \partial v}\Delta q + \frac{\partial^2 d}{\partial x \partial x}\frac{\partial x}{\partial q}\Delta q\frac{\partial x}{\partial v} + \frac{\partial^2 d}{\partial q \partial x}\Delta q\frac{\partial x}{\partial v} \\
&\quad + \frac{\partial^2 d}{\partial x \partial v}\frac{\partial x}{\partial q}\Delta q + \frac{\partial^2 d}{\partial q \partial v}\Delta q, \\
\frac{\mathrm{d}J_2}{\mathrm{d}s}\Delta s &= \frac{\mathrm{d}}{\mathrm{d}s}\left(\frac{\partial d}{\partial x}\frac{\partial x}{\partial v}+\frac{\partial d}{\partial v}\right)\Delta s \\
&= \frac{\partial d}{\partial x}\frac{\partial^2 x}{\partial s \partial v}\Delta s + \frac{\partial^2 d}{\partial x \partial x}\frac{\partial x}{\partial s}\Delta s\frac{\partial x}{\partial v} + \frac{\partial^2 d}{\partial s \partial x}\Delta s\frac{\partial x}{\partial v} \\
&\quad + \frac{\partial^2 d}{\partial x \partial v}\frac{\partial x}{\partial s}\Delta s + \frac{\partial^2 d}{\partial s \partial v}\Delta s, \\
\frac{\mathrm{d}J_2}{\mathrm{d}w}\Delta w &= 0.
\end{aligned}$$

6.2.7 Ableitung der Lösung des DAE-Systems nach Versuchsplanungsgrößen und Parametern

Aus den Resultaten des vorigen Abschnitts liest man ab, daß das Integrationsverfahren neben der Lösung $x = (y, z)$ (*Nominaltrajektorie*) des DAE-Systems

$$A(t, y(t), z(t), p, q)\ \dot{y}(t) = f(t, y(t), z(t), p, q) \tag{6.10}$$

$$0 = g(t, y(t), z(t), p, q), \tag{6.11}$$

auch die folgenden Ableitungen (*Variationstrajektorien*) bereitstellen muß:

$$\frac{\partial x}{\partial v}, \quad \frac{\partial x}{\partial q}, \quad \frac{\partial x}{\partial s}, \quad \frac{\partial^2 x}{\partial q \partial v}, \quad \frac{\partial^2 x}{\partial s \partial v},$$

ausgewertet an den Stellen (t_i, p, q, s), $i = 1, \ldots, M$.

Variations-DAE

Wir betrachten zunächst exemplarisch $\frac{\partial x}{\partial q} =: x^q = (y^q, z^q)$. Es ist Lösung einer sogenannten *Variations-DAE*, die man durch Ableiten von (6.10)-(6.11) nach q erhält:

$$\begin{aligned}
\left(\frac{\partial A}{\partial x}x^q + \frac{\partial A}{\partial q}\right)\dot{y} + A\dot{y}^q &= \frac{\partial f}{\partial x}x^q + \frac{\partial f}{\partial q} \\
0 &= \frac{\partial g}{\partial x}x^q + \frac{\partial g}{\partial q}
\end{aligned}$$

bzw.

$$A\dot{y}^q = \frac{\partial f}{\partial x}x^q + \frac{\partial f}{\partial q} - \left(\frac{\partial A}{\partial x}x^q + \frac{\partial A}{\partial q}\right)\dot{y} \tag{6.12}$$

$$0 = \frac{\partial g}{\partial x}x^q + \frac{\partial g}{\partial q}. \tag{6.13}$$

Bemerkung 6.2.14 *Die Variations-DAE ist linear.*

BDF-Diskretisierungsschema für die DAE

Mit $x_{n+1} = (y_{n+1}, z_{n+1})$ bezeichnen wir wie in Abschnitt 3.6 die Approximation der Lösung $x(t_{n+1}) = (y(t_{n+1}), z(t_{n+1}))$ der DAE (6.10)-(6.11) durch ein BDF-Diskretisierungsschema mit Ordnung k und Schrittweite h. Dann ist

$$\dot{y}_{n+1} = \frac{1}{h}\cdot\left(\alpha_0\cdot y_{n+1} + \sum_{i=1}^{k}\alpha_i\cdot y_{n+1-i}\right) =: \frac{1}{h}\cdot\left(\alpha_0\cdot y_{n+1} + \beta\right). \tag{6.14}$$

Wir definieren die folgenden Abkürzungen:

$$\begin{aligned} A_{n+1} &:= A(t_{n+1}, y_{n+1}, z_{n+1}, p, q),\\ f_{n+1} &:= f(t_{n+1}, y_{n+1}, z_{n+1}, p, q),\\ g_{n+1} &:= g(t_{n+1}, y_{n+1}, z_{n+1}, p, q). \end{aligned}$$

BDF-Diskretisierungsschema für die Variations-DAE

Bezeichne nun

$$x^q_{n+1} = \left(y^q_{n+1}, z^q_{n+1}\right)$$

die Approximation der Lösung

$$x^q(t_{n+1}) = (y^q(t_{n+1}), z^q(t_{n+1}))$$

der Variations-DAE (6.12)-(6.13).

Das BDF-Diskretisierungsschema für die Variations-DAE mit Ordnung k und Schrittweite h^q lautet

$$\dot{y}^q_{n+1} = \frac{1}{h^q}\cdot\left(\alpha^q_0\cdot y^q_{n+1} + \sum_{i=1}^{k}\alpha^q_i\cdot y^q_{n+1-i}\right) =: \frac{1}{h^q}\cdot\left(\alpha^q_0\cdot y^q_{n+1} + \beta^q\right). \tag{6.15}$$

Dies setzen wir in die Variations-DAE ein:

$$\begin{aligned} A_{n+1}\dot{y}^q_{n+1} - \frac{\partial f_{n+1}}{\partial x}x^q_{n+1} + \frac{\partial f_{n+1}}{\partial q} + \left(\frac{\partial A_{n+1}}{\partial x}x^q_{n+1} + \frac{\partial A_{n+1}}{\partial q}\right)\dot{y}_{n+1} &= 0\\ \frac{\partial g_{n+1}}{\partial x}x^q_{n+1} + \frac{\partial g_{n+1}}{\partial q} &= 0. \end{aligned}$$

Nehmen wir nun die gleiche Wahl der Ordnung und Schrittweite $h^q = h$ wie zur Berechnung der Nominaltrajektorie vor, so ist $\alpha_i^q = \alpha_i$, $i = 0, \ldots, k$, und wir erhalten nach Einsetzen von (6.14) und (6.15)

$$\begin{aligned} A_{n+1}\left(\alpha_0 \cdot y_{n+1}^q + \sum_{i=1}^{k} \alpha_i \cdot y_{n+1-i}^q\right) & \\ + \left(\frac{\partial A_{n+1}}{\partial x} x_{n+1}^q + \frac{\partial A_{n+1}}{\partial q}\right)(\alpha_0 \cdot y_{n+1} + \beta) & \\ -h \cdot \left(\frac{\partial f_{n+1}}{\partial x} x_{n+1}^q + \frac{\partial f_{n+1}}{\partial q}\right) &= 0 \qquad (6.16) \\ \frac{\partial g_{n+1}}{\partial x} x_{n+1}^q + \frac{\partial g_{n+1}}{\partial q} &= 0. \qquad (6.17) \end{aligned}$$

Ableiten des Diskretisierungsschemas der DAE

Ein anderer Ansatz zur Berechnung von x^q besteht darin, das Diskretisierungsschema für die DAE

$$\begin{aligned} A_{n+1}\left(\alpha_0 \cdot y_{n+1} + \sum_{i=1}^{k} \alpha_i \cdot y_{n+1-i}\right) - h \cdot f_{n+1} &= 0 \\ g_{n+1} &= 0 \end{aligned}$$

nach q abzuleiten:

$$\begin{aligned} A_{n+1}\left(\alpha_0 \cdot y_{n+1}^q + \sum_{i=1}^{k} \alpha_i \cdot y_{n+1-i}^q\right) & \\ + \left(\frac{\partial A_{n+1}}{\partial x} x_{n+1}^q + \frac{\partial A_{n+1}}{\partial q}\right)(\alpha_0 \cdot y_{n+1} + \beta) & \\ -h \cdot \left(\frac{\partial f_{n+1}}{\partial x} x_{n+1}^q + \frac{\partial f_{n+1}}{\partial q}\right) &= 0 \qquad (6.18) \\ \frac{\partial g_{n+1}}{\partial x} x_{n+1}^q + \frac{\partial g_{n+1}}{\partial q} &= 0, \qquad (6.19) \end{aligned}$$

Ordnung und Schrittweite werden dabei festgehalten.

Vergleich der beiden Ansätze

Wir erkennen: Die Gleichungssysteme (6.16)-(6.17) und (6.18)-(6.19) sind identisch. Das heißt, beide Ansätze, Aufstellen des BDF-Schemas für die Variations-DAE und Ableiten des BDF-Schemas für die ursprüngliche DAE, führen zum selben Gleichungssystem. Dies ist das Prinzip der *Internen Numerischen Differentiation* [24, 25], die Vertauschbarkeit von Integrationsschema und Ableitungsoperator [88]. Die durch diese Vorgehensweise gewonnene numerische Lösung der Variations-DAE ist somit die exakte Ableitung der numerischen Lösung der DAE.

Matrix zur Lösung des impliziten Problems

Für System (6.16)-(6.17) bzw. (6.18)-(6.19) als Gleichungssystem für x^q_{n+1} schreiben wir kurz:

$$\mathcal{F}^q(x^q_{n+1}) = 0.$$

Die Matrix dieses (linearen) Gleichungssystems lautet:

$$\frac{\partial \mathcal{F}^q}{\partial x^q_{n+1}} = \left(\begin{array}{c|c} \alpha_0 A_{n+1} + \frac{\partial A_{n+1}}{\partial y}\left(\alpha_0 y_{n+1} + \beta\right) - h\frac{\partial f_{n+1}}{\partial y} & \frac{\partial A_{n+1}}{\partial z}\left(\alpha_0 y_{n+1} + \beta\right) - h\frac{\partial f_{n+1}}{\partial z} \\ \hline \frac{\partial g_{n+1}}{\partial y} & \frac{\partial g_{n+1}}{\partial z} \end{array} \right).$$

Dies ist dieselbe Matrix wie zur Berechnung der Nominaltrajektorie, siehe (3.23). Das heißt, die zusätzliche Berechnung von Ableitungen bedeutet im wesentlichen keinen Mehraufwand bei der Behandlung der Linearen Algebra.

Andere erste und zweite Ableitungen

Die Ableitungen von x nach p und s berechnen wir analog.

$$x^p := \frac{\partial x}{\partial p}, \quad x^s := \frac{\partial x}{\partial s}.$$

Außerdem benötigen wir auch gemischte zweite Ableitungen nach q und p:

$$\frac{\partial^2 x}{\partial q \partial p} =: x^{qp} = \left(y^{qp}, z^{qp}\right).$$

Zu deren Berechnung leiten wir die Variations-DAE (6.12)-(6.13) zusätzlich auch nach p ab und erhalten die folgende Variations-DAE zweiter Ordnung:

$$\begin{aligned} \left(\frac{\partial A}{\partial x}x^p + \frac{\partial A}{\partial p}\right)\dot{y}^q + A\dot{y}^{qp} &= \left(\frac{\partial^2 f}{\partial x \partial x}x^p + \frac{\partial^2 f}{\partial p \partial x}\right)x^q + \frac{\partial f}{\partial x}x^{qp} + \frac{\partial^2 f}{\partial x \partial q}x^p + \frac{\partial^2 f}{\partial p \partial q} \\ &- \left(\left(\frac{\partial^2 A}{\partial x \partial x}x^p + \frac{\partial^2 A}{\partial p \partial x}\right)x^q + \frac{\partial A}{\partial x}x^{qp} + \frac{\partial^2 A}{\partial x \partial q}x^p + \frac{\partial^2 A}{\partial p \partial q}\right)\dot{y} \\ &- \left(\frac{\partial A}{\partial x}x^q + \frac{\partial A}{\partial q}\right)\dot{y}^p \\ 0 &= \left(\frac{\partial^2 g}{\partial x \partial x}x^p + \frac{\partial^2 g}{\partial p \partial x}\right)x^q + \frac{\partial g}{\partial x}x^{qp} + \frac{\partial^2 g}{\partial x \partial q}x^p + \frac{\partial^2 g}{\partial p \partial q}. \end{aligned}$$

Die Diskretisierungsschemata für Nominaltrajektorie x und erste Ableitungen x^q und x^p liefern die Darstellungen

$$\dot{y}_{n+1} = \frac{1}{h}\left(\alpha_0 \cdot y_{n+1} + \beta\right), \quad \dot{y}^q_{n+1} = \frac{1}{h}\left(\alpha_0 \cdot y^q_{n+1} + \beta^q\right), \quad \dot{y}^p_{n+1} = \frac{1}{h}\left(\alpha_0 \cdot y^p_{n+1} + \beta^p\right).$$

Da jeweils die gleiche Schrittweiten- und Ordnungssteuerung gewählt wird, ist der Koeffizient α_0 gleich.

Wir approximieren $x^{qp}(t_{n+1})$ nun durch

$$x_{n+1}^{qp} = \left(y_{n+1}^{qp}, z_{n+1}^{qp}\right).$$

und wenden dafür ein BDF-Schema an, ebenfalls mit der gleichen Schrittweiten- und Ordnungssteuerung wie für Nominaltrajektorie und Variationstrajektorien erster Ordnung:

$$\dot{y}_{n+1}^{qp} = \frac{1}{h}\left(\alpha_0 \cdot y_{n+1}^{qp} + \sum_{i=1}^{k+1} \alpha_i \cdot y_{n+1-i}^{qp}\right) =: \frac{1}{h}\left(\alpha_0 \cdot y_{n+1}^{qp} + \beta^{qp}\right).$$

Eingesetzt in die Variations-DAE zweiter Ordnung, ergibt sich das Gleichungssystem

$$\begin{aligned}
0 &= A_{n+1}\left(\alpha_0 \cdot y_{n+1}^{qp} + \sum_{i=1}^{k+1} \alpha_i \cdot y_{n+1-i}^{qp}\right) \\
&\quad -h \cdot \left(\frac{\partial^2 f_{n+1}}{\partial x \partial x} x_{n+1}^p x_{n+1}^q + \frac{\partial^2 f_{n+1}}{\partial p \partial x} x_{n+1}^q + \frac{\partial^2 f_{n+1}}{\partial x \partial q} x_{n+1}^p + \frac{\partial^2 f_{n+1}}{\partial p \partial q} + \frac{\partial f_{n+1}}{\partial x} x_{n+1}^{qp}\right. \\
&\quad \left. + \left(\frac{\partial^2 A_{n+1}}{\partial x \partial x} x_{n+1}^p x_{n+1}^q + \frac{\partial^2 A_{n+1}}{\partial p \partial x} x_{n+1}^q + \frac{\partial^2 A_{n+1}}{\partial x \partial q} x_{n+1}^p + \frac{\partial^2 A_{n+1}}{\partial p \partial q} + \frac{\partial A_{n+1}}{\partial x} x_{n+1}^{qp}\right) \dot{y}_{n+1}\right. \\
&\quad \left. + \left(\frac{\partial A_{n+1}}{\partial x} x_{n+1}^p + \frac{\partial A_{n+1}}{\partial p}\right) \dot{y}_{n+1}^q + \left(\frac{\partial A_{n+1}}{\partial x} x_{n+1}^q + \frac{\partial A_{n+1}}{\partial q}\right) \dot{y}_{n+1}^p\right) \\
0 &= \frac{\partial^2 g_{n+1}}{\partial x \partial x} x_{n+1}^p x_{n+1}^q + \frac{\partial^2 g_{n+1}}{\partial p \partial x} x_{n+1}^q + \frac{\partial^2 g_{n+1}}{\partial x \partial q} x_{n+1}^p + \frac{\partial^2 g_{n+1}}{\partial p \partial q} + \frac{\partial g_{n+1}}{\partial x} x_{n+1}^{qp}
\end{aligned}$$

oder kurz

$$\mathcal{F}^{qp}(x_{n+1}^q) = 0$$

mit der Matrix

$$\frac{\partial \mathcal{F}^{qp}}{\partial x_{n+1}^{qp}} = \left(\begin{array}{c|c} \alpha_0 A_{n+1} + \frac{\partial A_{n+1}}{\partial y}\left(\alpha_0 y_{n+1} + \beta\right) - h\frac{\partial f_{n+1}}{\partial y} & \frac{\partial A_{n+1}}{\partial z}\left(\alpha_0 y_{n+1} + \beta\right) - h\frac{\partial f_{n+1}}{\partial z} \\ \hline \frac{\partial g_{n+1}}{\partial y} & \frac{\partial g_{n+1}}{\partial z} \end{array}\right).$$

Dies ist wieder dieselbe Systemmatrix wie bei Nominaltrajektorie und ersten Ableitungen, was induktiv klar ist.

6.3 Ableitungen der Modellfunktionen

Folgende Ableitungen der benutzerdefinierten Funktionen werden zur Berechnung der Lösungen von Nominal- und Variationstrajektorien benötigt:

- Ableitungen der DAE-Modellfunktionen

$$\frac{\partial f}{\partial x}, \frac{\partial f}{\partial p}, \frac{\partial f}{\partial q}, \frac{\partial^2 f}{\partial x \partial x}, \frac{\partial^2 f}{\partial p \partial x}, \frac{\partial^2 f}{\partial x \partial q}, \frac{\partial^2 f}{\partial p \partial q},$$

$$\frac{\partial g}{\partial x}, \frac{\partial g}{\partial p}, \frac{\partial g}{\partial q}, \frac{\partial^2 g}{\partial x \partial x}, \frac{\partial^2 g}{\partial p \partial x}, \frac{\partial^2 g}{\partial x \partial q}, \frac{\partial^2 g}{\partial p \partial q},$$

$$\frac{\partial A}{\partial x}, \frac{\partial A}{\partial p}, \frac{\partial A}{\partial q}, \frac{\partial^2 A}{\partial x \partial x}, \frac{\partial^2 A}{\partial p \partial x}, \frac{\partial^2 A}{\partial x \partial q}, \frac{\partial^2 A}{\partial p \partial q}.$$

- Ableitungen der Meßfunktionen, $i = 1, \ldots, M$

$$\frac{\partial h_i}{\partial x}, \frac{\partial h_i}{\partial p}, \frac{\partial}{\partial x}\left(\frac{1}{\varsigma_i}\frac{\partial h_i}{\partial x}\right), \frac{\partial}{\partial x}\left(\frac{1}{\varsigma_i}\frac{\partial h_i}{\partial p}\right), \frac{\partial}{\partial q}\left(\frac{1}{\varsigma_i}\frac{\partial h_i}{\partial x}\right), \frac{\partial}{\partial q}\left(\frac{1}{\varsigma_i}\frac{\partial h_i}{\partial p}\right).$$

- Ableitungen der Nebenbedingungen des Parameterschätzproblems

$$\frac{\partial d}{\partial x}, \frac{\partial d}{\partial p}, \frac{\partial d}{\partial s}, \frac{\partial^2 d}{\partial x \partial x}, \frac{\partial^2 d}{\partial q \partial x}, \frac{\partial^2 d}{\partial s \partial x}, \frac{\partial^2 d}{\partial x \partial p}, \frac{\partial^2 d}{\partial q \partial p}, \frac{\partial^2 d}{\partial s \partial p}, \frac{\partial^2 d}{\partial x \partial s}, \frac{\partial^2 d}{\partial q \partial s}, \frac{\partial^2 d}{\partial s \partial s}.$$

- Ableitungen der Zustandsbeschränkungen

$$\frac{\partial b}{\partial x}, \frac{\partial b}{\partial q}.$$

- Ableitungen der Steuerbeschränkungen

$$\frac{\partial c}{\partial q}.$$

Die Erzeugung dieser Ableitungen bewerkstelligen wir durch Automatische Differentiation mit Hilfe des Programms ADIFOR [22]. Der folgende Abschnitt gibt zum Abschluß dieses Kapitels einen kurzen Überblick über diese Technik.

6.4 Automatische Differentiation

Das Prinzip der Automatischen Differentiation besteht in der Ableitung der einzelnen Schritte der Berechnungsvorschrift einer Funktion und im Zusammensetzen dieser Ableitungen nach der Kettenregel. Damit erhält man eine exakte Berechnungsvorschrift für die Ableitung der Funktion. Diese kann zusammen mit dem Funktionscode effizient ausgewertet werden.

Einen Überblick über die Theorie der Automatischen Differentiation gibt das Buch von Griewank [58]. Spezielle Softwarepakete werden in [59], [21], [23], [22] und [100] vorgestellt.

Als Alternativen zur Automatischen Differentiation können für einfache Funktionsauswertungen

- die Ableitungen *per Hand* bereitgestellt werden, dies ist jedoch aufwendig und fehleranfällig. Oder man setzt
- Computer-Algebra-Software (Maple, Mathematica,...) zur *Symbolischen Differentiation* ein. Dabei werden aus einer Zeichenkette, die die Funktion beschreibt, Zeichenketten für die Ableitungen generiert. Dies kann bei komplexen Funktionen zu sehr hohem Ressourcenverbrauch führen und ist daher nicht effizient in der numerischen Praxis. Weil Funktion und Ableitung getrennt dargestellt werden, ist die gemeinsame Auswertung von Teilausdrücken nicht möglich. Außerdem können keine Schleifen und Verzweigungen behandelt werden.

Auch für komplexe Algorithmen geeignet ist die

- *Numerische Differentiation* mittels Differenzenquotienten [57], zum Beipiel einseitig, $\frac{\mathrm{d}f}{\mathrm{d}x} \approx \frac{f(x+h)-f(x)}{h}$, oder zentral, $\frac{\mathrm{d}f}{\mathrm{d}x} \approx \frac{f(x+h)-f(x-h)}{2 \cdot h}$. Die Funktion f kann dabei als Black-Box aufgefaßt werden. Bei der Wahl der Störung h ist zu beachten, daß kleine Störungen numerische Auslöschung, große Störungen schlechte Approximation bewirken. So erhält man durch Verfahrensfehler nur Näherungen der Ableitungen.

Im folgenden wollen wir die Vorgehensweise der Automatischen Differentiation genauer darstellen. Wir betrachten eine differenzierbare Abbildung

$$f: \ D \subseteq I\!R^n \to I\!R^m: \ x \mapsto y = f(x).$$

Die x nennen wir *unabhängige Variablen* und die y *abhängige Variablen*.

Sei f *faktorisierbar*, das heißt der Algorithmus zur Berechnung von f bestehe aus einer (eventuell langen) Sequenz von *elementaren Operationen* (Additionen, Multiplikationen, usw. und elementare Funktionsaufrufe exp(.), sin(.), $\sqrt{.}$, usw.). Deren Reihenfolge kann dabei auch durch Schleifen und Verzweigungen bestimmt werden. Diese Sequenz nennt man den *Computational Trace*, den man schematisch folgendermaßen aufschreiben kann:

Initialisierung:	v_{i-n}	:=	x_i	$i = 1, \ldots, n$
Berechnungen:	v_i	:=	$\varphi_i(v_{j \prec i})$	$i = 1, \ldots, l$
Ergebnis:	y_{m-i}	:=	v_{l-i}	$i = m-1, \ldots, 0$

$v_{j \prec i}$ beschreibt dabei das Tupel aller Argumente v_j, von denen die elementare Operation φ_i abhängt, $j < i$.

Durch Ableiten dieser elementaren Schritte und wiederholte Anwendung der Kettenregel kann die Ableitung von f auf völlig mechanische Art und Weise berechnet werden. Die Auswertung der Ableitung ist dabei eng verknüpft mit der Funktionsauswertung. Wir unterscheiden zwei Möglichkeiten der Reihenfolge, in der man dabei vorgehen kann:

1. Beim *Vorwärtsmodus* werden Ableitungen der Zwischenergebnisse nach den unabhängigen Variablen x berechnet. Typischerweise wird dieser Modus für Richtungsableitungen eingesetzt:

$$\bar{y} = \frac{\mathrm{d}f}{\mathrm{d}x}\bar{x}$$

mit einem Richtungsvektor $\bar{x} \in \mathbb{R}^n$. Der schematische Ablauf ist der folgende:

Initialisierung:	v_{i-n}	$:=$	x_i	$i = 1, \ldots, n$
	$\bar{v}_{i-n}$	$:=$	$\bar{x}_i$	
Berechnungen:	v_i	$:=$	$\varphi_i(v_{j \prec i})$	$i = 1, \ldots, l$
	$\bar{v}_i$	$:=$	$\sum_{j: j \prec i} \frac{\partial \varphi_i}{\partial v_j} \, \bar{v}_j$	
Ergebnis:	y_{m-i}	$:=$	v_{l-i}	$i = m-1, \ldots, 0$
	$\bar{y}_{m-i}$	$:=$	$\bar{v}_{l-i}$	

2. Beim *Rückwärtsmodus* werden umgekehrt Ableitungen des Endergebnisses y nach den Zwischenergebnissen berechnet. Dies verwendet man für Linearkominationen

$$\underline{x} = \underline{y} \frac{\mathrm{d}f}{\mathrm{d}x}$$

mit $\underline{y} \in \mathbb{R}^m$. Bei der Ableitungsberechnung ist beim Rückwärtsmodus der Zugriff auf die Operationen umgekehrt wie bei der Funktionsauswertung. Deshalb macht man zunächst einen Vorwärtslauf

Initialisierung:	v_{i-n}	$:=$	x_i	$i = 1, \ldots, n$
Berechnungen:	v_i	$:=$	$\varphi_i(v_{j \prec i})$	$i = 1, \ldots, l$
Ergebnis:	y_{m-i}	$:=$	v_{l-i}	$i = m-1, \ldots, 0$

und speichert die Zwischenergebnisse. Im Rückwärtslauf (hier: inkrementelle Variante) werden dann die Ableitungen berechnet:

Initialisierung:	$\underline{v}_{l-i}$	$:=$	$\underline{y}_{m-i}$	$i = 0, \ldots, m-1$
	$\underline{v}_i$	$:=$	0	$i = 1-n, \ldots, l-m$
Berechnungen:	$\underline{v}_j$	$:=$	$\underline{v}_j + \underline{v}_i \frac{\partial \varphi_i}{\partial v_j}$	für alle $j \prec i$, $\quad i = l, \ldots, 1$
Ergebnis:	$\underline{x}_i$	$:=$	v_{i-n}	$i = n, \ldots, 1$

Bemerkung 6.4.1 (Genauigkeit der Automatischen Differentiation)
Da jede Ableitung der elementaren Operationen analytisch berechnet wird, ist die durch Automatische Differentiation berechnete Ableitung von f exakt. Es tritt kein Verfahrensfehler auf. Rundungsfehler beim numerischen Auswerten können — wie bei der Funktionsauswertung — natürlich vorkommen.

Bemerkung 6.4.2 (Komplexität der Automatischen Differentiation)
Komplexitätsuntersuchungen für Vorwärts- und Rückwärtsmodus der Automatischen Differentiation, siehe zum Beispiel [58], liefern die folgenden Resultate:

- *Die Gesamtkomplexität des Algorithmus zur Berechnung von f ist die Summe der Komplexitäten seiner elementaren Operationen.*

- *Die Auswertung der Ableitung einer beliebigen elementaren Operation ist nur um ein konstantes Vielfaches Q teurer als die Auswertung der elementaren Operation selbst.*
$$\textit{Division: } \left(\frac{a}{b}\right)' = \frac{a' \cdot b - a \cdot b'}{b^2} \quad \rightarrow \textit{Faktor 5.}$$

- *Zur gemeinsamen Berechnung von f und $\frac{df}{dx}\bar{x}$ oder $\underline{y}\frac{df}{dx}$ mit Automatischer Differentiation ist daher nur ein Q-facher Aufwand wie zur Berechnung von f allein nötig.*

- *Wertet man den Vorwärtsmodus mit d statt einer Richtung aus, so ist der Aufwand d mal so hoch, der Aufwand ist proportional zur Anzahl der Richtungen.*

- *Beim Rückwärtsmodus mit d statt einer Linearkombination ist der Aufwand proportional zur Anzahl der Linearkombinationen.*

- *Zur Berechnung der gesamten Jacobimatrix $\frac{df}{dx}$ braucht man im Vorwärtsmodus n Richtungen $e_1, \ldots, e_n$ und erhält den $Q \cdot n$-fachen Aufwand wie für die Funktionsauswertung allein. Im Rückwärtsmodus benötigt man m Linearkombinationen $e_1^T, \ldots, e_m^T$, und der Aufwand ist $Q \cdot m$ mal so hoch. Zur Berechnung der Jacobimatrix ist der Vorwärtsmodus also besser geeignet bei mehr abhängigen als unabhängigen Variablen, der Rückwärtsmodus bei mehr unabhängigen als abhängigen Variablen.*

Bemerkung 6.4.3 (Speicherbedarf der Automatischen Differentiation)
Beim Rückwärtsmodus ist zusätzlicher Speicheraufwand zum Zwischenspeichern des Vorwärtslaufs erforderlich, während beim Vorwärtsmodus dieser Overhead nicht notwendig ist.

Zur Realisierung der Automatischen Differentiation gibt es verschiedene Möglichkeiten:

- *Überladen* der elementaren Operationen mit zusätzlicher Berechnung der Ableitungen in objektorientierten Programmiersprachen. Nach diesem Prinzip arbeitet zum Beispiel die Software ADOL-C [59].

- Durch *Quelltext-Transformation* wird aus dem Programmcode zur Berechnung von f Programmcode für die Ableitungen erzeugt. Die bekannteste Software, die nach diesem Konzept arbeitet, ist ADIFOR [21, 23, 22]. Dieses Programm arbeitet mit Fortran-77-Subroutinen. Es ist einfach zu bedienen und erzeugt effizienten und portablen Code. Bei der Quelltext-Transformation wird generell der Vorwärtsmodus verwendet, nur für Zuweisungsanweisungen der Rückwärtsmodus. Dadurch ist der Ableitungscode besonders für Richtungsableitungen effizient, und es ist kein Zwischenspeichern der gesamten Funktionsauswertung für den Rückwärtsmodus notwendig.

Wir entschieden uns aufgrund der Reife der Software und der Effizienz für ADIFOR. Dieses setzen wir ein, um aus Fortran-Funktionscode erste Ableitungen zu generieren.

Zweite Ableitungen werden als erste Ableitungen von ersten Ableitungen erzeugt. Die Aufrufe von ADIFOR sind in unserer Software vollständig automatisiert, für genauere Ausführungen zur Implementation siehe Abschnitt 9.4 in Kapitel 9.

Alternativ zur Verwendung von ADIFOR wurde von Rücker [100] ein kombinierter Modell- und Ableitungsgenerator entwickelt, der speziell für chemische Reaktionssysteme aus der Modellspezifikation effizient die Modellgleichungen und mittels Automatischer Differentiation die Ableitungsroutinen erzeugen kann.

Kapitel 7

Behandlung von Mehrfachexperimenten

In der Regel bestehen chemische Versuche nicht nur aus Einzelexperimenten. Diese reichen oft auch gar nicht aus, um genügend Informationen zur Identifizierung aller Parameter zu liefern. So kann man zum Beispiel mit nur einem isothermen Experiment niemals Aktivierungsenergie und Frequenzfaktor einer Reaktion gleichzeitig schätzen. Daher werden Serien von Experimenten durchgeführt, dabei werden im allgemeinen die Versuchsbedingungen und eventuell sogar der Versuchsaufbau variiert. Außerdem existieren oft schon experimentelle Daten aus vergangenen Versuchsreihen, die berücksichtigt werden sollen. Die Planung solcher *Mehrfachexperimente*, die aus gegebenen experimentellen Ressourcen maximalen Informationsgehalt liefern sollen, ist eine Aufgabenstellung, die sich intuitiv nur sehr schwer bewältigen läßt und bei der daher die in dieser Arbeit vorgestellten Methoden angewendet werden sollten. Wir haben anhand mehrerer Anwendungsfälle Vergleiche zwischen intuitiver Vorgehensweise und optimaler Versuchsplanung bei Mehrfachexperimenten durchgeführt, die diese These eindrucksvoll bestätigen, siehe Kapitel 10.

Eine erste, naive Herangehensweise an diese Problemklasse könnte einfach darin bestehen, die Modelle für alle Experimente zu einem großen Gesamtsystem zusammenzufassen und dieses mit den bisher diskutierten Methoden zu behandeln. Da dabei jedoch alle Strukturen, die sich durch die Mehrfachexperimentsituation ergeben, ungenutzt bleiben, ist diese Vorgehensweise sehr ineffizient, besonders bei einer großen Zahl von Experimenten. Wir betrachten daher alle Experimente einzeln, formulieren Mehrfachexperimentprobleme und lösen diese unter Berücksichtigung dieser Strukturen.

7.1 Modelle für die Experimente

Wir nehmen also nun an, eine Mehrfachexperimentversuchsreihe bestehe aus $N_{ex} \in \mathbb{N}$ Experimenten. Wir lassen größtmögliche Variabilität bei der Beschreibung der einzelnen Experimente zu. Diese können verschiedene Modelle und unterschiedliche Versuchsräume haben:

Für $j = 1, \ldots, N_{ex}$ werde der zugrundeliegende Prozeß durch das DAE-Modell

$$\begin{aligned} A^j(t, y^j(t), z^j(t), p, q^j)\ \dot{y^j}(t) &= f^j(t, y^j(t), z^j(t), p, q^j, u^j(t)) \\ 0 &= g^j(t, y^j(t), z^j(t), p, q^j, u^j(t)) \end{aligned}$$

mit

- $t \in [t_0^j, t_{end}^j] \subseteq I\!R$ („Zeit“),
- $x^j(t) = (y^j(t), z^j(t)) \in I\!R^{n_{y^j}+n_{z^j}}$ Zustandsgrößen,
- $q^j \in I\!R^{n_{q^j}}$ Steuergrößen und
- $u^j(t) \in I\!R^{n_{u^j}}$ Steuerfunktionen

beschrieben. Der Vektor der unbekannten Parameter $p \in I\!R^{n_p}$ sei dagegen allen Experimenten gemeinsam. Seiner Schätzung soll die Versuchsreihe ja dienen.

Für $j = 1, \ldots, N_{ex}$ seien weiterhin Meßdaten η_i^j mit Meßfehlern ϵ_i^j, $i = 1, \ldots, M^j$ gegeben. Wir machen wieder die folgende Annahme an das Meßmodell:

$$\eta_i^j = h_i^j(t_i^j, x^j(t_i^j), p, q^j, u^j(t_i^j)) + \epsilon_i^j, \quad i = 1, \ldots, M^j$$

mit unabhängigen und normalverteilten Meßfehlern

$$\epsilon_i^j \sim \mathcal{N}\left(0, \frac{{\sigma_i^j}^2}{w_i^j}\right), \quad i = 1, \ldots, M^j.$$

7.2 Mehrfachexperiment-Parameterschätzprobleme

Zur Schätzung der Parameter p aus den Meßdaten der N_{ex} Experimente formulieren wir nun das folgende Mehrfachexperiment-Parameterschätzproblem:

$$\begin{aligned} \min_{p, x^1, \ldots, x^{N_{ex}}} \quad & \frac{1}{2} \sum_{j=1}^{N_{ex}} \sum_{i=1}^{M^j} w_i^j \cdot \frac{(\eta_i^j - h_i^j(t_i^j, x^j(t_i^j), p, q^j, u^j(t_i^j)))^2}{{\sigma_i^j}^2} \\ \text{s. t.} \quad & \text{für } j = 1, \ldots, N_{ex} \\ A^j(t, y^j, z^j, p, q^j)\ \dot{y^j} &= f^j(t, y^j, z^j, p, q^j, u^j) \\ 0 &= g^j(t, y^j, z^j, p, q^j, u^j) \\ 0 &= \hat{d}^j(x^j(t_1^j), \ldots, x^j(t_{\hat{K}^j}^j), p, q^j) \end{aligned}$$

Zur Darstellung als endlichdimensionales Problem nehmen wir die Parametrisierung der Lösung der DAE experimentweise vor, dabei führen wir wie in Kapitel 4 Parametrisie-

rungsvariablen $s^j \in I\!R^{n_{sj}}$, $j = 1, \ldots, N_{ex}$ und zusätzliche Nebenbedingungen ein. Wir erhalten dadurch ein endlichdimensionales beschränktes Least-Squares-Problem:

$$\begin{aligned} \min_{p,s^1,\ldots,s^{N_{ex}}} \quad & \frac{1}{2}\sum_{j=1}^{N_{ex}}\sum_{i=1}^{M^j} w_i^j \cdot \frac{(\eta_i^j - h_i^j(t_i^j, x^j(t_i^j, p, q^j, u^j, s^j), p, q^j, u^j(t_i^j)))^2}{{\sigma_i^j}^2} && (7.1) \\ \text{s. t.} \quad & \text{für } j = 1, \ldots, N_{ex} \\ 0 \;=\; & d^j(x^j(t_1^j, p, q^j, u^j, s^j), \ldots, x^j(t_{K^j}^j, p, q^j, u^j, s^j), p, q^j, s^j) && (7.2) \end{aligned}$$

Durch die Definition von

$$\begin{aligned} F_1^j &:= \left(\sqrt{w_i^j} \cdot \frac{\eta_i^j - h_i^j(t_i^j, x^j(t_i^j, p, q^j, u^j, s^j), p, q^j, u^j(t_i^j))}{\sigma_i^j}\right)_{i=1,\ldots,M^j} \\ F_2^j &:= d^j \end{aligned}$$

sowie

$$F_1 := \begin{pmatrix} F_1^1 \\ \vdots \\ F_1^{N_{ex}} \end{pmatrix}, \qquad F_2 := \begin{pmatrix} F_2^1 \\ \vdots \\ F_2^{N_{ex}} \end{pmatrix}$$

und mit $s := \left(s^1, \ldots, s^{N_{ex}}\right)$ können wir (7.1)-(7.2) kompakt schreiben als

$$\begin{aligned} \min_{p,s} \quad & \frac{1}{2}\|F_1(p,s)\|_2^2 \\ & F_2(p,s) = 0. \end{aligned}$$

Die Jacobimatrizen dieses Problems haben für Mehrfachexperimentprobleme nun die folgende Struktur:

$$J_1 = \frac{\mathrm{d}F_1}{\mathrm{d}(p,s)} = \begin{pmatrix} J_{1,p}^1 & J_{1,s^1}^1 & 0 & \cdots & 0 \\ \vdots & & \ddots & & \\ J_{1,p}^{N_{ex}} & 0 & \cdots & 0 & J_{1,s^{N_{ex}}}^{N_{ex}} \end{pmatrix}$$

und

$$J_2 = \frac{\mathrm{d}F_2}{\mathrm{d}(p,s)} = \begin{pmatrix} J_{2,p}^1 & J_{2,s^1}^1 & 0 & \cdots & 0 \\ \vdots & & \ddots & & \\ J_{2,p}^{N_{ex}} & 0 & \cdots & 0 & J_{2,s^{N_{ex}}}^{N_{ex}} \end{pmatrix}$$

mit

$$J_{1,p}^j := \frac{\mathrm{d}F_1^j}{\mathrm{d}p}, \quad J_{1,s^j}^j := \frac{\mathrm{d}F_1^j}{\mathrm{d}s^j}, \quad J_{2,p}^j := \frac{\mathrm{d}F_2^j}{\mathrm{d}p}, \quad J_{2,s^j}^j := \frac{\mathrm{d}F_2^j}{\mathrm{d}s^j}, \quad j = 1, \ldots, N_{ex}.$$

Methoden und Software zur strukturausnutzenden Lösung von Mehrfachexperiment-Parameterschätzproblemen wurden von Schlöder [104] und v. Schwerin [113] entwickelt.

7.3 Mehrfachexperiment-Versuchsplanung

Die statistische Analyse im Lösungspunkt der Parameterschätzung liefert die Kovarianzmatrix und Konfidenzgebiete für die Schätzung aus den Daten aller Experimente. Die Minimierung dieser Konfidenzgebiete führt, wie in Kapitel 5 dargestellt, auf Mehrfachexperiment-Versuchsplanungsprobleme.

7.3.1 Feste und variable Experimente

Deren Formulierung eröffnet uns eine neue Möglichkeit für die Versuchsplanung. Wir können bei Mehrfachexperiment-Szenarien zusätzlich zu vorliegenden Daten aus bereits durchgeführten Versuchen neue Experimente entwerfen, um unter Berücksichtigung der Vorinformation einen Zugewinn an Schätzgüte durch Verbesserung und Ergänzung der Meßinformation zu erzielen.

Die alten Experimente werden bei dieser Mehrfachexperiment-Versuchsplanung festgehalten, nur die neuen werden durch die Optimierung ausgelegt. Die Zielfunktion beschreibt die Güte einer Schätzung aus den Daten aller Experimente, der schon vorliegenden und der neu zu entwerfenden.

Das Mehrfachexperiment aus N_{ex} Experimenten setzt sich dabei also zusammen aus

- N_{fix} bereits durchgeführten Versuchen, von denen schon Meßdaten vorliegen und deren Sensitivitäten (Jacobimatrizen) als Vorinformationen in der Versuchsplanung mitberücksichtigt werden sollen, und

- N_{var} neu zu planenden Experimenten, deren Steuergrößen, Steuerfunktionen und Meßlayout in der Versuchsplanung optimal bestimmt werden sollen.

Gemäß dieser Unterscheidung zwischen festen und variablen Experimenten lassen sich auch in der Jacobimatrix des Parameterschätzproblems (7.1)-(7.2) entsprechende Blöcke identifizieren:

$$J_1 = \left(\begin{array}{c|ccc|c} J_{1,p}^{1} & J_{1,s^1}^{1} & 0 \cdots & 0 & \\ \vdots & & \ddots & & 0 \\ J_{1,p}^{N_{fix}} & 0 & \cdots 0 & J_{1,s^{N_{fix}}}^{N_{fix}} & \\ \hline J_{1,p}^{N_{fix}+1} & & & & J_{1,s^{N_{fix}+1}}^{N_{fix}+1} \quad 0 \cdots 0 \\ \vdots & & 0 & & \ddots \\ J_{1,p}^{N_{ex}} & & & & 0 \quad \cdots 0 \quad J_{1,s^{N_{ex}}}^{N_{ex}} \end{array}\right)$$

bzw.

$$J_2 = \left(\begin{array}{c|ccc|c} J_{2,p}^1 & J_{2,s^1}^1 & 0 & \cdots & 0 & \\ \vdots & & \ddots & & & 0 \\ J_{2,p}^{N_{fix}} & 0 & \cdots & 0 & J_{2,s^{N_{fix}}}^{N_{fix}} & \\ \hline J_{2,p}^{N_{fix}+1} & & & & J_{2,s^{N_{fix}+1}}^{N_{fix}+1} & 0 \quad \cdots \quad 0 \\ \vdots & & 0 & & & \ddots \\ J_{2,p}^{N_{ex}} & & & & 0 & \cdots \quad 0 \quad J_{2,s^{N_{ex}}}^{N_{ex}} \end{array}\right).$$

Die Kovarianzmatrix in der Darstellung

$$C = \begin{pmatrix} I & 0 \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_1^T J_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-T} \begin{pmatrix} I \\ 0 \end{pmatrix}$$

hat damit ebenfalls eine spezielle Mehrfachexperimentstruktur mit

$$H := J_1^T J_1 = \left(\begin{array}{c|ccc} H_{pp} & H_{s^1 p}^T & \cdots & H_{s^{N_{ex}} p}^T \\ \hline H_{s^1 p} & H_{s^1 s^1} & & 0 \\ \vdots & & \ddots & \\ H_{s^{N_{ex}} p} & 0 & & H_{s^{N_{ex}} s^{N_{ex}}} \end{array}\right) \tag{7.3}$$

und

$$H_{pp} := \sum_{j=1}^{N_{ex}} {J_{1,p}^j}^T J_{1,p}^j = \underbrace{\sum_{j=1}^{N_{fix}} {J_{1,p}^j}^T J_{1,p}^j}_{=:H_{pp}^{fix} \text{ (fest)}} + \underbrace{\sum_{j=N_{fix}+1}^{N_{ex}} {J_{1,p}^j}^T J_{1,p}^j}_{\text{(variabel)}}, \tag{7.4}$$

$$H_{s^j p} := {J_{1,s^j}^j}^T J_{1,p}^j, \quad j = 1, \ldots, N_{ex}, \tag{7.5}$$

$$H_{s^j s^j} := {J_{1,s^j}^j}^T J_{1,s^j}^j, \quad j = 1, \ldots, N_{ex}. \tag{7.6}$$

7.3.2 Mehrfachexperiment-Versuchsplanungsprobleme

Die Steuergrößen, Steuerfunktionen und Meßgewichte der einzelnen Experimente spielen bei der Versuchsplanung verschiedene Rollen:

- Für die alten Experimente $j = 1, \ldots, N_{fix}$ sind sie feste Größen, das heißt Konstanten: $\left(q^1, u^1, w^1, \ldots, q^{N_{fix}}, u^{N_{fix}}, w^{N_{fix}}\right) = konst.$
- Für die neuen Experimente $j = N_{fix} + 1, \ldots, N_{ex}$ sind sie Versuchsplanungsgrößen, das heißt Variablen: $\xi = \left(q^{N_{fix}+1}, u^{N_{fix}+1}, w^{N_{fix}+1}, \ldots, q^{N_{ex}}, u^{N_{ex}}, w^{N_{ex}}\right)$.

Wir nehmen an, daß die Nebenbedingungen an Zustände und Versuchsplanungsgrößen experimentweise gestellt werden.

Die Formulierung des Mehrfachexperiment-Versuchsplanungsproblems lautet somit:

$$\min_{\xi, x^1, \ldots, x^{Nex}} \quad \phi(C)$$

Dabei berechnet sich die Kovarianzmatrix

$$C = \begin{pmatrix} I & 0 \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-1} \begin{pmatrix} J_1^T J_1 & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} J_1^T J_1 & J_2^T \\ J_2 & 0 \end{pmatrix}^{-T} \begin{pmatrix} I \\ 0 \end{pmatrix}$$

aus der Jacobimatrix $\begin{pmatrix} J_1^T & J_2^T \end{pmatrix}^T$ im Lösungspunkt
des beschränkten Mehrfachexperiment-Parameterschätzproblems

$$\begin{aligned} \min_{p, s^1, \ldots, s^{Nex}} \quad & \frac{1}{2} \sum_{j=1}^{Nex} \sum_{i=1}^{M^j} w_i^j \cdot \frac{(\eta_i^j - h_i^j(t_i^j, x^j(t_i^j, p, q^j, u^j, s^j), p, q^j, u^j(t_i^j)))^2}{{\sigma_i^j}^2} \\ \text{s. t.} \quad & \text{für } j = 1, \ldots, N_{ex} \\ 0 \;=\; & d^j(x^j(t_1^j, p, q^j, u^j, s^j), \ldots, x^j(t_{K^j}^j, p, q^j, u^j, s^j), p, q^j, s^j). \end{aligned}$$

Die Zustandsvariablen erfüllen experimentweise
die DAE-Systeme $j = 1, \ldots, N_{ex}$

$$\begin{aligned} A^j(t, y^j, z^j, p, q^j)\, \dot{y}^j &= f^j(t, y^j, z^j, p, q^j, u^j) \\ 0 &= g^j(t, y^j, z^j, p, q^j, u^j). \end{aligned}$$

Außerdem müssen die Nebenbedingungen

$$\begin{aligned} lo^j &\le \psi^j(t, x^j, p, q^j, u^j, w^j) \le up^j \\ 0 &= \chi^j(t, x^j, p, q^j, u^j, w^j) \\ w^j &\in \{0,1\}^{M^j} \end{aligned}$$

für $j = N_{fix} + 1, \ldots, N_{ex}$ eingehalten werden.

7.3.3 Ausnutzung der Strukturen bei der numerischen Behandlung

Die wesentliche Ersparnis an Rechenzeit beim Lösen von Mehrfachexperimentproblemen gegenüber unstrukturierter Behandlung besteht darin, daß die Modelle der einzelnen Experimente entkoppelt ausgewertet werden können, und zwar nicht nur für die Funktionswerte, sondern vor allem auch für die Berechnung der benötigten ersten und zweiten Ableitungen bei der Integration der jeweiligen DAE-Systeme.

Bei der Berechnung der Kovarianzmatrix wird die Blockstruktur der Jacobimatrizen gemäß (7.3)-(7.6) berücksichtigt.

Durch experimentweise Integration und Auswertung der Meßfunktionen werden $J_{1,p}^j$, J_{1,s^j}^j, $J_{2,p}^j$ und J_{2,s^j}^j bereitgestellt und daraus ${J_{1,p}^j}^T J_{1,p}^j$ als Summand für H_{pp} sowie $H_{s^j p}$ und $H_{s^j s^j}$ berechnet.

Die Modelle der festen Experimente müssen dabei nur einmal in der ersten Iteration ausgewertet oder können aus früheren Rechnungen, zum Beispiel zur Parameterschätzung,

übernommen werden. Die entsprechenden Blöcke der Jacobimatrizen

$$\left(\begin{array}{c|ccc} J^1_{1,p} & J^1_{1,s^1} & 0 \cdots & 0 \\ \vdots & & \ddots & \\ J^{N_{fix}}_{1,p} & 0 & \cdots 0 & J^{N_{fix}}_{1,s^{N_{fix}}} \end{array}\right), \qquad \left(\begin{array}{c|ccc} J^1_{2,p} & J^1_{2,s^1} & 0 \cdots & 0 \\ \vdots & & \ddots & \\ J^{N_{fix}}_{2,p} & 0 & \cdots 0 & J^{N_{fix}}_{2,s^{N_{fix}}} \end{array}\right)$$

und

$$H^{fix}_{pp} = \sum_{j=1}^{N_{fix}} {J^j_{1,p}}^T J^j_{1,p}$$

werden gespeichert.

Die Berechnung der Ableitungen der Kovarianzmatrix erfolgt gemäß der Formeln aus Kapitel 6. Dabei wird natürlich auch berücksichtigt, daß nur $J^j_{1,p}, J^j_{1,s^j}, J^j_{2,p}, J^j_{2,s^j}$ von q^j, u^j, w^j abhängen.

Kapitel 8

Robuste Versuchsplanung

Wir haben bisher die Gütekriterien auf der Kovarianzmatrix (4.21) nur als Funktionen der Versuchsplanungsgrößen betrachtet und optimiert. Im allgemeinen hängen sie jedoch auch von den Werten der Modellparameter ab. In der Regel sind diese nicht oder nur ungenau bekannt und sollen erst durch die Methoden der Parameterschätzung und Versuchsplanung genau bestimmt werden.

Im Gegensatz dazu hängt in der linearen Versuchsplanung, siehe zum Beispiel [93], die Modellantwort der Messungen linear von den Parametern ab, wodurch die Kovarianzmatrix keine Parameterabhängigkeit aufweist.

Zur Berücksichtigung der Parameterabhängigkeit bei nichtlinearen Versuchsplanungsproblemen schlagen wir zwei Ansätze vor, die auch kombiniert werden können.

8.1 Sequentielle Versuchsplanung und Parameterschätzung

Bei dieser Vorgehensweise werden abwechselnd Experimente geplant, Versuche durchgeführt und die Parameter geschätzt. So werden sukzessive Mehrfachexperimente entworfen. Durch die Versuchsplanung wird dabei der Zugewinn an Information optimiert. Dazwischen können die aus den experimentellen Daten neu gewonnenen Informationen genutzt werden, um den Parameterschätzer zu verbessern [68].

Man geht dabei folgendermaßen vor, vergleiche auch Abbildung 8.1:

1. Starte mit einem Anfangsschätzer $p_{(0)}$ für die Parameter. Dessen Werte erhält man zum Beispiel aus der Literatur, aus verwandten Prozessen oder durch die Auswertung von $N_{ex,0}$ Vorversuchen. $k := 0$.

2. Setze $k := k + 1$.

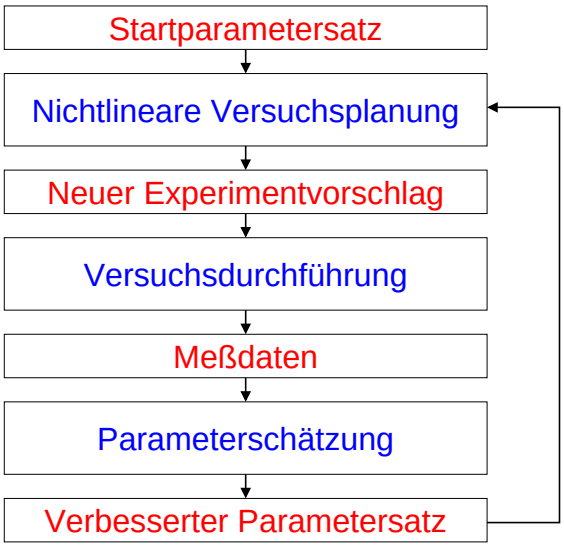


Abbildung 8.1: Sequentielle Versuchsplanung und Parameterschätzung

3. **Versuchsplanung:** Plane $N_{ex,k}$ zusätzliche Experimente unter Berücksichtigung aller schon durchgeführten $N_{ex,0} + \ldots + N_{ex,k-1}$ Experimente für die Parameterwerte $p_{(k-1)}$.

4. Die $N_{ex,k}$ neuen Experimente werden im Labor durchgeführt, dabei werden neue Meßdaten erhoben.

5. **Parameterschätzung:** Führe eine Mehrfachexperiment-Parameterschätzung für alle bisher erhobenen Daten der $N_{ex,0} + \ldots + N_{ex,k}$ Experimente durch, dies liefert den neuen Schätzer $p_{(k)}$.

6. Die statistische Analyse der Parameterschätzung liefert die Kovarianzmatrix und die Konfidenzintervalle des neuen Schätzers.

7. Entscheide abhängig von der Güte der Schätzung und dem insgesamt verfügbaren experimentellen Budget, ob die Schätzung akzeptiert oder in einer weiteren Iteration verbessert werden soll. Im zweiten Fall weiter bei 2.

Bemerkung 8.1.1 *Bei der Versuchsplanung in Schritt 3 werden die Vorinformationen aus den schon durchgeführten Experimenten wie in Kapitel 7 beschrieben ausgenutzt. Dabei wird ein Mehrfachexperiment aus $N_{ex,0} + \ldots + N_{ex,k-1}$ alten Experimenten mit festen Steuerungen und Meßgewichten und $N_{ex,k}$ neuen Experimenten mit variablen Steuerungen und Meßgewichten optimiert.*

Bemerkung 8.1.2 *Bei dieser sequentiellen Vorgehensweise sind nicht nur Änderungen der Parameterwerte möglich. Vielmehr kann von Iteration zu Iteration auch die Modellformulierung geändert oder verfeinert werden, wenn die experimentellen Daten dafür neue Hinweise ergeben. Dies kann durch Methoden zur Modelldiskriminierung unterstützt werden, siehe zum Beispiel [35].*

Konvergenzresultate zur sequentiellen Vorgehensweise bei nichtlinearen Modellen finden sich bei Chaudhuri und Mykland [32].

Beispiele zur erfolgreichen Anwendung der sequentiellen Versuchsplanung und Parameterschätzung geben wir in Kapitel 10.

Bemerkung 8.1.3 *Eine Weiterentwicklung dieser Vorgehensweise zur* Online-Versuchsplanung *ist prinzipiell möglich. Man könnte dabei die Meßinformation nicht erst nach Ende eines Experiments für die weitere Planung nutzen, sondern sofort nach jeder Probennahme eine neue Parameterschätzung durchführen und die weitere Durchführung des Prozesses auf der Grundlage dieses neuen Schätzers optimieren. Diese Verfahrensweise erfordert aber die Verfügbarkeit von Methoden zur Echtzeit-Parameterschätzung und Echtzeit-Versuchsplanungsoptimierung. Die Entwicklung solcher Methoden ist Ziel zukünftiger Arbeiten.*

8.2 Berücksichtigung der Parameterstreuung bei der Versuchsplanung

Bei diesem Ansatz berücksichtigen wir die Parameterstreuung durch eine Worst-Case-Formulierung: Ziel der Optimierung ist ein Versuchsplan, der auch unter einer gewissen Unsicherheit der Parameter das Gütekriterium minimiert.

8.2.1 Formulierung eines robusten Versuchsplanungsproblems

Wir betrachten hier nun zunächst unbeschränkte Versuchsplanungsprobleme.

Wir wollen im Auswertungspunkt p_0 einen optimalen Versuchsplan berechnen. Die Parameter seien jedoch nur ungenau bekannt, wir nehmen an, eine multinormalverteilte Streuung mit (symmetrischer) Kovarianzmatrix Σ und Konfidenzgebiet

$$\{p : \|p - p_0\|_{2,\Sigma^{-1}} \leq \gamma\}$$

sei gegeben. Dabei werde die Norm $\|.\|_{2,\Sigma^{-1}}$ durch das Skalarprodukt $< x, y >= x^T \Sigma^{-1} y$ induziert.

Diese Parameterstreuung wird durch das folgende *Worst-Case-Design* berücksichtigt:

$$\min_{\xi \in \Omega} \max_{\|p-p_0\|_{2,\Sigma^{-1}} \leq \gamma} \phi(C(\xi, p)). \tag{8.1}$$

Diese Formulierung führt auf Probleme der Semi-Infiniten Programmierung. Bei numerischen Verfahren dafür ist jeweils die Lösung globaler Optimierungsaufgaben notwendig. Einen Überblick über Theorie und Methoden der Semi-Infiniten Programmierung gibt der Artikel von Hettich und Kortanek [64].

Um diese Schwierigkeiten zu umgehen, betrachten wir eine vereinfachte Fragestellung [69]. Wir entwickeln die Zielfunktion in (8.1) nach den Parametern p und erhalten in erster Näherung

$$\min_{\xi\in\Omega} \max_{\|p-p_0\|_{2,\Sigma^{-1}}\leq\gamma} \phi(C(\xi,p_0)) + \frac{\mathrm{d}}{\mathrm{d}p}\phi(C(\xi,p_0))(p-p_0). \tag{8.2}$$

Die Lösung des inneren, linearen Problems können wir nun explizit angeben.

Lemma 8.2.1

$$\max_{\|p-p_0\|_{2,\Sigma^{-1}}\leq\gamma} \phi(C(\xi,p_0)) + \frac{d}{dp}\phi(C(\xi,p_0))(p-p_0) = \phi(C(\xi,p_0)) + \gamma \left\| \frac{d}{dp}\phi(C(\xi,p_0)) \right\|_{2,\Sigma}.$$

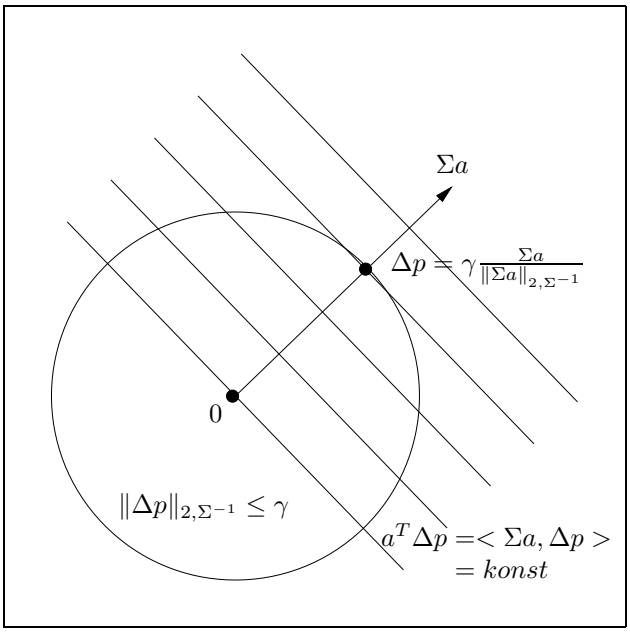


Abbildung 8.2: Geometrische Herleitung des Maximalpunktes des inneren Problems. Man beachte: der euklidische Raum ist mit dem Skalarprodukt $< x, y >= x^T\Sigma^{-1}y$ versehen.

Beweis. Vergleiche Abbildung 8.2. Sei $\Delta p := p - p_0$ und $a := \frac{\mathrm{d}}{\mathrm{d}p}\phi(C(\xi,p_0))^T$. Aus Monotoniegründen wird das Maximum am Rand angenommen, also betrachten wir

$$\max_{\|\Delta p\|^2_{2,\Sigma^{-1}}=\gamma^2} a^T\Delta p$$

mit der Lagrangefunktion

$$\mathcal{L}(\Delta p,\lambda) = a^T\Delta p - \lambda\cdot\left(\Delta p^T\Sigma^{-1}\Delta p - \gamma^2\right).$$

Nullsetzen des Gradienten

$$\nabla_{\Delta p}\mathcal{L}(\Delta p,\lambda) = a - 2\cdot\lambda\cdot\Sigma^{-1}\Delta p = 0$$

ergibt

$$\Delta p = \frac{\Sigma a}{2 \cdot \lambda}.$$

λ erhalten wir durch Einsetzen in die Nebenbedingung:

$$\lambda = \pm\frac{1}{2}\sqrt{\frac{a^T \Sigma a}{\gamma^2}} = \pm\frac{1}{2}\frac{\|a\|_{2,\Sigma}}{\gamma}.$$

Die negative Lösung führt auf ein Minimum, das Maximum

$$\Delta p = \gamma \frac{\Sigma a}{\|a\|_{2,\Sigma}}$$

hat den Zielfunktionswert

$$a^T \Delta p = \gamma \frac{a^T \Sigma a}{\|a\|_{2,\Sigma}} = \gamma \|a\|_{2,\Sigma} = \gamma \left\|a^T\right\|_{2,\Sigma}.$$

■

Durch die explizite Lösung des inneren, linearen Problems erhalten wir für die robuste Optimierungsaufgabe nun statt der Min-Max-Formulierung (8.2) die folgende Darstellung:

$$\min_{\xi \in \Omega} \phi(C(\xi, p_0)) + \gamma \left\| \frac{\mathrm{d}}{\mathrm{d}p} \phi(C(\xi, p_0)) \right\|_{2,\Sigma}.$$

8.2.2 Numerische Behandlung

Zur Lösung dieses Optimierungsproblems setzen wir wiederum SQP-Verfahren ein. Dazu müssen Zielfunktionswerte und Zielfunktionsgradienten bereitgestellt werden.

Aus der Darstellung der robusten Zielfunktion ϕ_R

$$\phi_R = \phi + \gamma \left\| \frac{\mathrm{d}\phi}{\mathrm{d}p} \right\|_{2,\Sigma} = \phi + \gamma \left(\sum_i \left(\sum_j \sigma_{ij} \frac{\partial \phi}{\partial p_j} \right)^2 \right)^{\frac{1}{2}}$$

ergibt sich

$$\frac{\mathrm{d}\phi_R}{\mathrm{d}\xi_k} = \frac{\mathrm{d}\phi}{\mathrm{d}\xi_k} + \gamma \cdot \frac{1}{2} \cdot \frac{1}{\left\| \frac{\partial \phi}{\partial p} \right\|_{2,\Sigma}} \cdot \sum_i \left(2 \cdot \sum_j \sigma_{ij} \frac{\partial \phi}{\partial p_j} \cdot \sum_j \sigma_{ij} \frac{\partial^2 \phi}{\partial \xi_k \partial p_j} \right).$$

Bei der Berechnung können wir einige Teilausdrücke mehrfach verwenden. Wir definieren daher

$$a_i := \sum_j \sigma_{ij} \frac{\partial \phi}{\partial p_j}, \quad b := \left(\sum_i a_i^2 \right)^{\frac{1}{2}}, \quad d := \sum_j \left(\frac{\partial^2 \phi}{\partial \xi_k \partial p_j} \sum_i a_i \sigma_{ij} \right)$$

und erhalten

$$\phi_R = \phi + \gamma \cdot b, \quad \frac{\mathrm{d}\phi_R}{\mathrm{d}\xi_k} = \frac{\mathrm{d}\phi}{\mathrm{d}\xi_k} + \gamma \cdot \frac{d}{b}.$$

Man erkennt, daß wir gegenüber der nichtrobusten Versuchsplanung zusätzliche Ableitungen benötigen, und zwar $\frac{\partial \phi}{\partial p_j}$ und $\frac{\partial^2 \phi}{\partial \xi_k \partial p_j}$, also die Sensitivitäten von nichtrobuster Zielfunktion und nichtrobustem Zielfunktionsgradient nach den Parametern.

Für die Ableitung von ϕ nach den Parametern p setzen wir einen Differenzenquotienten an:

$$\frac{\partial}{\partial p_j}\phi(C(\xi, p_0)) \doteq \frac{\phi(C(\xi, p_0 + \Delta p_j)) - \phi(C(\xi, p_0))}{\Delta p_j},$$

ebenso für die gemischte Ableitung von ϕ nach den Parametern p und den Versuchsplanungsgrößen ξ:

$$\begin{aligned}
\frac{\partial^2}{\partial \xi_k \partial p_j}\phi(C(\xi, p_0)) &= \frac{\partial}{\partial p_j}\left(\frac{\partial}{\partial \xi_k}\phi(C(\xi, p_0))\right) \\
&\doteq \frac{\frac{\partial}{\partial \xi_k}\phi(C(\xi, p_0 + \Delta p_j)) - \frac{\partial}{\partial \xi_k}\phi(C(\xi, p_0))}{\Delta p_j} \\
&= \frac{\partial}{\partial \xi_k}\left(\frac{\phi(C(\xi, p_0 + \Delta p_j)) - \phi(C(\xi, p_0))}{\Delta p_j}\right).
\end{aligned}$$

Dabei ist also die Approximation der Ableitung der robusten Zielfunktion die exakte Ableitung der Approximation der robusten Zielfunktion.

Der einzige Approximationsfehler entsteht durch das Ersetzen von $\frac{\partial \phi}{\partial p_j}$ durch einen Differenzenquotienten. Daß diese Ableitung dadurch nur mit eingeschränkter Genauigkeit berechnet wird, ist aber vertretbar, da es sich bei der Zielfunktion der robusten Versuchsplanung ohnehin schon um eine (lineare) Näherung handelt.

8.2.3 Behandlung von Beschränkungen

Bisher haben wir nur robuste Versuchsplanung für unbeschränkte Probleme betrachtet.

Bei beschränkten Versuchsplanungsproblemen können einerseits *parameterunabhängige* Nebenbedingungen wie zum Beispiel Steuerbeschränkungen oder Gewichtebedingungen auftreten. Ihre Behandlung erfolgt im robusten Fall wie in der nichtrobusten Versuchsplanung.

Für *parameterabhängige* Nebenbedingungen, wir betrachten hier Ungleichungen der Form

$$\psi(\xi, p_0) \leq 0,$$

verlangen wir, daß sie im Worst-Case-Sinne, das heißt von allen Parametern im Gebiet

$$\{p : \|p - p_0\|_{2,\Sigma^{-1}} \leq \gamma\}$$

erfüllt werden:

$$\max_{\|p-p_0\|_{2,\Sigma^{-1}}\leq\gamma} \psi(\xi,p) \leq 0.$$

Wie bei der Zielfunktion nehmen wir nun eine lineare Entwicklung um p_0 vor

$$\max_{\|p-p_0\|_{2,\Sigma^{-1}}\leq\gamma} \psi(\xi,p_0) + \frac{\mathrm{d}}{\mathrm{d}p}\psi(\xi,p_0)(p-p_0) \leq 0$$

und können wie oben die Lösung explizit darstellen:

$$\max_{\|p-p_0\|_{2,\Sigma^{-1}}\leq\gamma} \psi(\xi,p_0) + \frac{\mathrm{d}}{\mathrm{d}p}\psi(\xi,p_0)(p-p_0) = \psi(\xi,p_0) + \gamma \left\| \frac{\mathrm{d}}{\mathrm{d}p}\psi(\xi,p_0) \right\|_{2,\Sigma}.$$

Dies führt auf robuste Nebenbedingungen

$$\psi_R(\xi,p_0) := \psi(\xi,p_0) + \gamma \left\| \frac{\mathrm{d}}{\mathrm{d}p}\psi(\xi,p_0) \right\|_{2,\Sigma} \leq 0.$$

Zur Lösung des Versuchsplanungsproblems mit SQP-Verfahren wird neben der Auswertung der Nebenbedingungen auch die Bereitstellung deren Gradienten benötigt:

$$\frac{\mathrm{d}\psi_R}{\mathrm{d}\xi_k} = \frac{\mathrm{d}\psi}{\mathrm{d}\xi_k} + \gamma \cdot \frac{1}{2} \cdot \frac{1}{\left\| \frac{\partial\psi}{\partial p} \right\|_{2,\Sigma}} \cdot \sum_i \left(2 \cdot \sum_j \sigma_{ij} \frac{\partial\psi}{\partial p_j} \cdot \sum_j \sigma_{ij} \frac{\partial^2\psi}{\partial\xi_k \partial p_j} \right).$$

Wie bei der Zielfunktion setzen wir für die Ableitung nach den Parametern einen Differenzenquotienten an

$$\frac{\mathrm{d}}{\mathrm{d}p_j}\psi(\xi,p_0) \doteq \frac{\psi(\xi,p_0+\Delta p_j) - \psi(\xi,p_0)}{\Delta p_j}$$

bzw.

$$\begin{aligned} \frac{\partial^2}{\partial\xi_k\partial p_j}\psi(\xi,p_0) &= \frac{\partial}{\partial p_j}\left(\frac{\partial}{\partial\xi_k}\psi(\xi,p_0)\right) \\ &\doteq \frac{\frac{\partial}{\partial\xi_k}\psi(\xi,p_0+\Delta p_j) - \frac{\partial}{\partial\xi_k}\psi(\xi,p_0)}{\Delta p_j} \\ &= \frac{\partial}{\partial\xi_k}\left(\frac{\psi(\xi,p_0+\Delta p_j) - \psi(\xi,p_0)}{\Delta p_j}\right). \end{aligned}$$

Ein Beispiel für robuste Versuchsplanung unter Berücksichtigung der Parameterabhängigkeit findet sich in Abschnitt 10.4 in Kapitel 10.

Kapitel 9

Das Softwarepaket VPLAN

9.1 Überblick

Die in dieser Arbeit beschriebenen mathematischen Methoden wurden in dem Softwarepaket VPLAN realisiert. Bei Konzeption und Implementierung wurde auf die folgenden Prinzipien Wert gelegt:

- Verwendung bzw. Entwicklung von State-of-the-Art-Numerik,
- Unabhängigkeit von speziellen Anwendungsproblemen,
- Bedienungsfreundlichkeit,
- Plattformunabhängigkeit,
- Performance,
- Stabilität.

Als Programmiersprache wurde C++ [110] gewählt. Dadurch erhält man die nötige Flexibilität, um die komplexen Datenstrukturen von Versuchsplanungsproblemen im Rechner darstellen zu können. Da auch auf vorhandene Numerik-Bibliotheken zugegriffen wird, sind Teile des Programms in Fortran implementiert, die über geeignete Schnittstellen mit den C++-Klassen zusammenwirken.

Es wurde auf die strikte Trennung zwischen dem Programm einerseits und der Beschreibung und Implementation der Anwendungsprobleme andererseits Wert gelegt. So erreichen wir Flexibilität und Problemunabhängigkeit der Software bei gleichzeitigem hohem Bedienungskomfort. Möglich wird dies durch die Verwendung moderner Techniken für automatische Ableitungserzeugung und dynamisches Linken.

Die Anwendungsprobleme werden mit Hilfe von ASCII-Files beschrieben. Dies macht die Anbindung von grafischen Benutzeroberflächen und Modellgeneratoren einfach möglich,

zum Beispiel über Client-Server-Techniken. Mit dem Softwarepaket MGAD stellen wir dafür eine Java-Implementierung bereit.

Die Ergebnisausgabe erfolgt ebenfalls mittels ASCII-Files, zur Visualisierung der Daten ist eine Schnittstelle vorhanden.

Durch die Verwendung von Techniken der automatischen Ableitungserzeugung ist die Arbeit mit VPLAN für den Benutzer ableitungsfrei.

Das Softwarepaket VPLAN besteht im wesentlichen aus zwei Programmen: vplan98 und doit. doit („Design Of experiments Initialization Tool“) erledigt die Ableitungsgenerierung und Erzeugung des kompilierten Problemmoduls, vplan98 ist für die Durchführung der numerischen Berechnungen zuständig.

In der augenblicklichen Version benutzt VPLAN die Programmpakete SNOPT [54, 55] und ADIFOR [21, 23, 22] sowie NAG- und LAPACK-Routinen [84, 1]. Außerdem werden DAESOL [17] und eine Mehrfachexperimentversion von PARFIT [113] aufgerufen. Die Schnittstellen zu Integrator, Optimierungsslöser, Parameterschätzprogramm und Lineare-Algebra-Bibliotheken sind modular gehalten, so daß diese Komponenten gegebenenfalls durch alternative Software ausgetauscht werden können.

Bei der Entwicklung wurde auf Plattformunabhängigkeit Wert gelegt. So läuft VPLAN auf Unix-Betriebssystemen wie Linux, Solaris, Irix und AIX sowie auf Windows NT und 9x. Die Versionsverwaltung der Softwareentwicklung wird mit dem Werkzeug CVS [43] durchgeführt.

VPLAN hat sich inzwischen im professionellen Einsatz an der Universität und in der Industrie bewährt.

In den folgenden Abschnitten wollen wir eine Einführung in die Arbeitsweise und Verwendung des Softwarepakets geben. Außerdem sollen die Konzepte der Implementierung vorgestellt werden.

9.2 Arbeitsweise des Softwarepakets VPLAN

9.2.1 Leistungsumfang

Der Leistungsumfang des Programmpakets VPLAN umfaßt verschiedene mathematische Methoden zur Behandlung von Simulations- und Optimierungsaufgaben für Mehrfachexperiment-Systeme mit zugrundeliegenden DAE-Modellen.

Die folgende Übersicht stellt die einzelnen Anwendungsmodi kurz dar:

- *Integration:* Die DAE-Systeme für die Experimente des Versuchsplans werden integriert. Ausgabedaten werden bereitgestellt, anhand derer man die Trajektorien visualisieren kann. Dies benutzt man in der ersten Phase der Modellierung, bevor Nebenbedingungen und Meßverfahren spezifiziert werden.

- *Simulationsumgebung:* Zusätzlich zur Integration werden die Meßfunktionen ausgewertet. Kovarianzmatrix, Gütekriterien und die Näherungen für die Standardabweichungen der Parameter werden berechnet, außerdem die Kosten des Versuchs. Die Nebenbedingungen an Versuchsplanungsgrößen und Zustandsvariablen werden auf Zulässigkeit überprüft. Sind Meßwerte vorhanden, werden Residuen berechnet, pro Meßwert, pro Experiment und für das gesamte Mehrfachexperiment. Die Simulationsumgebung stellt damit ein machtvolles Werkzeug zur Simulation und Beurteilung von Versuchsplänen zur Verfügung. Sie kann, wenn sie interaktiv eingesetzt wird, benutzt werden, um ein intuitives Verständnis für das Verhalten des Modells zu bekommen und verschiedene Konstellationen auszuprobieren.

- *Erzeugen von Pseudo-Meßdaten:* Durch Integration der Experimente, Auswerten der Meßfunktionen und Aufaddieren von „Rauschen“ können simulierte Meßdaten erzeugt werden.

- *Parameterschätzung:* Dieser Modus ermöglicht das Lösen von Mehrfachexperiment-Parameterschätzproblemen durch Aufruf von PARFIT.

- *Versuchsplanung:* Dies ist die zentrale Aufgabe von VPLAN. Die Versuchsplanungsgrößen der variablen Experimente werden so berechnet, daß ein Gütekriterium auf der Kovarianzmatrix optimiert und die Nebenbedingungen erfüllt werden. In die Berechnung der Kovarianzmatrix können neben den zu optimierenden Experimenten auch feste, bereits durchgeführte Experimente eingehen.

9.2.2 Bedienung des Programmpakets

Die Bedienung von VPLAN ist einfach und benutzerfreundlich. Zunächst muß das Programmpaket installiert und kompiliert werden, dies wird weiter unten in diesem Kapitel beschrieben. Will man nun mit einem Anwendungsproblem arbeiten, muß man die folgenden Schritte durchführen:

1. Anlegen des Problemverzeichnisses,

2. Ablegen von Modell- und Datenfiles im Problemverzeichnis,

3. Aufruf des Programms doit zur automatischen Ableitungsgenerierung und Erzeugung des Problemmoduls,

4. Aufruf von vplan98 zur Durchführung der numerischen Berechnungen, wobei einer der oben beschriebenen Modi gewählt werden kann,

5. Zugriff auf die Ausgabefiles im Problemverzeichnis.

Da die Ausgabefiles die gleiche Struktur wie die Eingabefiles haben, ist auf einfache Weise der wiederholte und sukzessive Aufruf von vplan98 möglich.

9.2.3 Spezifikation der Ein- und Ausgabedaten

Die Anforderungen zur Beschreibung von Modellen und Daten wurden gemeinsam mit Projektpartnern, Anwendern aus der chemischen Industrie und Informatikern, entwickelt und formuliert. Dabei wurden die folgenden Prinzipien zugrundegelegt:

- Flexibilität,
- Modularität,
- Trennung von Modell und Daten einerseits und Programm andererseits,
- Vermeidung von Redundanz,
- Bedienungsfreundlichkeit.

Die Beschreibung eines Anwendungsbeispiels umfaßt

- ini-Files zur Problemspezifikation,
- Fortran-Files zur Beschreibung von Modellfunktionen und
- Meßdatenfiles.

Auf Einzelheiten gehen wir unten ein.

Alle Dateien zur Problembeschreibung können natürlich auch von entsprechenden Modellgeneratoren automatisch erzeugt werden. Wir stellen mit MGAD ein solches Tool zur Verfügung, siehe unten. Die Anbindung an kommerzielle Modellierungstools ist durch die Entwicklung von Import- und Export-Filtern leicht zu bewerkstelligen.

ini-Files

Zunächst wollen wir die Files zur Problemspezifikation betrachten. Ihre Syntax wurde an die von ini-Files, wie sie zum Beispiel bei Microsoft Windows verwendet werden, angelehnt. Diese Files werden sowohl zur Spezifikation der Eingaben als auch zur Ausgabe der Programmergebnisse verwendet.

Es gibt zwei Arten von ini-Files:

- `vplan.ini` beschreibt den Gesamtrahmen des Versuchsplanungsproblems. In diesem File werden in verschiedenen Abschnitten definiert: die Aktion, die durchgeführt werden soll (Integration, Simulation, Versuchsplanung, ...), die Pfade von Problemverzeichnis und Unterverzeichnissen, die Modellparameter, die Experimente, die zusammen den Versuchsplan ergeben, das anzuwendende Gütekriterium, die Namen der Meßdatenfiles und Ausgabefiles sowie die numerischen Optionen für Versuchsplanung und Parameterschätzung. Als Ausgabefile enthält `vplan.ini` außerdem Werte aller Gütekriterien, des Residuums und der Gesamtkosten des Mehrfachexperiments.

- Zur Spezifikation der einzelnen Experimente gibt es jeweils ein `experiment.ini`-File. Es enthält Abschnitte für: das Flag, ob das Experiment festgehalten werden soll oder nicht, das Integrationsintervall, die Namen der Modellfunktionen, differentielle und algebraische Zustandsvariablen, die Beschreibung der dynamischen Nebenbedingungen und des Gitters zur Überprüfung der Nebenbedingungen, Steuergrößen und Steuerfunktionen sowie Nebenbedingungen an die Steuergrößen, die Beschreibung der Meßfunktionen und der Meßstruktur sowie Optionen für die Integration dieses Experiments.

Werden durch die Berechnungen von vplan98 Werte von Variablen verändert, so stehen in den Ausgabe-ini-Files die Ergebnisse an den Stellen, an denen in den Eingabe-ini-Files die Ausgangswerte standen.

Abbildung 9.1 zeigt exemplarische Ausschnitte aus `vplan.ini` und `experiment.ini`.

Fortran-Files

Alle benutzerdefinierten Subroutinen für Modellfunktionen, Meßfunktionen usw. werden als Fortran-Source-Files abgelegt. Fortran wurde als Beschreibungssprache gewählt, da dafür für die automatische Ableitungserzeugung mit ADIFOR [22] ein leistungsfähiges Werkzeug zur Verfügung steht.

Die folgende Übersicht beschreibt die auftretenden Typen von Fortran-Routinen zur Modellspezifikation mit einer jeweiligen Erläuterung der Aufrufparameter. Jede Subroutine wird in einem eigenen Fortran-File abgelegt, das den Namen der Subroutine und das Suffix `.f` hat.

1. *Modellfunktionen:* rechte Seiten des DAE-System für differentielle und algebraische Gleichungen `ffcn.f` und `gfcn.f`

   ```
   subroutine ffcn( t, x, f, p, q, rwh, iwh, iflag )
   ```

   - `t`: Zeit (Input)
   - `x`: Vektor der Zustandsvariablen (Input)
   - `f`: Vektor der Funktionswerte von `ffcn` (Output)
   - `p`: Vektor der Modellparameter (Input)
   - `q`: Vektor der Steuergrößen (Input)
   - `rwh`: Hilfsvektor zur Parametrisierung der Steuergrößen (Input)
   - `iwh`: Hilfsvektor zur Parametrisierung der Steuergrößen (Input)
   - `iflag`: Statusflag (Output)

   ```
   subroutine gfcn( t, x, g, p, q, rwh, iwh, iflag )
   ```

   - `g`: Vektor der Funktionswerte von `gfcn` (Output)

```
; vplan.ini

[Aktion]
aktion=Simulationsumgebung

[Pfade]
problempath=phosphin/
inpath=in/
outpath=out/
messpath=mess/
plotpath=plot/
fortranpath=fortran/

[Parameter]
pAnzahl=4
p1=alpha1 1
p2=alpha2 1
p3=ea 0.99768
p4=f1 2.138658

[Versuchsplan]
expAnzahl=2
exp1=experiment.ini exp1.ini
exp2=experiment.ini exp2.ini

[Guetekriterium]
Optimierungskriterium=A

[OptionenVersuchsplanung]
maxit=250
opttol=1e-06
linfeas=1e-07
nlinfeas=0.3
maxitQP=40
realworkspace=30000
integerworkspace=10000

...
```

```
; experiment.ini

[Flags]
switch=1
; Experiment fest oder variabel

[Integrationsintervall]
t0=0
tend=180

[Zustandsvariablen]
; y: differentielle Variablen
yAnzahl=2
y1=n1 na1 -1e+10 1e+10
y2=temp temp0 -1e+10 365.16

; z: algebraische Variablen
zAnzahl=3
z1=n2 0 -1e+10 1e+10
z2=n3 0 -1e+10 1e+10
z3=n4 7.64 -1e+10 1e+10

[Steuergroessen]
qAnzahl=3
q1=na1 3.74 0.8 20 0 -1
q2=temp0 343 323.16 351.16 0 -1
q3=na4 7.64 3 14 0 -1

[Messungen]
tAnzahl=15

t1=12
t1Anzahl=2
t1m1=mfcn1 0.333 1e-06 1
t1m2=mfcn2 0.333 1e-06 1
t1minmax=0 2

...
```

Abbildung 9.1: Exemplarische Ausschnitte aus `vplan.ini` und `experiment.ini`

2. *Meßfunktionen* `mfcn.f` und *Sigmafunktionen* `sigma.f`: Modellantwort für Meßwert und Standardabweichung einer Messung

   ```
   subroutine mfcn( t, x, h, p, q, rwh, iwh, iflag )
   ```

   - `h`: Funktionswert von `mfcn` (Output)

   ```
   subroutine sigma( t, x, s, p, q, rwh, iwh, iflag )
   ```

   - `s`: Funktionswert von `sigma` (Output)

3. *Steuerconstraints* `cfcn.f`

   ```
   subroutine cfcn( c, q, iflag )
   ```

   - `c`: Vektor der Funktionswerte von `cfcn` (Output)

4. *Dynamische Nebenbedingungen* `bfcn.f`

   ```
   subroutine bfcn( t, x, b, p, q, rwh, iwh, iflag )
   ```

   - `b`: Vektor der Funktionswerte von `bfcn` (Output)

5. *Plotfunktion* `plot.f`

   ```
   subroutine plot( t, x, p, q, nga, nt, wron,
   &                ngq, gq, ngaq1, ngaq2, gaq, rwh, iwh )
   ```

   - `nga`: Leading Dimension von `wron` (Input)
   - `nt`: Anzahl Spalten von `wron` (Input)
   - `wron`: Ableitungsmatrix der Zustände nach den Parametern (Input)
   - `ngq`: Leading Dimension von `gq` (Input)
   - `gq`: Ableitungsmatrix der Zustände nach den Steuergrößen (Input)
   - `ngaq1`: Erste Leading Dimension von `gaq` (Input)
   - `ngaq2`: Zweite Leading Dimension von `gaq` (Input)
   - `gaq`: Zweite Ableitungsmatrix der Zustände nach Parametern und Steuergrößen (Input)

Meßdaten- und covmat-Files

Weitere Dateien zur Bereitstellung von Eingaben oder Ausgaben von VPLAN sind

- *Meßdatenfiles* `mess.dat`. In ihnen werden zeilenweise die aufsteigend geordneten Meßzeitpunkte und dahinter jeweils paarweise Meßwert und Standardabweichung abgelegt.
- *Kovarianzmatrix-File* `covmat.mat` zur Ausgabe der Kovarianzmatrix. Für robuste Versuchsplanung, siehe Kapitel 8, wird diese Datei auch als Inputfile verwendet.

9.2.4 Visualisierungsschnittstelle

Das Programmpaket VPLAN stellt eine Schnittstelle zur Verfügung, die die Visualisierung der berechneten Trajektorien ermöglicht. Grundlage dafür ist die kontinuierliche, fehlerkontrollierte Ausgabe durch den Integrator DAESOL, die durch Auswertung des BDF-Polynoms auf einem äquidistanten, benutzerdefinierten Gitter erfolgt, das nicht von der adaptiv gewählten Integratorschrittweite abhängig ist.

Die kontinuierliche Ausgabe wird somit spezifiziert durch Angabe der Ausgabeschrittweite und die Plotfunktion, die auf diesen Gitterpunkten aufgerufen werden soll.

VPLAN erzeugt dann Ausgabefiles, in denen zeilenweise die Zeitpunkte und die Ausgabewerte stehen. Ausgabewerte können Funktionen der Zustandsvariablen, Parameter, Steuergrößen, Steuerfunktionen und der Sensitivitäten sein.

Diese Datenfiles können leicht mit Visualisierungsprogrammen angezeigt werden. Wir stellen ein Interface zur Ausgabe von Mehrfachexperiment-Plots mit GNUPLOT [117] zur Verfügung. Mit dem darauf aufbauenden Tool vplan2html kann diese Ausgabe automatisch in HTML-Seiten eingebunden werden. Abbildung 9.2 zeigt ein Beispiel. Eine grafische Darstellung der Trajektorien ist aber zum Beispiel in einfacher Weise auch mit Matlab [76] möglich.

Eine andere Möglichkeit besteht darin, auch während der Iterationen der Optimierung auf die kontinuierliche Ausgabe zuzugreifen und so Zwischenergebnisse und den Verlauf der Optimierung grafisch darzustellen. Die Kommunikation zwischen vplan98 und dem Visualisierungsprogramm erfolgt dabei über named pipes. Ein Screenshot dieser Online-Visualisierung ist in Abbildung 9.3 dargestellt.

9.3 Struktur der Software

In diesem Abschnitt wollen wir eine Übersicht über die Implementierung und Struktur des Programmpakets VPLAN geben. Diese Struktur besteht zunächst aus der Organisation der Verzeichnisse, in denen Sourcecode, Programm und Anwendungen abgelegt werden. Der Sourcecode selbst ist in Module gegliedert für die verschiedenen methodischen und softwaretechnischen Teilaspekte des Programms. Die Daten des Programms werden in verschiedenen Klassen verwaltet. Schließlich wird der Informationsfluß der Programmausführung durch eine Reihe von Funktionsaufrufen bewerkstelligt.

9.3.1 Verzeichnisstruktur

Die Verzeichnisstruktur einer VPLAN-Installation, siehe Abbildung 9.4, besteht aus drei Unterverzeichnisbäumen unter dem Verzeichnis `vplanroot`. Unter `examples` werden in Problemverzeichnissen die Daten für die Anwendungsprobleme abgelegt. `src` enthält den kompletten Quellcode des Programms. Durch das Build-Tool make wird das Programmpaket kompiliert. Durch Aufruf von doit werden die Problemmodule erzeugt und kompiliert. Der dadurch erzeugte Binary-Code wird im Verzeichnisbaum `target` abgelegt.

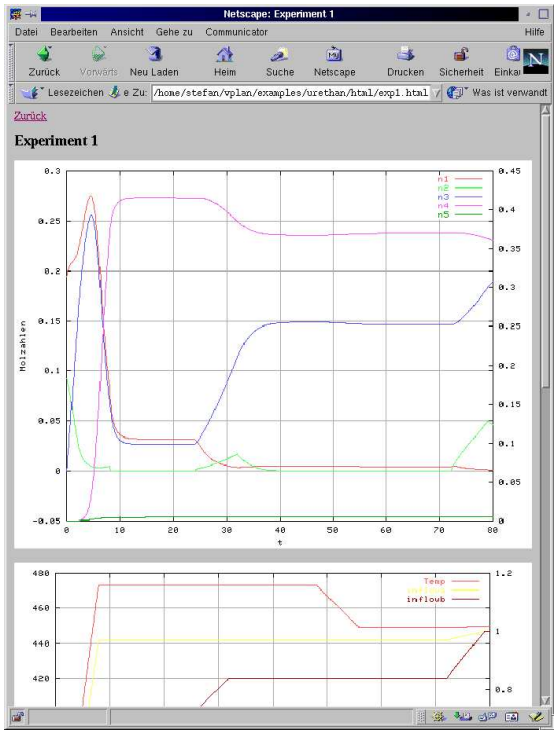


Abbildung 9.2: HTML-Ausgabe der Ergebnisse mit vplan2html

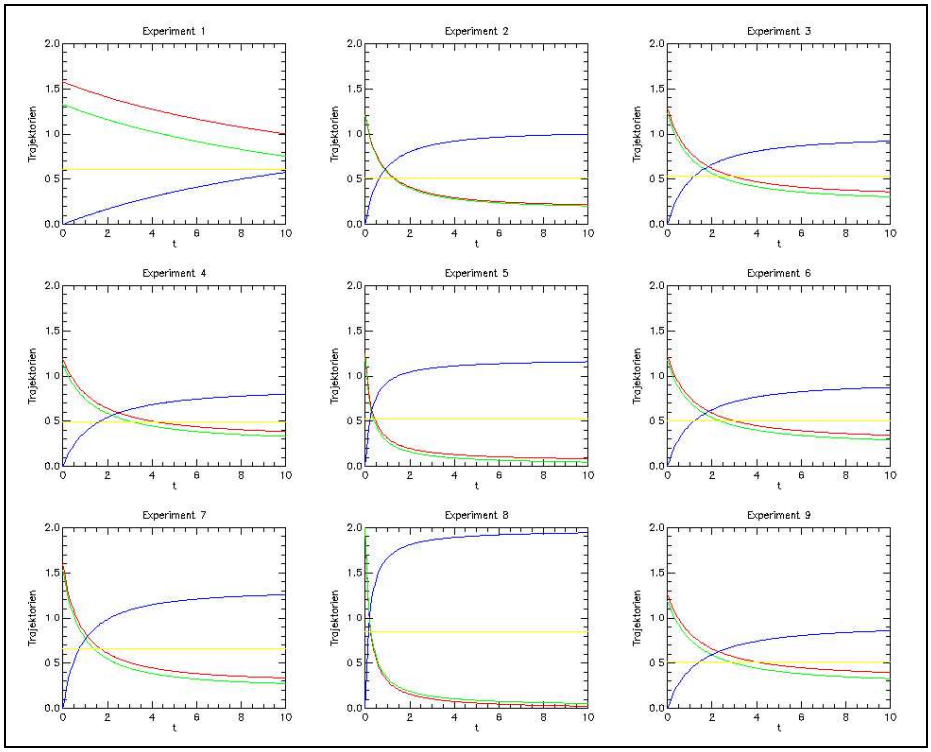


Abbildung 9.3: Screenshot der Online-Visualisierung

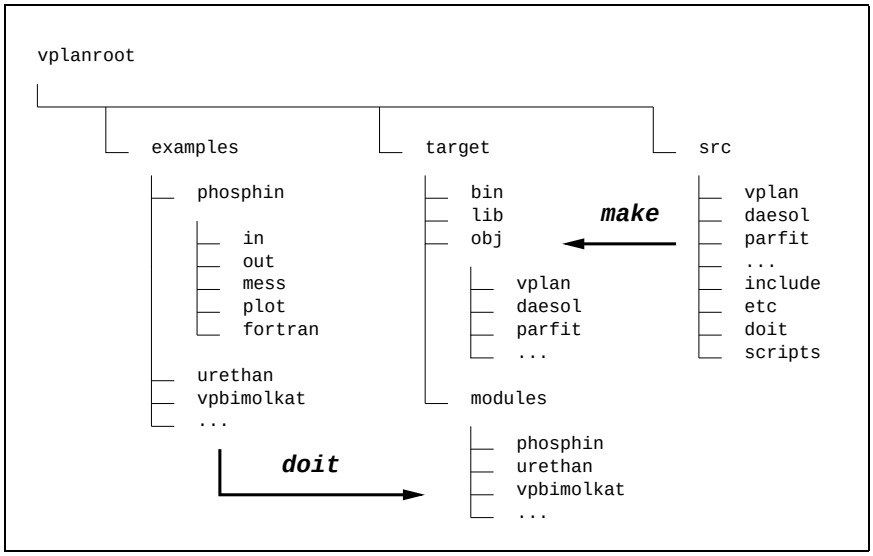


Abbildung 9.4: Verzeichnisstruktur des Softwarepakets VPLAN

Im Teilbaum `src` befinden sich Unterverzeichnisse für die verschiedenen Programmodule und für das Programm doit, Headerdateien in `include` sowie Installationsskripte in `etc`.

Der Teilbaum `examples` enthält Unterverzeichnisse für die einzelnen Anwendungsprobleme. Jedes Problemverzeichnis hat wiederum eine Unterstruktur mit den Verzeichnissen `in` für Input-ini-Files, `out` für Output-ini-Files, `mess` für Meßdatenfiles, `plot` für Output-Plotfiles und `fortran` für Fortran-Files.

Während in `src` und `examples` die plattformunabhängigen ASCII-Files für Sourcecode und Problembeschreibungen stehen, werden unter `target` alle plattformabhängigen, im wesentlichen durch Kompilation erzeugten Binary-Files abgelegt. Das Unterverzeichnis `bin` enthält dabei alle ausführbaren Programme und Skripten des Programmpakets, `lib` die Shared-Library-Files für die Programmodule, `obj` Unterverzeichnisse für die Objekt-Files der Programmodule und `modules` Unterverzeichnisse für die Problemmodule.

9.3.2 Module und Kompilation

Programmodule

Der Code von VPLAN besteht aus mehreren, logisch und softwaretechnisch zusammengehörenden Einheiten. Diese nennen wir Module. Für jedes Modul wird bei der Kompilation eine Shared-Library erzeugt. Jedes Modul selbst setzt sich aus mehreren C++- oder Fortran-Object-Files zusammen.

Im einzelnen gibt es die folgenden Programmodule:

- *vplan:* Hauptprogramm, Datenstrukturen, Funktionen und Methoden für Versuchsplanung und Simulationsumgebung; in C++ implementiert,

- *tools:* Einlese- und Ausgaberoutinen; C++,
- *sqp:* Schnittstelle zu SQP-Solvern; C++,
- *vplpar:* Schnittstelle zu PARFIT und seinen Datenstrukturen; C++,
- *daesol:* Implementierung des Integrators DAESOL; Fortran,
- *daesupp:* Schnittstelle zwischen DAESOL und den Modellfunktionen und deren Ableitungen; C++,
- *parfit*: Mehrfachexperiment-Version des Parameterschätzprogramms PARFIT; Fortran,
- *stat:* Routinen zur Auswertung statistischer Verteilungen; C++,
- *vplnag:* benötigte Routinen der NAG-Bibiothek; Fortran,
- *vplpack:* benötigte Routinen der LAPACK-Bibiothek; Fortran,
- *doit:* Quelltext des Programms doit zur automatischen Ableitungsgenerierung durch Aufruf von ADIFOR und zur Erzeugung der Problemmodule; C++,
- *cov2corr:* Quelltext eines Hilfsprogramms zur Berechnung der Korrelationsmatrix aus der Kovarianzmatrix; C++.

Problemmodule

Für jedes Anwendungsproblem wird durch Aufruf von doit ein Problemmodul erzeugt. Im entsprechenden Unterverzeichnis von `vplanroot/target/modules` wird eine Datei `libproblem.so` angelegt. Diese Bibliothek enthält alle Routinen für das Anwendungsbeispiel und kann zur Laufzeit von vplan98 dynamisch geladen werden, siehe unten. Durch diese Technik wird erreicht, daß Programmcode und Anwendungsbeispiele softwaretechnisch vollständig getrennt sind und erst zur Laufzeit von vplan98 zusammengeführt werden. Dies gibt dem Programm eine sehr weitgehende Flexibilität.

Kompilation

Die Kompilation des Programmpakets erfolgt einfach durch den Aufruf von make. Eine Hierarchie von Makefiles erledigt dabei alle notwendigen Aufgaben. Gegebenenfalls wird das `target`-Verzeichnis angelegt. Die Programmodule werden kompiliert und zu Dynamic Link Libraries (Shared Objects) gelinkt. Das ausführbare Programm `vplan98` kann so sehr kompakt gehalten werden. Alle Programmodule werden beim Programmstart von `vplan98` durch den dynamischen Linker geladen.

Die Makefiles sind getrennt in plattformunabhängige Teile `makefile.all` und plattformabhängige Teile `makefile`. Das Verzeichnis `src/etc` enthält plattformabhängige Makefiles mit speziellen Einstellungen für Rechner mit verschiedenen Betriebssystemen wie Linux,

Solaris, Irix, AIX, Windows NT, Windows 9x. Durch Aufruf eines Skripts update.sh wird jeweils das geeignete `makefile` in den `src`-Unterverzeichnissen installiert. So kann für jede VPLAN-Installation eine optimale Plattform- und Rechneranpassung durchgeführt werden.

Bei der Installation auf den Computern von Anwendern besteht die Möglichkeit, den kompletten Sourcecode nach der Kompilation zu entfernen. Das Programm bleibt dabei selbstverständlich voll lauffähig.

9.3.3 Speicherung der Daten in VPLAN

Die Programmiersprache C++ erlaubt die Definition adäquater Datenstrukturen für Versuchsplan, Experiment und Dynamik. In diesem Abschnitt wollen wir einen Überblick über diese Klassen geben.

- Die Klasse `data` definiert die umfassende Datenstruktur des Programms vplan98.

```
class data
{  ...

   public:
     int Ndynpool;             // Anzahl verschiedener Dynamiken
     DAEdynamics *dynpool;     // Array der Dynamiken

     int Nexpool;              // Anzahl verschiedener Experimentbeschreibungen
     experiment *expool;       // Array der Experimentbeschreibungen

     Mexperiment V;            // Mehrfachexperiment-Versuchsplan

     int NP;                   // Anzahl der Parameter
     Matrix *parray;           // Array der Parameter

     PATHSTR problempath;      // Problemverzeichnis
     PATHSTR inpath;           // Unterverzeichnis fuer Inputdateien
     PATHSTR outpath;          // Unterverzeichnis fuer Outputdateien
     PATHSTR messpath;         // Unterverzeichnis fuer Messdatenfiles
     PATHSTR plotpath;         // Unterverzeichnis fuer Plotfiles
     PATHSTR fortranpath;      // Unterverzeichnis fuer Fortran-Files

     ...

     data( int NP_ = 1, int NPSETS_ = 1,
           int Ndynpool_ = 1, int Nexpool_= 1 );   // Konstruktor
     ~data( void );                                // Destruktor

     int init( int NP_ = 1, int NPSETS_ = 1,
               int Ndynpool_ = 1,
               int Nexpool_= 1 );                  // Initialisierung

     ...
};
```

Von dieser Klasse wird das globale Objekt `db` angelegt. So kann auf die Daten auch unterhalb der Fortran-Funktionsaufrufe zugegriffen werden. Konstruktor und Destruktor von `db` erledigen Aufgaben, die am Programmstart bzw. Programmende durchgeführt werden müssen. Dies ermöglicht eine einfache Fehlerbehandlung: Der Destruktor von `db` wird auch im Fehlerfall am Programmende aufgerufen und kann dann alle Aufräumarbeiten erledigen.

- Die Klasse `Mexperiment` enthält alle Daten zur Beschreibung eines Mehrfachexperiment-Versuchsplans.

```
class Mexperiment
{  ...

   public:
     int Nex;              // Anzahl der Experimente
     experiment **ex;      // Array mit Zeigern auf Experimentbeschreibungen

     PATHSTR *infile;      // Input-ini-Files
     PATHSTR *outfile;     // Output-ini-Files
     PATHSTR *messfile;    // Messdatenfiles
     PATHSTR *pltfile;     // Outputfiles fuer Visualisierung

     plotfuntype *plotfcn; // Plot-Funktionen fuer kontinuierlichen Output
     NAMESTR *plotname;    // Namen
     double *plotstep;     // Schrittlaengen

     int *Nmessvals;       // Anzahl der Messwerte pro Experiment
     double **messvals;    // Messwerte
     double **sigmavals;   // Standardabweichungen der Messwerte

     int *Nmesstimes;      // Anzahl der Messzeitpunkte pro Experiment
     double **messtimes;   // Messzeitpunkte

     char Aktion;          // auszufuehrende Aktion
     char OptKrit;         // Guetekriterium, nach dem optimiert werden soll

     PATHSTR covmat;       // Outputfile fuer die Kovarianzmatrix

                              // Numerische Optionen fuer Versuchsplanung:
     int maxitSQP;            // Maximale Anzahl der SQP-Iterationen
     double opttolSQP;        // Abbruchkriterium

     ...

     Mexperiment( int Nex_ = 1 );       // Konstruktor
     virtual ~Mexperiment( void );      // Destruktor

     virtual int init( int Nex_ );      // Initialisierung

     ...
};
```

- In der Klasse `experiment` sind alle Daten zur Experimentbeschreibung abgelegt.

```
class experiment
```

```
{ ...

  public:
    int swflag;        // Schalter: Experiment festhalten oder nicht

    DAEdynamics *dyn;  // Zugrundeliegende Dynamik

    Matrix x0;         // Feste Werte der Anfangswerte
    Matrix q;          // Versuchsplanungsgroessen
    Matrix w;          // Gewichte

    int NU;            // Anzahl der kontinuierlichen Steuerfunktionen
    int *umode;        // Modus der Diskretisierung

    double *t;         // Zeitpunkte
    int *tflag;        // Typ des Zeitpunkts

    int Nmess;         // Anzahl Messfunktionen
    messtype *mess;    // Messfunktionen

    ...

    experiment( int NQ_ = 1, int NQO_ = 1,
                int NU_ = 1, int NS = 1,
                int Nmess_ = 1 );           // Konstruktor
    virtual ~experiment( void );            // Destruktor

    virtual int init( int NQ_, int NQO_,
                      int NU_, int NS_,
                      int Nmess_,
                      double minmess_,
                      double maxmess_,
                      DAEdynamics *DYN );   // Initialisierung

    ...
};
```

- Die Klasse `dynamic` implementiert die Daten zur Beschreibung jeweils eines dynamischen Modells.

```
class DAEdynamics
{ ...

  public:
    int nvar;                // Anzahl aller Zustandsvariablen
    int ndiff;               // Anzahl der differentiellen Variablen

    ffcndensefuntype FFCN;   // Rechte Seite f der diff. Gleichungen
    gfcndensefuntype GFCN;   // Rechte Seite g der alg. Gleichungen

    x_fgfcntype X_FFCN;      // Ableitungen von ffcn
    p_fgfcntype P_FFCN;
    q_fgfcntype Q_FFCN;
    x_x_fgfcntype X_X_FFCN;
    x_p_fgfcntype X_P_FFCN;
    q_x_fgfcntype Q_X_FFCN;
```

```
    q_p_fgfcntype Q_P_FFCN;

    ...

                            // Numerische Optionen fuer DAESOL:
    double rtol;            // relative Fehlertoleranz
    double atol;            // absolute Fehlertoleranz
    double stepsize;        // Anfangsschrittweite
    int maxorder;           // maximale Ordnung des BDF-Verfahrens
    int maxstepnumber;      // maximale Anzahl von Schritten
    double minstepsize;     // minimale Schrittweite
    double maxstepsize;     // maximale Schrittweite
    int printlevel;         // Outputlevel

    ...

    DAEdynamics( void );                // Konstruktor
    virtual ~DAEdynamics( void );       // Destruktor

    ...

    virtual int integrate( double *pt, double tout,  // Aufruf der Integration
                           Matrix &x, Matrix &p, Matrix &q,
                           int dflag, int startflag, int plotflag,
                           Matrix &GA, Matrix &GAdir,
                           Matrix &GQ, Matrix &GQdir,
                           double *GAQ, int nGAQ1, int nGAQ2,
                           double *rwh, int *iwh );

    ...
};
```

Verschiedene dynamische Modelle und Experimentbeschreibungen werden in den Arrays `expool` und `dynpool` in der Klasse `data` abgespeichert und flexibel durch Zeiger den Experimenten eines Mehrfachexperiment-Versuchsplans zugeordnet, siehe Abbildung 9.5.

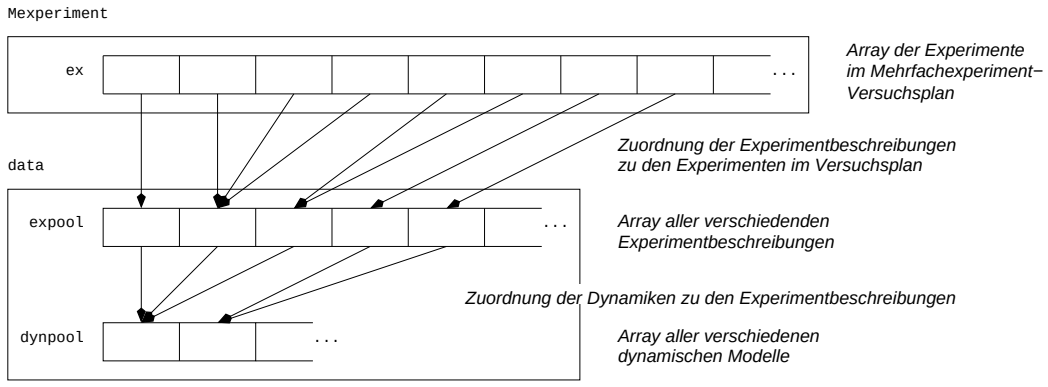


Abbildung 9.5: Datenstrukturen zur flexiblen Speicherung von Experimentbeschreibungen und dynamischen Modellen und deren Zuordung zueinander und zu den Experimenten in einem Versuchsplan

9.3.4 Programmablauf

Der Programmablauf von vplan98 umfaßt die folgenden Schritte:

1. *Programmstart:* Der Konstruktor des globalen Objekts `db` wird ausgeführt, dabei erfolgt die Initialisierung von Errorhandler und Signalhandler und die Belegung des benötigten Speichers.
2. *Initialisierung:* Das `vplan.ini`-File und die `experiment.ini`-Files werden eingelesen. Das zum Anwendungsbeispiel gehörende Problemmodul wird dynamisch geladen, problemabhängige Funktionspointer werden in die Datenstrukturen eingetragen. Gegebenenfalls werden Meßdaten eingelesen.
3. *Berechnungen:* Wahlweise erfolgt
   - der Aufruf der Parameterschätzung,
   - der Aufruf der Integration,
   - der Aufruf der Simulationsumgebung oder
   - der Aufruf der Versuchsplanungsoptimierung.
4. *Ergebnisausgabe:* Die Output-Experiment-Files, das covmat-File und das Output-vplan.ini-File werden geschrieben.
5. *Programmende:* Der Destruktor des globalen Objekts `db` wird ausgeführt. Er gibt den belegten Speicher wieder frei.

Im folgenden sind wichtige Routinen des Programms zusammengestellt.

- `main`: Hauptprogramm
- `lies_vplan`: Einleseroutine für `vplan.ini`-File
- `lies_experimente`: Einleseroutine für `experiment.ini`-Files
- `schreibe_vplan`: Schreibroutine für `vplan.ini`-File
- `schreibe_experimente`: Schreibroutine für `experiment.ini`-Files
- `create_constraints`: Abspeichern der linearen und nichtlinearen Nebenbedingungen
- `make_consistent`: Lösen der linearen Steuerbeschränkungen
- `ableitungstest`: Funktion zur Überprüfung der Ableitungen
- `constraint_check`: Test der Zulässigkeit der Nebenbedingungen
- `Integration`: Aufruf der Integration
- `Simulationsumgebung`: Aufruf der Simulationsumgebung

- `SQP_algorithm`: Schnittstelle zum Optimierungsverfahren
- `pardri`: Schnittstelle zu PARFIT
- `ADE_Guetekriterium`: Zielfunktion für Versuchsplanungsoptimierung
- `ADE_Guetekriterium_robust`: Zielfunktion für robuste Versuchsplanungsoptimierung
- `delta_Zielfunktion`: Zielfunktion für M-Kriterium
- `Guetekriterium_fuer_einen_Parametersatz`: Berechnung des Gütekriteriums für einen Parametersatz
- `confun`: Funktion zur Auswertung der nichtlinearen Nebenbedingungen
- `eval`: Funktion zur Auswertung eines Experiments: Integration des DAE-Systems, wahlweise Auswertung von Meßfunktionen und Nebenbedingungen, wahlweise mit Bereitstellung der benötigten ersten und zweiten Ableitungsinformationen, wahlweise Erzeugung von kontinuierlicher Ausgabe oder Pseudomeßdaten
- `calculate_measnode`: Auswertung eines Meßpunkts
- `calculate_constrnode`: Auswertung eines Nebenbedingungsgitterpunkts
- `daedrid`: Schnittstelle zu DAESOL
- `AOptKrit`: Berechnung des A-Kriteriums
- `deltaAOptKrit`: Berechnung der Ableitungen des A-Kriteriums
- `DOptKrit`: Berechnung des D-Kriteriums
- `deltaDOptKrit`: Berechnung der Ableitungen des D-Kriteriums
- `BeschrCovMat`: Berechnung der Kovarianzmatrix
- `deltaBeschrCovMat`: Berechnung der Ableitungen der Kovarianzmatrix
- `Nullraummethode`: Auswertung der Nullraummethode

9.3.5 Dynamisches Linken von Problemmodulen

Um eine hohe Flexiblität zu erzielen, ist der Programmcode von VPLAN problemunabhängig gehalten. Für die Anwendungsprobleme werden Problemmodule erzeugt, die zur Laufzeit von vplan98 geladen werden. Auf alle problemspezifischen Subroutinen kann damit vom Programm in kompilierter Form zugegriffen werden. Der Benutzer braucht sich um diese Vorgänge nicht zu kümmern, alle notwendigen Schritte werden durch die Angaben in den ini-Files spezifiziert und vom Programm vollautomatisch durchgeführt. Bewirkt wird dies durch den Einsatz von Techniken des dynamischen Linkens, die wir nun genauer beschreiben wollen.

Die Problembeschreibung eines Anwendungsbeispiels ist im jeweiligen Problemverzeichnis abgelegt, siehe Abschnitt 9.2.3. Daraus wird vom Programm doit das Problemmodul erzeugt und in einem Unterverzeichnis von `target/modules` abgespeichert. Das Problemmodul besteht aus allen problemspezifischen Fortran-Subroutinen inklusive der von doit mit Hilfe von ADIFOR erzeugten Ableitungen und zusätzlich aus dem Headerfile `problem.hf` und dem Schnittstellenfile `problem.cc`. Dieses implementiert neben der Treiberroutine `problem_interface` drei Routinen zum Eintragen von Zeigern auf die Fortran-Funktionen in die Programmdatenstrukturen: `init_dyn_functions` für die Modellfunktionen für Differentialgleichungen und algebraische Gleichungen, `init_exp_functions` für Meßfunktionen, Sigmafunktionen, Steuerconstraints und dynamische Nebenbedingungen sowie `init_Mexp_functions` für die Plotfunktionen. Von doit werden alle diese Source-Files kompiliert und die Objekt-Files dann zu der dynamischen Bibliothek `libproblem.so` gelinkt.

Zur Laufzeit von vplan98 wird `libproblem.so` dynamisch geladen. Dies erfolgt, nachdem der Problemname und die Problemdimensionen eingelesen wurden. Dabei wird durch Aufruf der C-Bibliotheksfunktion `dlopen` das Modul geladen:

```
if ( ( db.handle = dlopen( problemlib, RTLD_NOW ) ) == NULL )
   Fehler("Problemmodul kann nicht geladen werden");
```

Durch `dlsym` wird auf die Treiberfunktion `problem_interface` zugegriffen:

```
if ( ( printerface = (PROBLEM_INTERFACE_TYPE)
                     dlsym( db.handle, "problem_interface" ) ) == NULL )
   Fehler("Auf problem_interface kann nicht zugegriffen werden");
```

Der Aufruf von `problem_interface` liefert Zeiger auf die init-Funktionen:

```
init_Mexp_functions = (INIT_MEXP_FUNCTIONS_TYPE) printerface(INIT_MEXP);
init_exp_functions = (INIT_EXP_FUNCTIONS_TYPE) printerface(INIT_EXP);
init_dyn_functions = (INIT_DYN_FUNCTIONS_TYPE) printerface(INIT_DYN);
```

Schließlich werden durch Aufruf der init-Funktionen Zeiger auf die Modellfunktionen in die Datenstrukturen eingetragen:

```
for ( j = 0; j < db.Ndynpool; j++ )
{   db.dynpool[j].init_settings();
    if ( ( errstr = init_dyn_functions( &db.dynpool[j] ) ) != NULL )
       Fehler( errstr );
}

for ( j = 0; j < db.Nexpool; j++ )
    if ( ( errstr = init_exp_functions( &db.expool[j] ) ) != NULL )
       Fehler( errstr );

if ( ( errstr = init_Mexp_functions( &db.V ) ) != NULL )
   Fehler( errstr );
```

Nun können über diese Zeiger die problemspezifischen Routinen aufgerufen werden.

Am Programmende wird mit `dlclose` das Problemmodul entladen:

```
dlclose( handle );
```

9.3.6 Weitere Komponenten der Software

Bei der Implementierung von VPLAN wurden weitere Techniken verwendet, die hier kurz erwähnt werden sollen:

- Die Schnittstellen zu SQP-Software, Parameterschätzprogramm, Integrator und Lineare-Algebra-Bibiotheken sind so modular gestaltet, daß es einfach ist, die verwendeten Bibliotheken durch Alternativen zu ersetzten.
- Ein Errorhandler reagiert auf eventuelle Laufzeitfehler und sorgt für ein kontrolliertes Programmende.
- Ein Signalhandler ermöglicht das vorzeitige Abbrechen der Berechnung mit Zugriff auf die bis dahin erzielten Zwischenergebnisse.
- Zum komfortablen und effizienten Umgang mit Matrizen und Arrays wurden eine Matrixklasse und eine Arrayklasse implementiert, auf deren Objekte mittels Operator Overloading zugegriffen werden kann.

9.4 Werkzeug zur automatischen Ableitungserzeugung

Das Programm doit erzeugt vollautomatisch die für Simulation, Parameterschätzung und Versuchsplanung benötigten Ableitungen aller problemspezifischen Modellfunktionen eines Anwendungsbeispiels. Alle Fortran-Dateien, die gegebenen und die erzeugten, werden gemeinsam mit der Treiberroutine zum Problemmodul `libproblem.so` zusammengelinkt.

Ein Programmlauf von doit führt im wesentlichen die folgenden Arbeiten durch:

1. Aus den ini-Files werden die benötigen Informationen über das Anwendungsbeispiel eingelesen.
2. Wenn noch nicht vorhanden, wird unter `target/modules` das Problemmodulverzeichnis angelegt.
3. Die gegebenenfalls beim letzten Aufruf von doit verwendeten und ins Problemmodulverzeichnis kopierten Fortran-Files werden mit den aktuellen verglichen. Nur bei Änderungen müssen die Ableitungen neu erzeugt werden.

4. Der C-Präprozessor wird auf die Fortran-Files angewendet.
5. Zur Parametrisierung der Steuerfunktionen wurden vom Benutzer DISCRETIZE-Kommandos in die Fortran-Files eingetragen. doit ersetzt diese durch den konkreten Sourcecode für den Zugriff auf die Parametrisierungsvariablen.
6. Für Modell-, Meß- und Constraintfunktionen werden die benötigen Ableitungen erzeugt. Dies geschieht durch Aufruf von ADIFOR. doit generiert alle von ADIFOR benötigten Input- und Treiberfiles. Da ADIFOR zweite Ableitungen nicht direkt bereitstellen kann, generieren wir diese in einer zweistufigen Prozedur: Zunächst erzeugen wir die ersten Ableitungen und leiten diese nochmals mit ADIFOR ab, um die zweiten Ableitungen zu erhalten.
7. doit erzeugt außerdem die Files `problem.cc` und `problem.hf` und ein makefile.
8. Zuletzt ruft doit make zur Erzeugung von `libproblem.so` auf.

Zum Löschen des Problemmodulverzeichnisses steht das Skript undoit zur Verfügung.

9.5 Grafische Benutzeroberflächen

9.5.1 minigui

minigui ist ein kleines, zu Demonstrationszwecken entwickeltes Frontend, implementiert in der objektorientierten Skriptsprache Python [112]. Es ist vor allem zur Vorführung der Simulationsumgebung gedacht. Mit Slidern können die Werte der Steuergrößen interaktiv verändert werden, die Auswirkungen werden auf Knopfdruck angezeigt. Abbildung 9.6 zeigt einen Screenshot.

9.5.2 MGAD

MGAD ist eine unter Betreuung des Autors im Rahmen der Diplomarbeit von Rücker [100] und mehrerer Softwarepraktikumsprojekte entwickelte Java-Oberfläche für VPLAN.

Sie wurde in einer Client-Server-Architektur realisiert.

Auf Client-Seite soll die Benutzeroberfläche plattformunabhängig und sehr flexibel einsetzbar sein. Deshalb wurde sie als Java-Applet implementiert. Damit kann sie aus jedem Java-fähigen, mit dem Internet verbundenen Browser heraus ausgeführt werden ohne weitere Softwareinstallation oder Systemvoraussetzungen.

Die Benutzeroberfläche ist ein grafisches Frontend, implementiert unter Benutzung der Java-Klassenbibliothek Swing [115]. Es ermöglicht die Spezifikation von Modellen für chemische Reaktionssysteme und die Beschreibung chemischer Prozesse. Aus diesen Eingabedaten erzeugt ein Modellgenerator die mathematische Problembeschreibung, also die

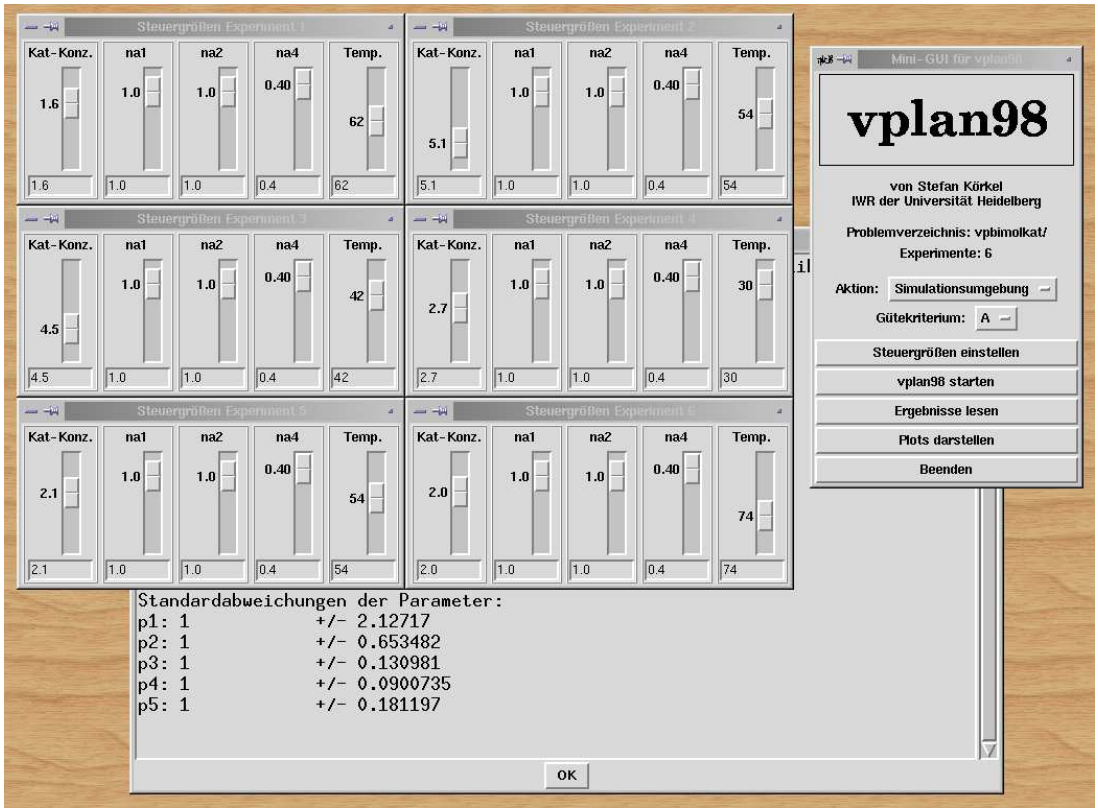


Abbildung 9.6: Screenshot der minigui-Oberfläche für VPLAN

rechten Seiten des DAE-Systems, in Form von Fortran- oder C-Source-Files und generiert mit automatischer Differentiation die Source-Files zur Berechnung der benötigten ersten und zweiten Ableitungen, siehe Abschnitt 6.3. In weiteren Eingabemasken werden die Problemdaten und Werte der Variablen eingegeben. Dann können die Simulations- und Optimierungsmethoden von VPLAN aufgerufen werden.

Die Daten werden nun serialisiert und über das Internet zum Web-Server transferiert, der hier als Application-Server dient. Dort nimmt ein Java-Servlet die Anfragen entgegen, legt ein Problemverzeichnis und darin die problemspezifischen ini- und Fortran-Files an und ruft doit zur Erzeugung des Problemmoduls und vplan98 zur numerischen Berechnung auf. Der Client kann per Statusanfrage feststellen, ob die Berechnung beendet ist (Polling), und dann die Ergebnisdaten abholen. Zu deren Visualisierung stellt die Benutzeroberfläche Werkzeuge bereit.

Den Ablauf der Kommunikation zwischen Client und Server veranschaulicht Abbildung 9.7.

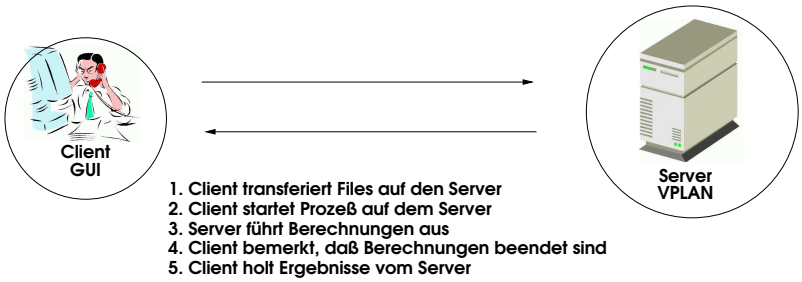
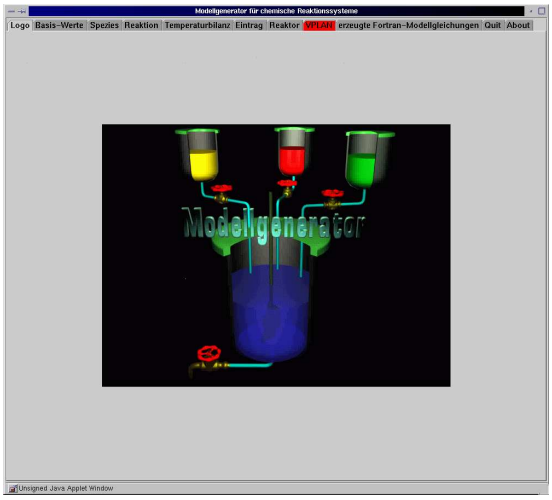
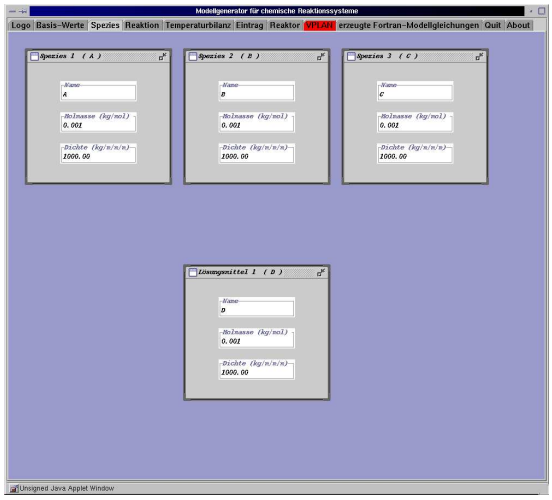


Abbildung 9.7: Kommunikation zwischen Client und Server

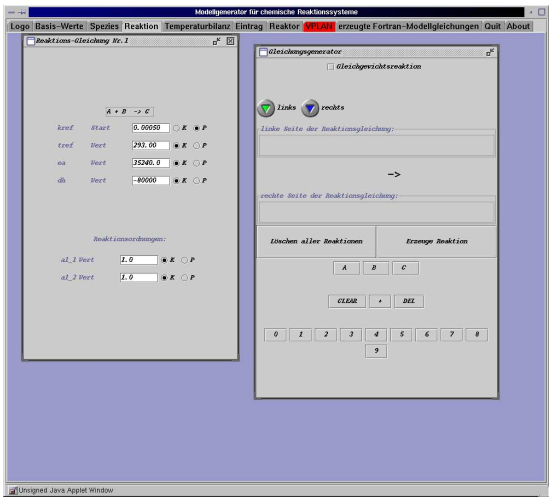
Abbildung 9.8 zeigt Screenshots der grafischen Benutzeroberfläche von MGAD.



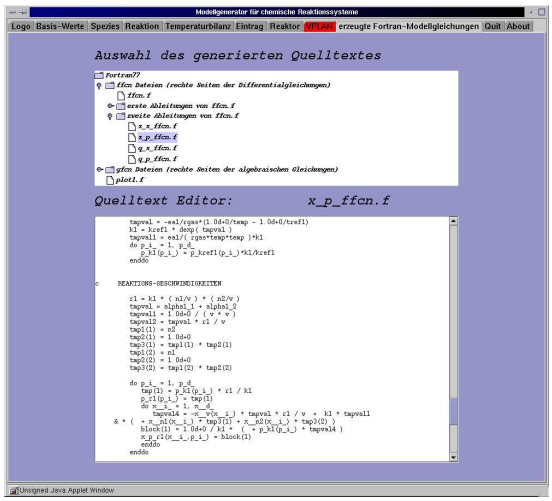
Titelseite



Spezifikation der chemischen Spezies



Eingabe der chemischen Reaktionen



Anzeige der erzeugten Source-Files

Abbildung 9.8: Screenshots der MGAD-Oberfläche für VPLAN

Kapitel 10

Anwendungen und Numerische Ergebnisse

In diesem Kapitel wollen wir anhand ausgewählter Beispiele den Einsatz der Methoden der Nichtlinearen Optimalen Versuchsplanung bei Anwendungen der chemischen Reaktionskinetik und Verfahrenstechnik demonstrieren.

Die Modellierung der chemischen Reaktionssysteme erfolgte anhand der Prinzipien, die wir in Kapitel 2 behandelt haben. Alle numerischen Rechnungen wurden mit dem Programm VPLAN, siehe Kapitel 9, durchgeführt. So schließt sich hier der Kreis dieser Arbeit, von der Modellierung über die mathematischen und numerischen Methoden bis hin zur Entwicklung eines Softwarepakets und dessen Verwendung.

Als erstes Beispiel, das begleitend zur Softwareentwicklung untersucht wurde, betrachten wir die **Phosphinreaktion**, eine einfache chemische Reaktion, deren Modellierung aber bereits die typischen Schwierigkeiten, ein nichtlineares Modell sowie eine Zustandsbeschränkung, beinhaltet. Der Entwurf eines Mehrfachexperiments durch sequentielle Versuchsplanung und Parameterschätzung wird hier mit einer intuitiven Versuchsplanung durch einen erfahrenen Experimentator verglichen. Dabei zeigt sich, daß sich durch die Optimierung ein deutlicher Vorteil bei der Güte der Parameterschätzungen erzielen läßt.

Bei der **Urethanreaktion** liegt ein wesentlich komplexeres Reaktionssystem vor. Die Formulierung des Modells berücksichtigt hier viele Aspekte der praktischen Realisierung der Experimente im Labor. Wieder wurde ein Vergleich zwischen intuitiver und optimaler Versuchsplanung durchgeführt. Die Ergebnisse zeigen die enorme Einsparmöglichkeit an experimentellen Ressourcen durch den Einsatz der Optimierungsmethoden: man benötigt nur zwei statt fünfzehn Experimente bei gleichzeitig deutlich besserer Schätzgüte.

Anschließend stellen wir die Anwendung der Methodik auf eine **Veresterungsreaktion** vor, ein Beispiel aus der industriellen Praxis. Es handelt sich dabei um ein vierphasiges Reaktionssystem mit einem komplexen Modell. Ziel ist die Validierung des Modells, um anschließend dieses Reaktionssystem in die Entwicklung eines industriellen Produkts einbringen zu können. Nach optimaler Versuchsplanung ist die Schätzung aller Parameter mit genügend hoher Genauigkeit möglich.

Schließlich wollen wir anhand einer **Bimolekularen Katalyse** eine Studie zur robusten Versuchsplanung vorstellen. Die Ergebnisse zeigen hier, daß die Empfindlichkeit des optimalen Versuchsplans gegenüber Störungen der Parameter deutlich verringert werden kann.

10.1 Die Phosphinreaktion

Die Phosphinreaktion wurde im Rahmen dieser Arbeit als erstes Anwendungsbeispiel untersucht, begleitend zur Softwareentwicklung. Es handelt sich dabei um ein einfaches chemisches Reaktionssystem, dessen Modellierung aber trotzdem schon die typischen Schwierigkeiten aufweist: Die Gleichungen des DAE-Systems hängen nichtlinear von den Modellparametern ab. Optimiert werden sollen Meßlayout und Steuerungen, insbesondere auch zeitabhängige Steuerfunktionen. Zudem ist eine Zustandsbeschränkung gegeben.

Ziel der Untersuchungen war es, die Funktionsweise der Methodik zu erproben und dabei die Softwareentwicklung voranzutreiben. In sequentieller Vorgehensweise aus wiederholter optimaler Versuchsplanung, Experimentdurchführung und Parameterschätzung wurde ein Mehrfachexperiment entworfen, um die Parameter des Modells, nämlich Frequenzfaktor, Aktivierungsenergie und Reaktionsordnungen, möglichst genau zu schätzen.

Um die Leistungsfähigkeit der optimalen Versuchsplanung beurteilen zu können, wurde diese Versuchsreihe mit einem von einem Chemiker der BASF intuitiv erstellten Versuchsplan verglichen. Dabei zeigt sich bereits bei diesem ersten Beispiel, daß durch den Einsatz der Optimierung eine signifikante Verbesserung der Schätzgüte bei gleichem experimentellem Aufwand erreicht werden kann.

Die experimentellen Daten für dieses Anwendungsbeipiel wurden von der BASF geliefert.

10.1.1 Chemische Reaktion und Prozeß

Bei der Phosphinreaktion handelt es sich um eine Reaktion, die im Hauptlabor der BASF untersucht wurde [18]:

$$\underset{\text{Phosphin}}{A} + \underset{\text{Halogenid}}{B} \rightarrow \underset{\text{Phosphoniumsalz}}{C}$$

in einem niedersiedenden Lösungsmittel L.

Da die Reaktion stark exotherm ist, muß sie aus technischen Gründen bei hohen Konzentrationen durchgeführt werden. Eine isotherme Reaktionsführung ist nicht oder nur sehr schwer möglich. Der verwendete Reaktor ist ein Semi-Batch-Reaktor mit einem Zulauf und Kühl- bzw. Heizvorichtung, siehe Abbildung 10.1. In der Vorlage befindet sich zu Beginn der Reaktion Stoff A, gelöst in L, im Zulauf wird B in L zugegeben. Damit kein Phasenübergang stattfindet, darf die Temperatur des Reaktionsgemischs die Siedetemperatur des Lösungsmittels nicht überschreiten.

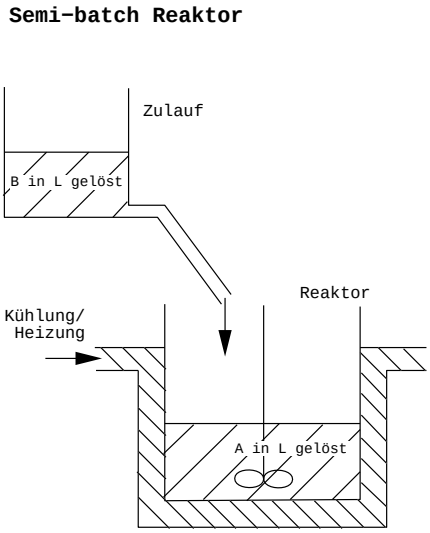


Abbildung 10.1: Reaktor der Phosphinreaktion

10.1.2 Mathematisches Modell

Die Stöchiometriematrix des Reaktionssystems lautet

$$\nu = \begin{pmatrix} -1 & -1 & 1 & 0 \end{pmatrix}.$$

Ihr Rang ist 1, also kann die Reaktion durch ein DAE-System mit einer Differentialgleichung und drei algebraischen Gleichungen beschrieben werden. Zustandsvariablen sind die Molzahlen der Spezies inklusive Lösungsmittel n_1, n_2, n_3 und n_4.

Wir nehmen ein Geschwindigkeitsgesetz mit Arrhenius-Kinetik an:

$$\dot{n}_1 = -V^{1-\alpha_1-\alpha_2} \cdot f \cdot \exp\left(-\frac{10^5 \cdot E_a}{R} \cdot \left(\frac{1}{T} - \frac{1}{T_{ref}}\right)\right) \cdot n_1^{\alpha_1} \cdot n_2^{\alpha_2}.$$

Die Anfangsbedingung lautet

$$n_1(t_0) = n_{1,0},$$

die Experimentdauer ist

$$t \in [t_0; t_{end}] = [0, 180].$$

Eine mögliche Bilanzmatrix des Systems, vergleiche Abschnitt 2.6.2, ist

$$\epsilon = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Aus der Teilchenerhaltung $\epsilon \cdot \Delta n = 0$

$$\begin{aligned} \Delta n_2 + \Delta n_3 &= 0 \\ \Delta n_1 + \Delta n_3 &= 0 \\ \Delta n_4 &= 0 \end{aligned}$$

folgen mit den Molzahländerungen

$$\begin{aligned}\Delta n_1 &= n_1 - n_{1,0}\\ \Delta n_2 &= n_2 - n_{2,e}\\ \Delta n_3 &= n_3\\ \Delta n_4 &= n_4 - n_{4,0} - n_{4,e}\end{aligned}$$

die drei algebraischen Gleichungen

$$\begin{aligned}n_2 &= -n_3 + n_{2,e}\\ n_3 &= -n_1 + n_{1,0}\\ n_4 &= n_{4,0} + n_{4,e}.\end{aligned}$$

Den Zulauf modellieren wir über das Zulaufprofil $feed(t) \in [0,1]$ mit den Randbedingungen $feed(t_0) = 0$ und $feed(t_{end}) = 1$ und nichtnegativer Steigung $\dot{feed}(t) \geq 0$:

$$\begin{aligned}n_{2,e} &= n_{2,e,0} \cdot feed\\ n_{4,e} &= n_{4,e,0} \cdot feed\\ \dot{m}_e &= (M_2 \cdot n_{2,e,0} + M_4 \cdot n_{4,e,0}) \cdot \dot{feed}.\end{aligned}$$

Masse und Volumen des Reaktionsgemischs sind

$$\begin{aligned}m &= M_1 \cdot n_1 + M_2 \cdot n_2 + M_3 \cdot n_3 + M_4 \cdot n_4\\ V &= m/\rho.\end{aligned}$$

Die Innentemperatur des Reaktionsgemischs wird beeinflußt durch die Reaktionswärme, die Temperatur des Zulaufs und die Heizung bzw. Kühlung. Die Wärmebilanz ergibt die zusätzliche Differentialgleichung

$$\begin{aligned}\dot{T} = \Bigg(&\underbrace{-\Delta H \cdot V^{1-\alpha_1-\alpha_2} \cdot f \cdot \exp\left(-\frac{10^5 \cdot E_a}{R} \cdot \left(\frac{1}{T} - \frac{1}{T_{ref}}\right)\right) \cdot n_1^{\alpha_1} \cdot n_2^{\alpha_2}}_{\text{Reaktion}}\\ + &\underbrace{\dot{m}_e \cdot c_p \cdot (T_e - T)}_{\text{Zulauf}} + \underbrace{k_W \cdot A_W \cdot (T_W - T)}_{\text{Kühlung}}\Bigg)/(c_p \cdot m)\end{aligned}$$

mit dem Anfangswert

$$T(t_0) = T_0.$$

Wärmeaustauschfläche und -temperatur werden dabei beschrieben durch

$$\begin{aligned}A_W &= \pi \cdot d \cdot h + \pi/4 \cdot d^2\\ h &= 0.004 \cdot V/(d^2 \cdot \pi)\\ T_W &= \frac{k_W \cdot A_W \cdot T + 2 \cdot \dot{V}_W \cdot \rho_W \cdot c_W \cdot T_{in}}{k_W \cdot A_W + 2 \cdot \dot{V}_W \cdot \rho_W \cdot c_W}.\end{aligned}$$

Die auftretenden Größen in diesem Modell können nun folgendermaßen klassifiziert werden:

Zustandsvariablen des DAE-Modells sind die

- Molzahlen n_1, n_2, n_3, n_4 und die
- Temperatur T.

Die unbekannten und zu bestimmenden **Parameter** sind

- Reaktionsordnung α_1,
- Reaktionsordnung α_2,
- Aktivierungsenergie E_a und
- Frequenzfaktor f.

Als **Steuergrößen** treten auf

- die Anfangsmolzahlen für die Einwaage in den Reaktor von Spezies A $n_{1,0} \in [0.8, 20]$ und Lösungsmittel L $n_{4,0} \in [3, 14]$
- und die Anfangstemperatur $T_0 \in [323.16, 351.16]$

und als zeitlich veränderbare **Steuerfunktionen**

- das Zulaufprofil $feed(t) \in [0, 1]$ mit $feed(t_0) = 0$, $feed(t_{end}) = 1$ und $\dot{feed}(t) \geq 0$
- sowie die Kühltemperatur $T_{in}(t) \in [273.16, 393.16]$ mit $\dot{T}_{in}(t) \in [-2, 2]$.

Schließlich sind die folgenden Größen **Konstanten**:

- Zulaufmengen $n_{2,e,0} = 4$ und $n_{4,e,0} = 8$,
- Temperatur des Zulaufs $T_e = 293.16$,
- Molmassen M_1, M_2, M_3, M_4,
- Dichte und Wärmekapazität des Reaktionsgemischs $\rho = 1000$, $c_p = 2$,
- Reaktionsenthalpie ΔH,
- Referenztemperatur $T_{ref} = 365.16$,
- Dichte und Wärmekapazität des Kühlmediums $\rho_W = 900$, $c_W = 0.12$,
- Wärmedurchgangskoeffizient $k_W = 1600$,

- Reaktordurchmesser $d = 0.15$ und
- Durchflußrate des Kühlmittels $\dot{V}_W = 1$.

Alle Einheiten sind konsistent gewählt und werden daher hier weggelassen.

Aufgrund von Geheimhaltungsvereinbarungen mit der BASF AG können die Zahlenwerte der Molmassen und der Reaktionsenthalpie nicht veröffentlicht werden.

10.1.3 Zustandsbeschränkung und Messungen

Da in den Experimenten kein Phasenübergang stattfinden soll, darf das Reaktionsgemisch nicht sieden. Durch die Siedetemperatur des Lösungsmittels ist somit eine Zustandsbeschränkung gegeben:

$$T(t) \leq 365.16.$$

Die folgenden Meßverfahren stehen zur Verfügung:

1. Molzahlmessung von n_3 mit Standardabweichung des Meßfehlers 0.01 und
2. Temperaturmessung T mit Standardabweichung des Meßfehlers 0.5.

Es können pro Experiment 5 Probenahmen und 5 Temperaturmessungen durchgeführt werden. Als Gitter der möglichen Meßzeitpunkte verwenden wir 15 äquidistante Zeitpunkte.

10.1.4 Versuchsplanung

Zur möglichst genauen Bestimmung der vier Modellparameter sollten insgesamt fünf Experimente durchgeführt werden. In einer Art Wettbewerb wurde einerseits von einem Chemiker aus dem Hauptlabor der BASF mit langjähriger experimenteller Erfahrung auf intuitive Weise ein Versuchsplan aufgestellt. Dies wurde andererseits verglichen mit der methodengestützten optimalen Versuchsplanung unter Anwendung der Software VPLAN.

Im folgenden stellen wir die Ergebnisse der beiden Vorgehensweisen vor.

Als Startwerte für die Parameter wurden Literaturwerte angenommen:

$$\begin{aligned} \alpha_1 &= 1.0 \\ \alpha_2 &= 1.0 \\ E_a &= 0.99768 \\ f &= 2.138658. \end{aligned}$$

Intuitive Versuchsplanung

Der Chemiker schlug zunächst drei Experimente vor. Mit den daraus ermittelten Daten wurde eine Parameterschätzung durchgeführt. Dann plante er weitere zwei Experimente.

Die Wahl der Steuerungen und Messungen für den intuitiven Versuchsplan ist in Tabelle 10.1 zusammengestellt.

Experiment	$n_{1,0}$	$n_{4,0}$	T_0	Probenahmen	Temperaturmessungen
1	0.8	3	323.16	5, 10, 15, 20, 40	5, 10, 15, 20, 40
2	4	3	353.16	5, 10, 15, 25, 40	5, 10, 15, 25, 40
3	4	6	353.16	5, 10, 15, 25, 40	5, 10, 15, 25, 40
4	5	3	333.16	20, 40, 80, 120, 160	20, 40, 80, 120, 160
5	3	6	353.16	20, 40, 80, 120, 160	20, 40, 80, 120, 160

Tabelle 10.1: Intuitive Experimentvorschläge für die Phosphinreaktion

Die Steuerfunktionen wurden folgendermaßen gewählt: Die gesamte Zulaufmasse wird im Zeitraum von t_0 bis t_e mit konstanter Zulaufrate zugegeben. Die Temperatursteuerung erfolgt mittels eines Kühlungsdreiecks, von t_0 bis t_1 wird die T_{in}-Temperatur mit konstanter Rate geändert und von t_1 bis t_{end} wieder auf den Ausgangswert zurückgefahren. Die entsprechenden Werte für die fünf Experimente zeigt Tabelle 10.2, exemplarische Profile sind in Abbildung 10.2 dargestellt.

Experiment	$T_{in}(t_0)$	t_1	$T_{in}(t_1)$	$T_{in}(t_{end})$	t_e
1	303.16	15	273.16	303.16	15
2	273.16	-	273.16	273.16	30
3	273.16	-	273.16	273.16	30
4	320.16	50	295.16	320.16	4
5	333.16	30	303.16	333.16	4

Tabelle 10.2: Steuerfunktionen der intuitiven Experimentvorschläge für die Phosphinreaktion

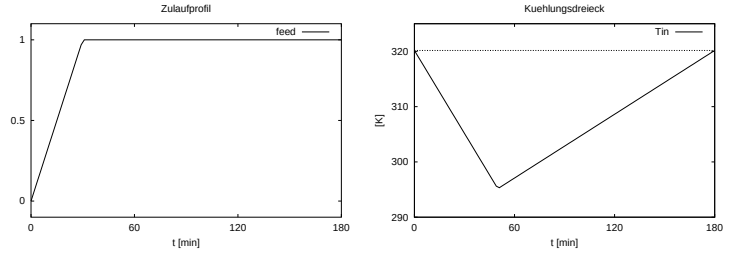


Abbildung 10.2: Exemplarische Profile der Steuerfunktionen der intuitiv geplanten Experimente für die Phosphinreaktion

Parameterschätzungen wurden für die ersten drei bzw. alle fünf Experimente durchgeführt, siehe Tabelle 10.3. Am Ende konnten die Parameter mit Standardabweichungen

von höchstens 2.2% geschätzt werden. Die Verbesserung der Schätzgüten ist in Diagramm 10.3 dargestellt.

Par.	Exp. 3			Exp. 5		
α_1	0.84	$\pm$	0.05	0.79	$\pm$	0.02
α_2	1.15	$\pm$	0.07	1.19	$\pm$	0.01
E_a	1.02	$\pm$	0.04	1.037	$\pm$	0.005
f	0.36	$\pm$	0.06	0.406	$\pm$	0.006

Tabelle 10.3: Parameterschätzungen für die intuitiv geplanten Experimente für die Phosphinreaktion nach drei und fünf Experimenten

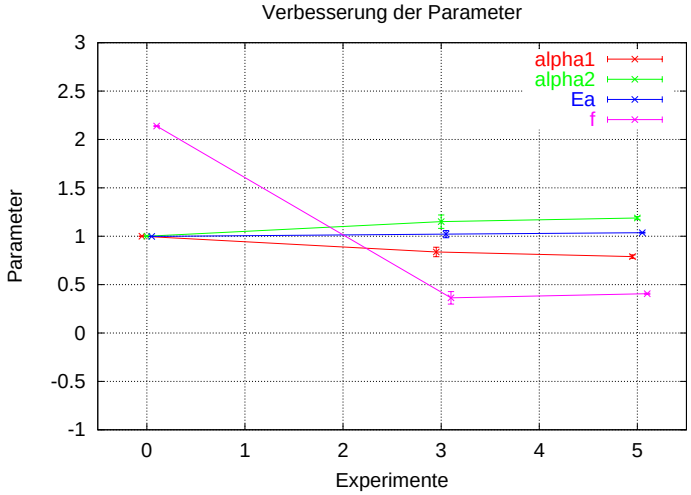


Abbildung 10.3: Geschätzte Parameter und Standardabweichungen bei der intuitiven Versuchsplanung für die Phosphinreaktion

Optimale Versuchsplanung

Die optimale Versuchsplanung wurde in sequentieller Vorgehensweise realisiert. Mit unserer Software VPLAN wurde dabei sukzessive jeweils ein zusätzliches Experiment geplant. Für dieses wurden Meßdaten erhoben und daraus in einer Parameterschätzung ein neuer Schätzer für den Parametervektor berechnet. Die so gewonnenen Informationen wurden dann bei der nächsten Versuchsplanung berücksichtigt.

Die Versuchsplanungs-Optimierungsprobleme, die zu lösen waren, hatten jeweils 61 Variablen für das neue, variable Experiment, nämlich 3 Steuergrößen, 28 Variablen für die Parametrisierung der Steuerfunktionen und 30 Meßgewichte. Als Zielfunktion der Optimierung wurde das D-Kriterium verwendet.

Tabelle 10.4 zeigt die Steuerungen und Messungen der fünf optimierten Experimente. Steuerfunktionen und Trajektorien sind in Abbildung 10.4 dargestellt.

Die Lösungen für die Gewichte zur Plazierung der Messungen erfüllten automatisch die Ganzzahligkeitsbedingungen, so daß keine Rundungsheuristiken eingesetzt werden mußten.

Experiment	$n_{1,0}$	$n_{4,0}$	T_0	Probenahmen	Temperaturmessungen
1	4.001	9.762	347.58	12, 24, 36, 168, 180	12, 24, 36, 48, 60
2	4.227	3	351.16	24, 36, 48, 168, 180	12, 24, 36, 48, 60
3	9.220	3	346.40	132, 144, 156, 168, 180	72, 84, 96, 108, 120
4	4.052	3	351.16	60, 72, 84, 96, 108	36, 48, 60, 72, 84
5	3.788	3	323.16	60, 72, 84, 96, 108	48, 60, 72, 84, 96

Tabelle 10.4: Steuerungen der optimierten Experimente für die Phosphinreaktion

Die Ergebnisse der Parameterschätzungen sind in Tabelle 10.5 zusammengestellt. Die maximale Standardabweichung nach 5 Experimenten beträgt 0.27%. Die Verbesserung der Parameter bei der sequentiellen Vorgehensweise zeigt Abbildung 10.5, die Verbesserung der Gütekriterien Abbildung 10.6.

Par.	Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5
α_1	0.9 ± 0.6	0.794 ± 0.005	0.795 ± 0.005	0.795 ± 0.002	0.795 ± 0.002
α_2	1.3 ± 0.5	1.20 ± 0.02	1.199 ± 0.008	1.199 ± 0.002	1.199 ± 0.002
E_a	1.1 ± 0.2	1.040 ± 0.006	1.0381 ± 0.0009	1.0381 ± 0.0008	1.0381 ± 0.0007
f	0.5 ± 0.5	0.41 ± 0.01	0.407 ± 0.001	0.407 ± 0.001	0.4075 ± 0.0009

Tabelle 10.5: Parameterschätzungen für die optimierten Experimente für die Phosphinreaktion nach 1, 2, 3, 4 und 5 Experimenten

10.1.5 Vergleich und Fazit

Obwohl es sich bei der Phosphinreaktion um ein sehr einfaches chemisches Reaktionssystem handelt, dessen Verhalten noch intuitiv erfaßbar scheint, kann mit Hilfe von Optimierungsmethoden trotzdem eine deutliche Verbesserung der Schätzgenauigkeit erzielt werden. Die Parameter können bei gleichem experimentellen Aufwand um eine Stelle genauer ermittelt werden. Das nächste Beispiel zeigt, daß bei komplexeren Systemen dieser Vorteil noch viel deutlicher ausfällt.

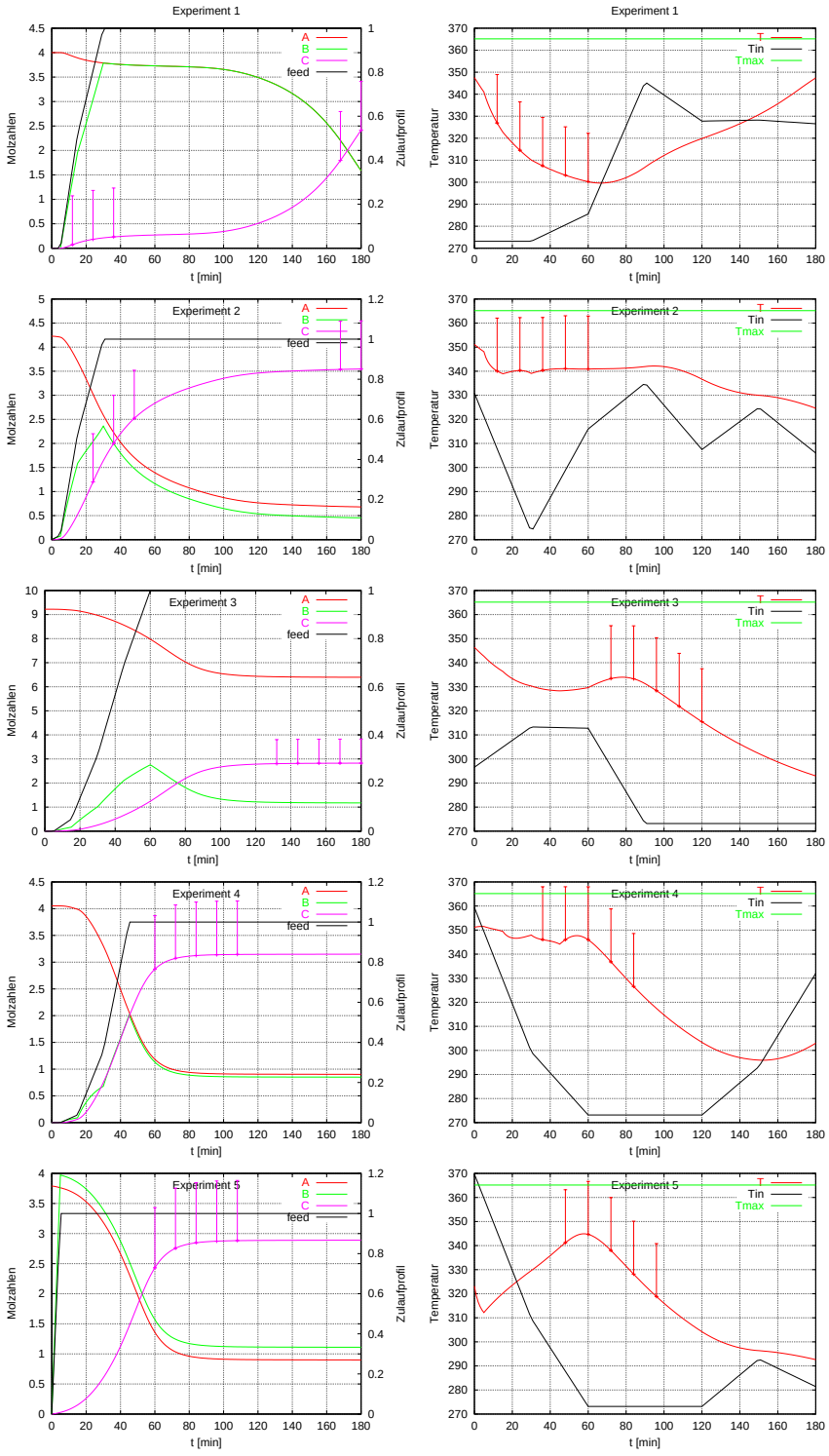


Abbildung 10.4: Trajektorien und Steuerfunktionen der fünf Experimente der sequentiellen Versuchsplanung für die Phosphinreaktion. Die Balken auf den Trajektorien geben die Probenahmen und Temperaturmessungen an.

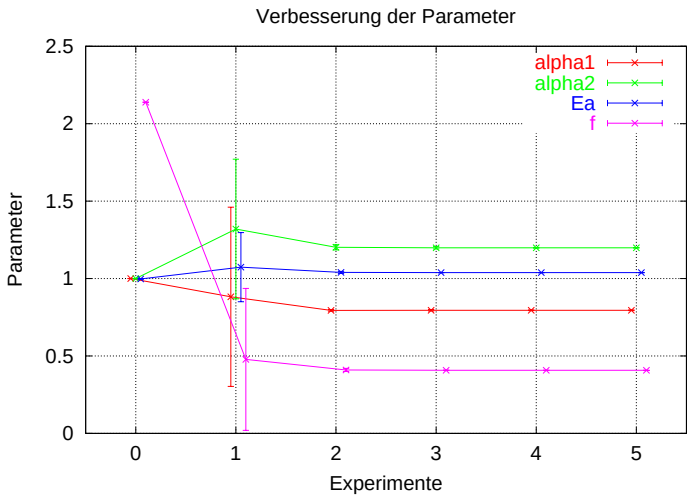


Abbildung 10.5: Geschätzte Parameter und Standardabweichungen bei der sequentiellen Versuchsplanung für die Phosphinreaktion

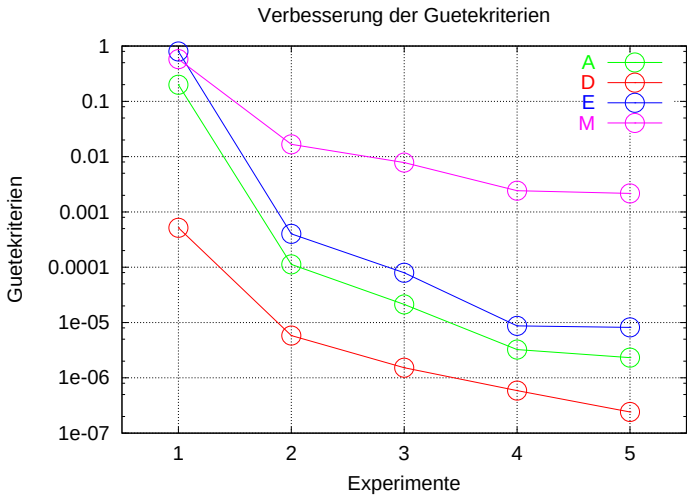


Abbildung 10.6: Verbesserung der Gütekriterien bei der sequentiellen Versuchsplanung für die Phosphinreaktion. Optimierungskriterium ist das D-Kriterium.

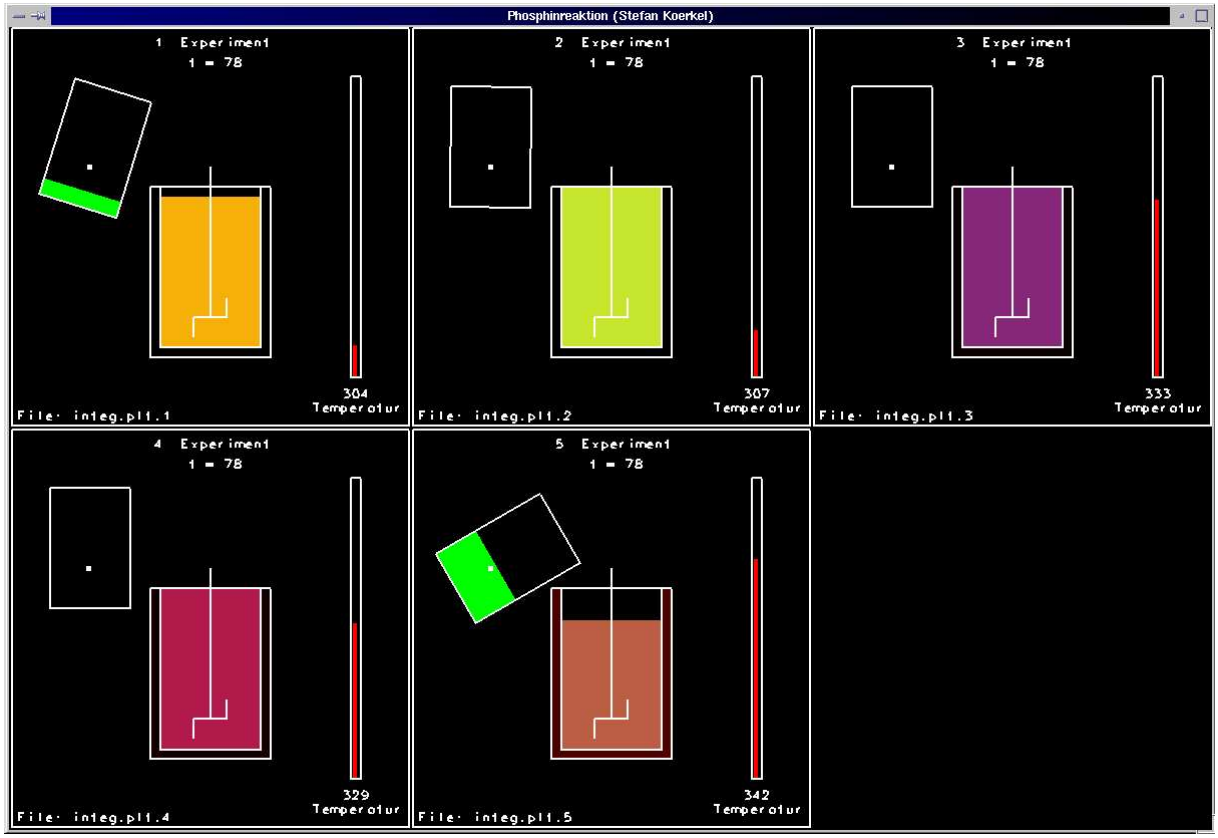


Abbildung 10.7: Screenshot der Open-GL-Visualisierung der Phosphinreaktion

10.2 Die Urethanreaktion

Das in diesem Anwendungsbeispiel betrachtete Reaktionssystem spielt eine wichtige Rolle bei der Herstellung von Polyurethan-Kunststoffen. Polyurethane finden Verwendung als Schaum, zum Beispiel für Polstermöbel oder Schwämme, in der Wärmedämmung oder als Verpackungsmaterial. Darüber hinaus können sie als Lacke, Klebstoffe, Elastomere, thermoplastische Kunststoffe oder Fasern eingesetzt werden. Hergestellt werden Polyurethane durch Polyadditionsreaktionen, ausgehend von Diisocyanaten und Dialkoholen [51].

Um das Verhalten der bei der Polyurethanherstellung ablaufenden Prozesse besser verstehen zu können, betrachtet man zunächst stattdessen die Reaktion von (Mono-)Isocyanaten und (Mono-)Alkoholen [72], bei der keine Kettenbildung stattfindet. Dabei entsteht ein Urethan. In einer reversiblen Nebenreaktion reagiert außerdem das Isocyanat mit dem Urethan zu einem Allophanat. Je nach verwendetem Katalysator und Reaktionsbedingungen kann das Isocyanat zu einem Isocyanurat trimerisieren. Dieses Reaktionssystem soll hier untersucht werden.

Die Eigenschaften der Polyurethane hängen unter anderem davon ab, wie stark bei ihrer Herstellung solche Nebenreaktionen ablaufen. Um dies zu quantifizieren, müssen die kinetischen Parameter möglichst genau bestimmt werden. Im Vergleich zur Phosphinreaktion ist die Urethanreaktion wesentlich komplexer, ihr Verhalten ist viel schwerer intuitiv erfaßbar. Auch hier wird wieder ein Vergleich zwischen intuitiver einerseits und methodengestützter optimaler Versuchsplanung andererseits durchgeführt.

Bei der Modellierung wurde darauf geachtet, daß durch die Nebenbedingungen an die Versuchsplanungsgrößen viele Gegebenheiten des Laboralltags berücksichtigt werden. Die Möglichkeit der Behandlung solcher Aspekte ist wichtig im Hinblick auf den Einsatz des Softwarepakets VPLAN in der industriellen Praxis.

Die experimentellen Daten wurden auch hier von der BASF zur Verfügung gestellt.

10.2.1 Reaktion

Bei der Urethanreaktion handelt es sich um eine Simultan- und Konsekutivreaktion mit einem chemischen Gleichgewicht. Solche Reaktionen kommen in der chemischen Reaktionskinetik häufig vor.

Die Urethanreaktion hat das folgende Reaktionsschema:

$$\begin{array}{rcl} A + B & \rightarrow & C \\ A + C & \rightleftharpoons & D \\ 3A & \rightarrow & E \end{array}$$

Edukte sind Phenylisocyanat A und Butanol B im Lösungsmittel Dimethylsulfoxid L. Gebildet werden das Wertprodukt Urethan C, Allophanat D als Folgeprodukt und das Nebenprodukt Isocyanurat E.

10.2.2 Reaktor und Betriebsweise

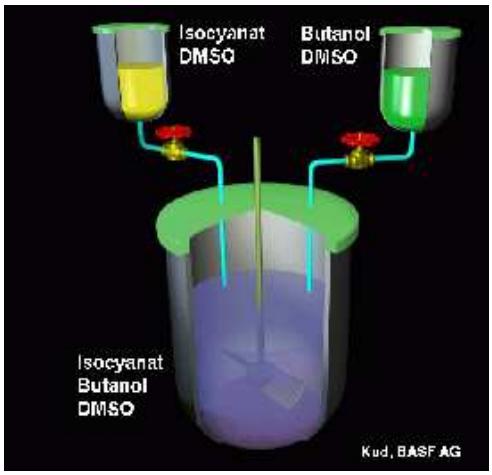


Abbildung 10.8: Darstellung des Reaktors für die Urethanreaktion

Der Reaktor für die Urethanreaktion ist ein Rührkessel und kann als Batch oder Semi-Batch mit bis zu zwei Einträgen betrieben werden. Im Reaktor können Phenylisocyanat und Butanol im Lösungsmittel Dimethylsulfoxid vorgelegt werden. In Eintrag 1 kann Phenylisocyanat in Dimethylsulfoxid, in Eintrag 2 Butanol in Dimethylsulfoxid zugeführt werden. Die Innentemperatur des Reaktors ist steuerbar.

10.2.3 Mathematisches Modell

Zur Aufstellung des mathematischen Modells gehen wir aus von der ν-Matrix der Stöchiometriekoeffizienten:

$$\nu = \begin{pmatrix} -1 & -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & -1 & 1 & 0 & 0 \\ 1 & 0 & 1 & -1 & 0 & 0 \\ -3 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}.$$

Die Zeilen stehen für die Reaktionen, die Spalten für die Spezies einschließlich Lösungsmittel.

Das Differentialgleichungssystem für die Molzahländerungen lautet bei einem Semi-Batch-Reaktor mit Einträgen allgemein

$$\dot{n} = V \cdot \nu^T r + \dot{n}_e.$$

Weil die Reaktionsprodukte weder vorgelegt noch zugegeben werden, gilt für die Stoffe C, D und E

$$\dot{n}_{.,e} = 0, \quad n_{.,e} = 0, \quad n_{.,0} = 0.$$

So ergibt sich folgendes Differentialgleichungssystem:

$$\begin{aligned}
\dot{n}_A &= V \cdot (-r_1 - r_2 + r_3 - 3 \cdot r_4) + \dot{n}_{A,e} \\
\dot{n}_B &= V \cdot (-r_1) + \dot{n}_{B,e} \\
\dot{n}_C &= V \cdot (r_1 - r_2 + r_3) \\
\dot{n}_D &= V \cdot (r_2 - r_3) \\
\dot{n}_E &= V \cdot r_4 \\
\dot{n}_L &= \dot{n}_{L,e}.
\end{aligned}$$

Da Rang $\nu = 3$, genügen zur Beschreibung der chemischen Reaktion die Gleichungen für drei Spezies, deren Spalten in der ν-Matrix linear unabhängig sind, zum Beispiel für C, D und E.

Dann muß aber noch die Teilchenerhaltung zur vollständigen Beschreibung des Prozesses verwendet werden. Sei $r := m - \text{Rang } \nu = 3$. Die Matrix

$$\epsilon := \begin{pmatrix} 1 & 0 & 1 & 2 & 3 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

ist vollrangig und erfüllt

$$\epsilon \cdot \nu^T = 0.$$

Daher ergibt die Teilchenerhaltung

$$\epsilon \cdot \Delta n_R = 0,$$

das heißt

$$\begin{aligned}
\Delta n_{A,R} + \Delta n_{C,R} + 2 \cdot \Delta n_{D,R} + 3 \cdot \Delta n_{E,R} &= 0 \\
\Delta n_{B,R} + \Delta n_{C,R} + \Delta n_{D,R} &= 0 \\
\Delta n_{L,R} &= 0.
\end{aligned}$$

Mit den Stoffmengenänderungen

$$\begin{aligned}
\Delta n_{A,R} &= n_A - n_{A,0} - n_{A,e} \\
\Delta n_{B,R} &= n_B - n_{B,0} - n_{B,e} \\
\Delta n_{C,R} &= n_C \\
\Delta n_{D,R} &= n_D \\
\Delta n_{E,R} &= n_E \\
\Delta n_{L,R} &= n_L - n_{L,0} - n_{L,e}
\end{aligned}$$

erhalten wir somit als DAE-System für die Urethanreaktion

$$\begin{aligned}
\dot{n}_C &= V \cdot (r_1 - r_2 + r_3) \\
\dot{n}_D &= V \cdot (r_2 - r_3) \\
\dot{n}_E &= V \cdot r_4 \\
n_A &= n_{A,0} + n_{A,e} - n_C - 2 \cdot n_D - 3 \cdot n_E \\
n_B &= n_{B,0} + n_{B,e} - n_C - n_D \\
n_L &= n_{L,0} + n_{L,e}.
\end{aligned}$$

Zustandsvariablen des Modells sind also n_A, n_B, n_C, n_D, n_E und n_L.

Da wir als differentielle Variablen die Molzahlen der Reaktionsprodukte gewählt haben, haben die Anfangsbedingungen für ein Anfangswertproblem die folgende einfache Form:

$$n_C(t_0) = n_D(t_0) = n_E(t_0) = 0.$$

Für die Zeit t gilt

$$t \in [t_0; t_{end}] = [0\,h, 80\,h].$$

Das Reaktionsvolumen berechnet sich aus

$$V = \frac{n_A \cdot M_A}{\rho_A} + \frac{n_B \cdot M_B}{\rho_B} + \frac{n_C \cdot M_C}{\rho_C} + \frac{n_D \cdot M_D}{\rho_D} + \frac{n_E \cdot M_E}{\rho_E} + \frac{n_L \cdot M_L}{\rho_L}.$$

Für die Reaktionsgeschwindigkeiten setzen wir die Geschwindigkeitsgesetze

$$\begin{aligned} r_1 &= k_1 \cdot \frac{n_1}{V} \cdot \frac{n_2}{V} \\ r_2 &= k_2 \cdot \frac{n_1}{V} \cdot \frac{n_3}{V} \\ r_3 &= k_3 \cdot \frac{n_4}{V} \\ r_4 &= k_4 \cdot \left(\frac{n_1}{V}\right)^2 \end{aligned}$$

an, für die reversible Reaktion ist

$$k_3 = \frac{k_2}{K_C}.$$

Die Temperaturabhängigkeit der Reaktionsgeschwindigkeiten modellieren wir mit dem Arrhenius-Ansatz

$$\begin{aligned} k_1 &= k_{ref1} \cdot \exp\left(-\frac{E_{a,1}}{R} \cdot \left(\frac{1}{T} - \frac{1}{T_{ref1}}\right)\right) \\ k_2 &= k_{ref2} \cdot \exp\left(-\frac{E_{a,2}}{R} \cdot \left(\frac{1}{T} - \frac{1}{T_{ref2}}\right)\right) \\ k_4 &= k_{ref4} \cdot \exp\left(-\frac{E_{a,4}}{R} \cdot \left(\frac{1}{T} - \frac{1}{T_{ref4}}\right)\right) \\ K_C &= K_{C2} \cdot \exp\left(-\frac{\Delta H_2}{R} \cdot \left(\frac{1}{T} - \frac{1}{T_{C2}}\right)\right). \end{aligned}$$

Unbekannte Modellparameter sind hierbei die

- Aktivierungsenergien $E_{a,1}$, $E_{a,2}$, $E_{a,4}$,
- Frequenzfaktoren k_{ref1}, k_{ref2}, k_{ref4} und
- Gleichgewichtskonstante K_{C2} und Reaktionsenthalpie ΔH_2.

Die Einträge modellieren wir mittels Zulaufprofilen: Das Zulaufprofil $feed_1(t) \in [0,1]$ für Eintrag 1 bzw. das Zulaufprofil $feed_2(t) \in [0,1]$ für Eintrag 2 beschreibt, welcher Anteil der Gesamtzulaufmenge des jeweiligen Zulaufs schon in den Reaktor zugeführt wurde. Dann ist

$$\begin{aligned} n_{A,e} &= n_{A,e1,0} \cdot feed_1 \\ n_{B,e} &= n_{B,e2,0} \cdot feed_2 \\ n_{L,e} &= n_{L,e1,0} \cdot feed_1 + n_{L,e2,0} \cdot feed_2. \end{aligned}$$

Als Steuergrößen treten im Modell somit die Anfangsmolzahlen in

- Vorlage: $n_{A,0}$, $n_{B,0}$, $n_{L,0}$,
- Zulauf 1: $n_{A,e1,0}$, $n_{L,e1,0}$ und
- Zulauf 2: $n_{B,e2,0}$, $n_{L,e2,0}$

auf, Steuerfunktionen sind die

- Zulaufprofile: $feed_1(t)$, $feed_2(t)$ und die
- Innentemperatur $T(t)$.

Alle im Modell auftretenden Stoff- und Naturkonstanten sind in Tabelle 10.6 zusammengestellt.

Molmassen	Dichten	Referenztemperaturen
$M_A = 0.11911\,kg/mol$	$\rho_A = 1095.0\,kg/m^3$	$T_{ref1} = 363.16\,K$
$M_B = 0.07412\,kg/mol$	$\rho_B = 809.0\,kg/m^3$	$T_{ref2} = 363.16\,K$
$M_C = 0.19323\,kg/mol$	$\rho_C = 1415.0\,kg/m^3$	$T_{ref4} = 363.16\,K$
$M_D = 0.31234\,kg/mol$	$\rho_D = 1528.0\,kg/m^3$	$T_{C2} = 363.16\,K$
$M_E = 0.35733\,kg/mol$	$\rho_E = 1451.0\,kg/m^3$	Gaskonstante
$M_L = 0.07806\,kg/mol$	$\rho_L = 1101.0\,kg/m^3$	$R = 8.314\,J/(K \cdot mol)$

Tabelle 10.6: Konstanten im Modell der Urethanreaktion

10.2.4 Messungen

Zur Ermittlung von experimentellen Daten stehen die folgenden Meßverfahren zur Verfügung:

- Titration: Messung der Massenprozent von Phenylisocyanat mit einem absoluten Meßfehler von 0.5,

- HPLC-1: Messung der Massenprozent von Urethan mit absolutem Meßfehler 0.5 und Allophanat mit absolutem Meßfehler 0.005,
- HPLC-2: Messung der Massenprozent von Isocyanurat mit absolutem Meßfehler 0.0005.

Zur Beschreibung der HPLC-Meßmethode siehe [66].

Pro Experiment können maximal 16 Messungen durchgeführt werden.

10.2.5 Nebenbedingungen an die Steuerungen

An Molverhältnisse, Wirkstoffgehalte und das Anfangsvolumen werden die folgenden Nebenbedingungen gestellt:

$$\begin{aligned}
MV_1 &:= \frac{n_{B,0} + n_{B,e2,0}}{n_{A,0} + n_{A,e1,0}} && \in [0.1; 10] \\
MV_2 &:= \frac{n_{A,e1,0}}{n_{A,0}} && \in [0; 1000] \\
MV_3 &:= \frac{n_{B,e2,0}}{n_{A,0}} && \in [0; 10] \\
g_a &:= \frac{n_{A,0} \cdot M_A + n_{B,0} \cdot M_B}{n_{A,0} \cdot M_A + n_{B,0} \cdot M_B + n_{L,0} \cdot M_L} && \in [0; 0.8] \\
g_{a,e1} &:= \frac{n_{A,e1,0} \cdot M_A}{n_{A,e1,0} \cdot M_A + n_{L,e1,0} \cdot M_L} && \in [0; 0.9] \\
g_{a,e2} &:= \frac{n_{B,e2,0} \cdot M_B}{n_{B,e2,0} \cdot M_B + n_{L,e2,0} \cdot M_L} && \in [0; 1] \\
V_0 &:= \frac{n_{A,0} \cdot M_A}{\rho_A} + \frac{n_{B,0} \cdot M_B}{\rho_B} + \frac{n_{L,0} \cdot M_L}{\rho_L} && \in [0\,m^3; 0.00075\,m^3]
\end{aligned}$$

Leitsubstanz für die Molverhältnisse ist $n_{A,0}$. Daher haben wir die untere Grenze

$$n_{A,0} \geq 0.05\,mol.$$

Die Edukte können nicht direkt eingewogen werden. Sie liegen vielmehr als Reagenzien in gegebenen Konzentrationen vor. Drei verschiedene Reagenzien sind verfügbar:

- NCO besteht aus 90 (Massen-)% Phenylisocyanat A und 10% Dimethylsulfoxid L.
- BUOH besteht aus 100%igem Butanol B.
- DMSO besteht aus 100%igem Dimethylsulfoxid L.

Für die einzuwiegenden Massen der Reagenzien

- für die Vorlage: R_{NCO}, R_{BUOH}, R_{DMSO},

- für Zulauf 1: $R_{NCO,e1}$, $R_{DMSO,e1}$ und
- für Zulauf 2: $R_{BUOH,e2}$, $R_{DMSO,e2}$

gelten also die folgenden Gleichungen:

$$\begin{aligned} n_{A,0} \cdot M_A &= 0.9 \cdot R_{NCO} \\ n_{B,0} \cdot M_B &= 1.0 \cdot R_{BUOH} \\ n_{L,0} \cdot M_L &= 0.1 \cdot R_{NCO} + 1.0 \cdot R_{DMSO} \\ n_{A,e1,0} \cdot M_A &= 0.9 \cdot R_{NCO,e1} \\ n_{L,e1,0} \cdot M_L &= 0.1 \cdot R_{NCO,e1} + 1.0 \cdot R_{DMSO,e1} \\ n_{B,e2,0} \cdot M_B &= 1.0 \cdot R_{BUOH,e2} \\ n_{L,e2,0} \cdot M_L &= 1.0 \cdot R_{DMSO,e2}. \end{aligned}$$

Da die gesamte Zulaufmenge während der Reaktion zugeführt werden soll, werden an die Steuerfunktionen der Zulaufprofile die folgenden Nebenbedingungen gestellt:

$$feed_1(t_0) = 0, \quad feed_1(t_{end}) = 1, \quad \dot{feed}_1 >= 0$$

und

$$feed_2(t_0) = 0, \quad feed_2(t_{end}) = 1, \quad \dot{feed}_2 >= 0.$$

Die Werte und Änderungsraten der Temperatur sollen in den folgenden Grenzen bleiben:

$$T(t_0) = 293.16\,K, \quad T(t) \in [293.16\,K, 473.16\,K], \quad \dot{T} \in [-40\,K/h, 40\,K/h].$$

Stunden	Schicht
0 bis 8	Montag
8 bis 24	1. Nacht
24 bis 32	Dienstag
32 bis 48	2. Nacht
48 bis 56	Mittwoch
56 bis 72	3. Nacht
72 bis 80	Donnerstag

Tabelle 10.7: Zeitplan für ein Experiment der Urethanreaktion. Nur in den Tagschichten sollen Messungen und Steuerungsänderungen durchgeführt werden, nachts soll das System weitgehend ruhen.

Ein Experiment dauert von Montag Morgen bis Donnerstag Nachmittag, siehe Tabelle 10.7. Neben den vier Tagschichten gibt es auch Nachtruhe, was durch die Ruhebedingungen

$$\dot{feed}_1(t) = \dot{feed}_2(t) = \dot{T}(t) = 0, \quad t \in [8\,h, 24\,h] \cup [32\,h, 48\,h] \cup [56\,h, 72\,h]$$

erreicht werden soll.

10.2.6 Parametrisierung der Steuerfunktionen und Meßgitter

Für die Steuerfunktionen verwenden wir die folgenden Parametrisierungen:

- Die beiden Zulaufprofile stückweise linear und stetig mit den Schaltpunkten

$$t = 8, 24, 32, 48, 56, 72, 79\, h$$

- und die Temperatur stückweise linear und stetig mit den Schaltpunkten

$$t = 8, 24, 32, 48, 56, 72\, h.$$

Als Gitter der möglichen Meßzeitpunkte legen wir fest:

$$t = 0.5, 1, 4, 8, 24, 32, 48, 56, 72, 80\, h.$$

Daraus folgt, daß es 30 mögliche Messungen gibt, aus denen 16 ausgewählt werden sollen.

Es ergibt sich somit der Zeitplan für jedes einzelne Experiment, wie er in Abbildung 10.9 dargestellt ist.

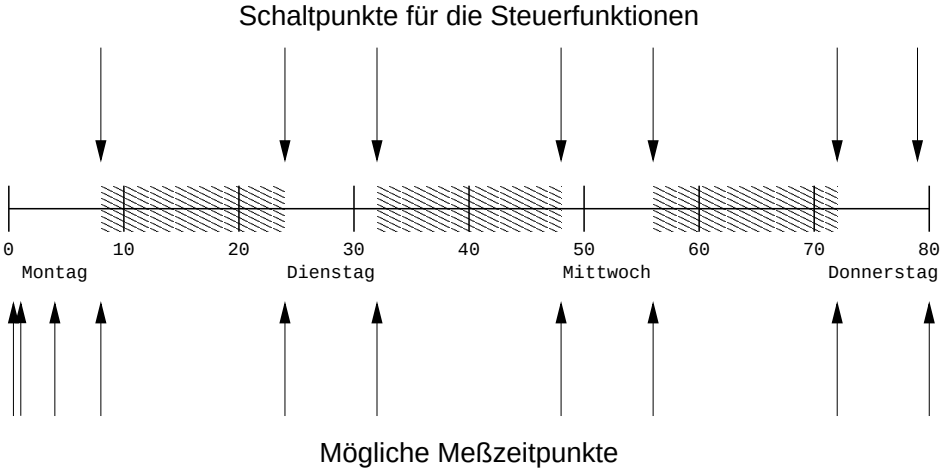


Abbildung 10.9: Schaltpunkte und mögliche Meßzeitpunkte bei der Urethanreaktion

10.2.7 Startschätzung für die Parameter

Für die Parameter wählen wir die Startwerte

$$\begin{aligned} k_{ref1} &= 5.0 \cdot 10^{-4}\, m^3/(h \cdot mol) \\ E_{a,1} &= 35240.0\, J/mol \\ k_{ref2} &= 8.0 \cdot 10^{-8}\, m^3/(h \cdot mol) \end{aligned}$$

$$\begin{aligned} E_{a,2} &= 85000.0\,J/mol \\ k_{ref4} &= 1.0 \cdot 10^{-8}\,m^3/(h \cdot mol) \\ E_{a,4} &= 35000.0\,J/mol \\ \Delta H_2 &= -17031.0\,J/mol \\ K_{C2} &= 0.17\,m^3/mol. \end{aligned}$$

Wir numerieren die Parameter in dieser Reihenfolge und skalieren sie so, daß sie zunächst alle den Wert 1.0 haben:

$$p_i = 1.0, \quad i = 1, \ldots 8.$$

Also ist

$$\begin{aligned} k_{ref1} &= p_1 \cdot 5.0 \cdot 10^{-4}\,m^3/(h \cdot mol) \\ E_{a,1} &= p_2 \cdot 35240.0\,J/mol \\ k_{ref2} &= p_3 \cdot 8.0 \cdot 10^{-8}\,m^3/(h \cdot mol) \\ E_{a,2} &= p_4 \cdot 85000.0\,J/mol \\ k_{ref4} &= p_5 \cdot 1.0 \cdot 10^{-8}\,m^3/(h \cdot mol) \\ E_{a,4} &= p_6 \cdot 35000.0\,J/mol \\ \Delta H_2 &= p_7 \cdot (-17031.0\,J/mol) \\ K_{C2} &= p_8 \cdot 0.17\,m^3/mol. \end{aligned}$$

10.2.8 Expertendesign

Um die Leistungsfähigkeit der optimalen Versuchsplanung zu erproben, wurde eine Vergleichsstudie mit einem intuitivem Experimententwurf durchgeführt. Ein erfahrener Experimentator der BASF wurde gebeten, Experimente für die Urethanreaktion vorzuschlagen.

Seine Planung basierte auf den folgenden Prinzipien:

- Separation der Reaktionen,
- Batch-Fahrweise,
- isotherme Experimente.

Insgesamt plante er 15 Experimente mit zusammen 90 Probennahmen. Eine Übersicht über die experimentellen Einstellungen gibt Tabelle 10.8.

Die Anfangstemperatur des Reaktionsgemischs ist dabei immer $T = 293.16\,K$ bei $t_0 = 0$. Dann wird die Temperatur innerhalb einer halben Stunde gleichmäßig auf Reaktionstemperatur erhöht und diese bis Reaktionsende konstant gehalten.

Wegen der Batch-Fahrweise sind die folgenden Steuerungen immer gleich null:

$$MV_2 = MV_3 = g_{a,e1} = g_{a,e2} = 0.$$

Exp.	Temp. [K]	MV_1	g_a	$V_0[cm^3]$	Meßverf.	Meßzeitpunkte
1	400	1.0	0.3	750	Titration	$5, 30, 50\,min$
2	400	1.0	0.3	750	Titration	$5, 10, 15, 20, 25, 30, 35,$ $40, 50, 60, 70, 90, 120\,min$
3	390	1.0	0.3	750	Titration	$30, 40, 50, 60, 80, 100, 120\,min$
4	370	1.0	0.3	750	Titration	$30, 40, 50, 60, 70, 90, 120\,min$
5	400	1.0	0.3	750	HPLC-1	$3, 6, 9, 12, 15, 30, 60\,h$
6	370	1.0	0.3	750	HPLC-1	$12, 24\,h$
7	400	1.0	0.3	750	HPLC-1	$1, 1.5, 2, 2.5, 3\,h$
8	340	1.0	0.3	750	HPLC-1	$1, 2, 4, 6, 8, 12\,h$
9	400	1.0	0.3	750	HPLC-1	$20, 40, 60, 80\,h$
10	430	1.0	0.3	750	HPLC-1	$20, 40, 60, 80\,h$
11	400	1.0	0.177	750	HPLC-2	$20, 40, 60, 80\,h$
12	430	1.0	0.177	750	HPLC-2	$20, 40, 60, 80\,h$
13	460	1.0	0.177	750	HPLC-2	$10, 20, 40, 60, 80\,h$
14	400	1.0	0.3	750	HPLC-1	$4, 6, 10, 20, 40, 60, 80\,h$
15	420	1.0	0.3	750	HPLC-1	$1, 1.5, 2, 2.5, 3, 4, 6,$ $10, 20, 40, 60, 80\,h$

Tabelle 10.8: Übersicht über die intuitiv geplanten Experimente für die Urethanreaktion

Par.	Intuitive Schätzung		
p_1	2.50	$\pm$	0.02
p_2	0.835	$\pm$	0.007
p_3	91.3	$\pm$	0.7
p_4	0.834	$\pm$	0.002
p_5	57.991	$\pm$	0.009
p_6	0.657225	$\pm$	0.00007
p_7	0.9	$\pm$	0.3
p_8	1.1	$\pm$	0.3

Tabelle 10.9: Parameterschätzung für die Meßdaten aus den 15 intuitiv geplanten Experimenten für die Urethanreaktion

Eine Parameterschätzung für die experimentellen Daten aus den 15 Experimenten ergibt die in Tabelle 10.9 dargestellten Ergebnisse.

Während also die Parameter für die Hinreaktionen gut geschätzt werden können, ist die Rückreaktion des Gleichgewichts, deren Edukt ja nicht eingesetzt werden konnte, nur sehr schlecht bestimmt. Das Prinzip der Separation der Reaktionen führt hier trotz der hohen Anzahl der Experimente also nicht zum Erfolg.

10.2.9 Wahl des initialen Experimentvorschlags

Wir wählen als Anfangsvorschlag für die Versuchsplanungsoptimierung ein Experiment mit einer Semi-Batch-Fahrweise. Der Inhalt der Eintragsgefäße wird während der Tagschichten gleichmäßig zugeführt, für Eintrag 1 während der ersten zwei, für Eintrag 2 während der ersten drei Tage. Die Temperatur wird am ersten Tag auf $473\,K$ erhitzt und an den folgenden Tagen wieder bis auf $338\,K$ abgekühlt. Die Steuergrößen wurden folgendermaßen gewählt:

MV_1	MV_2	MV_3	g_a	$g_{a,e1}$	$g_{a,e2}$	V_0
1.0	0.3	0.3	0.75	0.5	0.4	$2.75 \cdot 10^{-5}\,m^3$

Abbildung 10.10 zeigt den Verlauf der Steuerfunktionen und die daraus resultierenden Trajektorien.

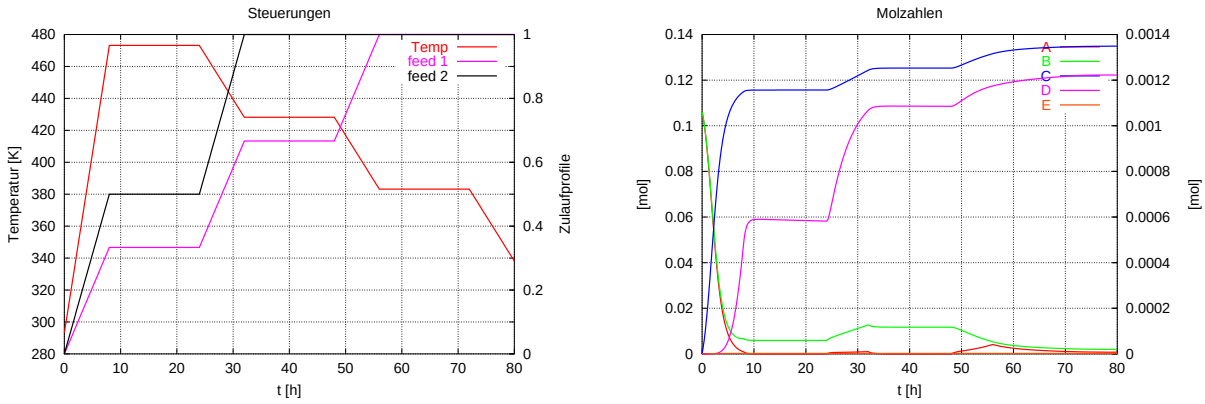


Abbildung 10.10: Steuerfunktionen und Trajektorien der Molzahlen für den initialen Experimentvorschlag für die Urethanreaktion

10.2.10 Parameterschätzung und Optimale Versuchsplanung in sequentieller Vorgehensweise

Zunächst wurde, ausgehend von dem initialen Experimentvorschlag und für die Startschätzung der Parameter, ein erstes Experiment optimiert. Als Zielfunktion wurde das A-Kriterium

$$\phi_A(C) = \frac{1}{n} \cdot \mathrm{Spur}C$$

gewählt.

Durch die Optimierung verbesserte sich das Gütekriterium von 35.5481 auf 0.00113361.

Im optimierten Experiment haben die Steuergrößen die folgenden Werte:

MV_1	MV_2	MV_3	g_a	$g_{a,e1}$	$g_{a,e2}$	V_0
0.637635	15.9264	10.0	0.8	0.9	1.0	$2.09045 \cdot 10^{-5}\, m^3$

Die Gewichte der Messungen waren ganzzahlig bis auf drei, von denen zwei auf- und eins abgerundet wurde. Dadurch verschlechterte sich der Zielfunktionswert aber nur unwesentlich von 0.00113361 auf 0.00113406.

Für die die Probenahmen folgen daraus die Zeitpunkte:

- Titration bei $t = 0.5, 1\, h$,
- HPLC-1 bei $t = 0.5, 1, 4, 8, 24, 32, 48, 56, 72, 80\, h$,
- HPLC-2 bei $t = 4, 8, 32, 48\, h$.

Eine Parameterschätzung mit experimentellen Daten aus dem optimierten Experiment ergibt die Werte und Standardabweichungen in Tabelle 10.10.

Par.	Erste Schätzung		
p_1	2.51	$\pm$	0.03
p_2	0.835	$\pm$	0.004
p_3	91.2	$\pm$	0.1
p_4	0.8355	$\pm$	0.0002
p_5	58.0	$\pm$	0.1
p_6	0.658	$\pm$	0.001
p_7	1.09	$\pm$	0.02
p_8	1.30	$\pm$	0.03

Tabelle 10.10: Parameterschätzung für die Meßdaten aus dem ersten optimierten Experiment für die Urethanreaktion

Abbildung 10.11 zeigt die Steuerfunktionen und die Trajektorien für das erste Experiment.

Um die Güte der Parameterschätzung weiter zu verbessern, wurde eine weitere Versuchsplanung durchgeführt. Unter Berücksichtigung der Vorinformationen aus dem ersten Experiment wurde nun ein Zweifachexperiment mit festgehaltenem ersten Experiment optimiert, wieder mit dem A-Kriterium und dem initialen Experimentvorschlag für das zusätzliche zweite Experiment, aber nun für den aus der ersten Schätzung ermittelten Parametersatz.

Bei dieser Optimierung verbesserte sich der Wert des Gütekriteriums von 0.00213083 auf 0.0000557949.

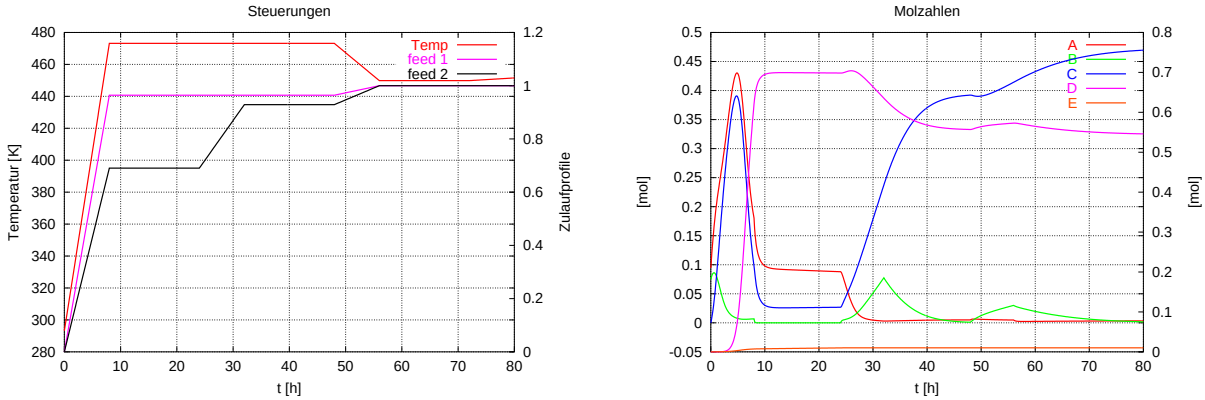


Abbildung 10.11: Steuerfunktionen und Trajektorien der Molzahlen für das erste optimierte Experiment für die Urethanreaktion

Für die Steuergrößen des zweiten Experiments ergab die Optimierung:

MV_1	MV_2	MV_3	g_a	$g_{a,e1}$	$g_{a,e2}$	V_0
0.370994	25.9546	10.0	0.8	0.9	1.0	$9.00762 \cdot 10^{-6}\, m^3$

Hier war keine Rundung der Meßgewichte nötig, da die Lösung schon ganzzahlig war. Probenahmen für die verschiedenen Meßverfahren wurden wie folgt vorgeschlagen:

- Titration: keine,
- HPLC-1 bei $t = 4, 8, 24, 32, 48, 56, 72, 80\, h$,
- HPLC-2 bei $t = 4, 8, 24, 32, 48, 56, 72, 80\, h$.

Eine Parameterschätzung aus den Meßdaten beider Experimente liefert die Parameterwerte in Tabelle 10.11.

Par.	Zweite Schätzung		
p_1	2.504	$\pm$	0.006
p_2	0.836	$\pm$	0.001
p_3	91.25	$\pm$	0.02
p_4	0.83537	$\pm$	0.00008
p_5	58.004	$\pm$	0.006
p_6	0.6574	$\pm$	0.0002
p_7	1.082	$\pm$	0.006
p_8	1.29	$\pm$	0.01

Tabelle 10.11: Parameterschätzung für die Meßdaten aus dem ersten und zweiten optimierten Experiment für die Urethanreaktion

Die Steuerfunktionen und Trajektorien für das zweite Experiment sind in Abbildung 10.12 dargestellt.

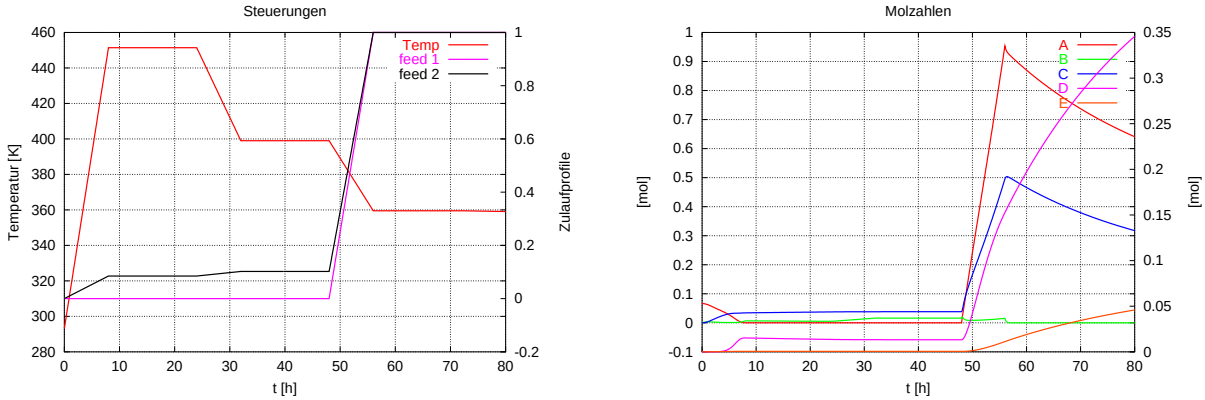


Abbildung 10.12: Steuerfunktionen und Trajektorien der Molzahlen für das zweite optimierte Experiment für die Urethanreaktion

Die optimierten Experimente zeigen, daß mit den Möglichkeiten, die sich aus der Semi-Batch-Fahrweise und der Temperatursteuerung ergeben, sehr virtuos gespielt werden kann, natürlich im Rahmen der vorgegebenen Beschränkungen. Durch Variation der Zulaufraten kann zum Beispiel der Verlauf der Reaktionen und insbesondere die Lage des Gleichgewichts beeinflußt werden. So kann die Rückreaktion oder die Trimerisierungsnebenreaktion gezielt erzwungen und untersucht werden. Durch nur zwei Experimente erhält man dadurch genügend Informationen, um alle Parameter mit zufriedenstellender Güte schätzen zu können.

10.2.11 Vergleich zwischen intuitiver und optimaler Versuchsplanung

In Tabelle 10.12 sind die Ergebnisse der Parameterschätzungen für die intuitive Vorgehensweise und für die optimierten Experimente nochmals zusammengestellt.

Par.	Intuitive Schätzung			Erste Schätzung			Zweite Schätzung		
p_1	2.50	$\pm$	0.02	2.51	$\pm$	0.03	2.504	$\pm$	0.006
p_2	0.835	$\pm$	0.007	0.835	$\pm$	0.004	0.836	$\pm$	0.001
p_3	91.3	$\pm$	0.7	91.2	$\pm$	0.1	91.25	$\pm$	0.02
p_4	0.834	$\pm$	0.00	0.8355	$\pm$	0.0002	0.83537	$\pm$	0.00008
p_5	57.991	$\pm$	0.009	58.0	$\pm$	0.1	58.004	$\pm$	0.006
p_6	0.65725	$\pm$	0.00007	0.658	$\pm$	0.001	0.6574	$\pm$	0.0002
p_7	0.9	$\pm$	0.3	1.09	$\pm$	0.02	1.082	$\pm$	0.006
p_8	1.1	$\pm$	0.3	1.30	$\pm$	0.03	1.29	$\pm$	0.01

Tabelle 10.12: Vergleich der Parameterschätzungen bei intuitiver Vorgehensweise mit 15 Experimenten und optimaler Versuchsplanung nach einem und nach zwei Experimenten für die Urethanreaktion.

Vergleicht man Aufwand und Ergebnis für den intuitiven Versuchsplan und die optimierten Experimente, siehe Tabelle 10.13, erkennt man, welch großes Einsparpotential sich

durch die Anwendung der Methoden der optimalen Versuchsplanung ergibt.

Vorgehens-weise	Anzahl Experimente	Anzahl Messungen	maximaler Fehler
intuitiv	15	90	$> 25\%$
methodengestützt	2	32	$< 1\%$

Tabelle 10.13: Vergleich zwischen intuitiver Vorgehensweise und optimaler Versuchsplanung für die Urethanreaktion hinsichtlich Aufwand der Experimente und Güte der Schätzungen

10.3 Eine mehrphasige Veresterungsreaktion

Bei diesem Beispiel handelt es sich um einen Prozeß aus der industriellen Praxis der chemischen Produktentwicklung. Er besteht aus einem komplexen, mehrphasigen Reaktionssystem und wird in einem Reaktor mit Temperatur- und Drucksteuerung sowie Destillation der aus der Gasphase durchgeführt. Ziel der Untersuchungen ist die Modellvalidierung durch möglichst genaue Schätzung der Modellparameter.

10.3.1 Reaktion und Prozeß

Dieses Reaktionssystem wurde in einem gemeinsamen Projekt mit der BASF untersucht. Da der Prozeß Geheimhaltungsbestimmungen unterliegt, können hier keine Beschreibung des chemischen Hintergrunds, keine expliziten Bezeichnungen der beteiligten Spezies und keine Zahlenwerte von Stoffeigenschaften und Parametern angegeben werden.

Das Reaktionssystem hat vier Phasen:

1. In der Flüssigphase laufen die chemischen Reaktionen ab.
2. Da die Reaktion in einem geschlossenen Reaktor durchgeführt wird, bildet sich über der Flüssigkeit durch Verdampfung eine Gasphase.
3. Ein Edukt wird als Feststoff zugegeben, der sich nur teilweise in der Flüssigkeit löst.
4. Ein Reaktionsprodukt fällt als Feststoff aus.

Die Reaktion wird in einem Reaktor mit einem Zulauf, einem Gasaustrag durch Dampfdestillation und Temperatur- und Drucksteuerung durchgeführt. Das Schema des Reaktors, die Zugehörigkeit der Spezies zu den Phasen sowie die Reaktionen und Phasenübergänge sind in Abbildung 10.13 dargestellt.

Der Prozeß kann durch die Einstellung folgender Größen gesteuert werden:

- Anfangsmolzahlen der Stoffe A, C, D, E, F und J in der Vorlage und der Stoffe D und F im Zulauf,
- Dosierungsrate $feed(t)$ des Zulaufs,
- Profil der Temperatur $T(t)$ des Reaktionsgemischs,
- Druckprofil $p_{st}(t)$ oder Destillationsrate $m_a(t)$.

Zur Erhebung von experimentellen Daten stehen zwei Meßverfahren zur Verfügung:

1. High Performance Liquid Chromatography (HPLC) [66]. Gemessen werden die Massenanteile der Spezies, und zwar für feste und flüssige Phasen von $A + B$ und $G + J$, für die flüssige Phase von C, E und F und für die Gasphase von I. Die absoluten Meßfehler liegen zwischen 0.2 und 1.5 Prozentpunkten.
2. Druckmessung mit einem Meßfehler von $1000\,Pa$.

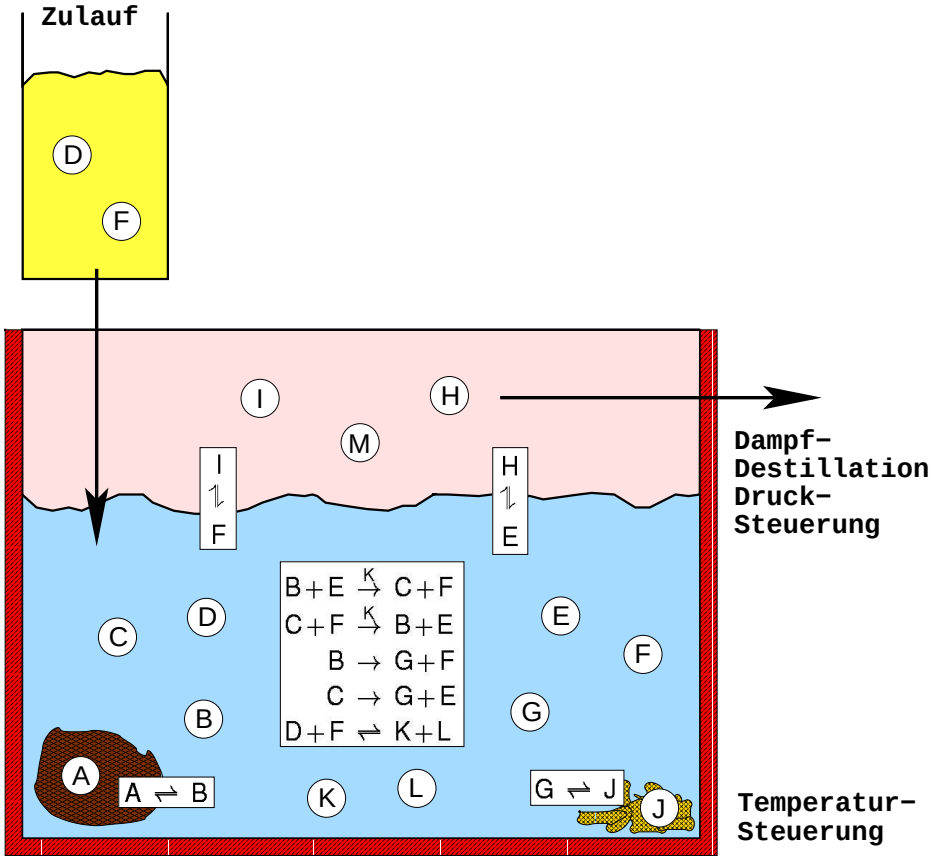


Abbildung 10.13: Reaktor und Reaktionsschema des Veresterungsbeispiels

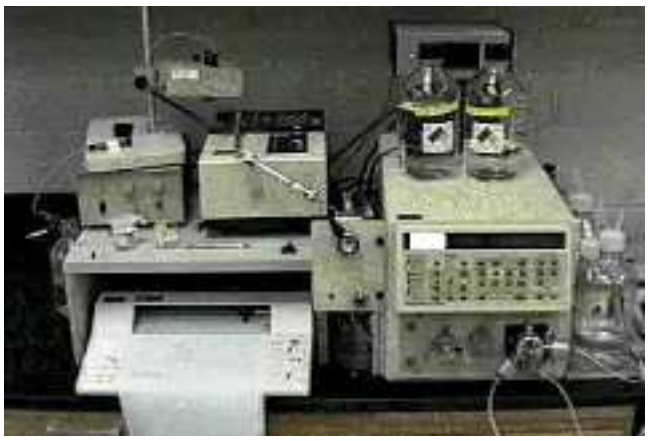


Abbildung 10.14: Apparatur der High Performance Liquid Chromatography (HPLC) (Photo aus [66])

10.3.2 Mathematisches Modell

In diesem Abschnitt werden die Grundprinzipien der mathematischen Modellformulierung vorgestellt. Auf die komplette Darstellung des Modells wird aus Platz- und Geheimhaltungsgründen verzichtet.

Die **Stoffmengenbilanz** für die im Reaktor vorhandenen Stoffe lautet unter Berücksichtigung von Termen für Reaktionen und Phasenübergänge n_R, Eintrag n_e, Austrag n_a und Einwaagen n_0 als Differentialgleichungssystem

$$\dot{n} = \dot{n}_R + \dot{n}_e - \dot{n}_a$$

bzw. in integrierter Form

$$\Delta n_R = n - n_0 - n_e + n_a.$$

Mit einer geeigneten Bilanzmatrix ϵ ergeben sich die algebraischen Gleichungen

$$\epsilon \cdot \Delta n_R = 0.$$

Alle Reaktionen finden in der Flüssigphase statt. Wir modellieren die **Geschwindigkeitsgesetze** mit dem Arrhenius-Ansatz. Nicht katalysierte Reaktionen haben die Reaktionsgeschwindigkeit

$$r_i = k_{refi} \cdot \exp\left(-\frac{E_{ai}}{R_{gas}} \cdot \left(\frac{1}{T} - \frac{1}{T_{refi}}\right)\right) \cdot \prod_j \left(\frac{n_j}{V_{fl}}\right).$$

V_{fl} ist das Volumen der Flüssigphase und T_{refi} die jeweilige Referenztemperatur.

Für katalysierte Reaktionen hängt die Reaktionsgeschwindigkeit außerdem von der Konzentration $c_{kat} = \frac{n_K}{V_{fl}}$ des Katalysators K ab:

$$r_i = k_{kati} \cdot c_{kat} \cdot \exp\left(-\frac{E_{ai}}{R_{gas}} \cdot \left(\frac{1}{T} - \frac{1}{T_{refi}}\right)\right) \cdot \prod_j \left(\frac{n_j}{V_{fl}}\right).$$

Der **Stoffaustausch** zwischen den Phasen ergibt ebenfalls Änderungsraten für die Molzahlen der einzelnen Spezies. Für den Phasenübergang flüssig – gasförmig, zum Beispiel zwischen den Spezies E (flüssig) und H (gasförmig), gilt:

$$\begin{aligned} \dot{n}_{E,SA} &= -\frac{k_{SA,E,H}}{R_{gas} \cdot T} \cdot (p_{H,NRTL} - p_H) \\ \dot{n}_{H,SA} &= -\dot{n}_{E,SA}. \end{aligned}$$

Dabei ist $p_{H,NRTL}$ der Gleichgewichtspartialdruck von H unter Berücksichtigung der Wechselwirkungen realer Gase nach dem NRTL-Modell [99, 111] und $p_H = \frac{n_H \cdot R_{gas} \cdot T}{V_{gas}}$ der aktuelle Partialdruck von H. V_{gas} bezeichnet das Volumen der Gasphase. Für den Stoffübergang F – I gelten analoge Gleichungen.

Für den Stoffaustausch fest – flüssig, zum Beispiel zwischen A (fest) und B (flüssig), ist die Differenz aus aktueller Konzentration und Grenzlöslichkeit c_l ausschlaggebend:

$$\begin{aligned} \dot{n}_{A,SA} &= -k_{SA,A,B} \cdot (c_l - \frac{n_B}{V_{fl}}) \\ \dot{n}_{B,SA} &= -\dot{n}_{A,SA}. \end{aligned}$$

Der Phasenübergang G – J wird analog modelliert.

Die **Zulaufraten** der Spezies im Eintrag berechnen sich aus dem Zulaufprofil und den Anfangsmolzahlen im Zulauf:

$$n_e = n_{0,e} \cdot feed.$$

Der **Austrag** erfolgt durch Destillation aus der Gasphase. Hier sind zwei Abschnitte der Prozeßführung zu unterscheiden, zwischen denen ein Modellwechsel durchgeführt wird:

1. Zunächst wird der Gasdruck gesteuert:

   $$p = p_{st}.$$

   Dann gilt für die Austragsrate

   $$\dot{out} = \frac{\dot{T}}{T} - \frac{\dot{p}_{st}}{p_{st}} - \frac{\dot{V}_{gas}}{V_{gas}} + \frac{\dot{n}_{H,SA} + \dot{n}_{I,SA}}{n_{gas}}.$$

2. Anschließend ist die Massenaustragsrate $\dot{m}_a$ die Steuerfunktion. Dann ist

   $$\dot{out} = \frac{\dot{m}_a}{m_{gas}} - m_a \cdot \frac{\dot{m}_{gas} \cdot V_{gas} - m_{gas} \cdot \dot{V}_{gas}}{m_{gas}^2 \cdot V_{gas}},$$

   und der Gasdruck läßt sich gemäß

   $$p = n_{gas} \cdot \frac{R_{gas} \cdot T}{V_{gas}}$$

   berechnen.

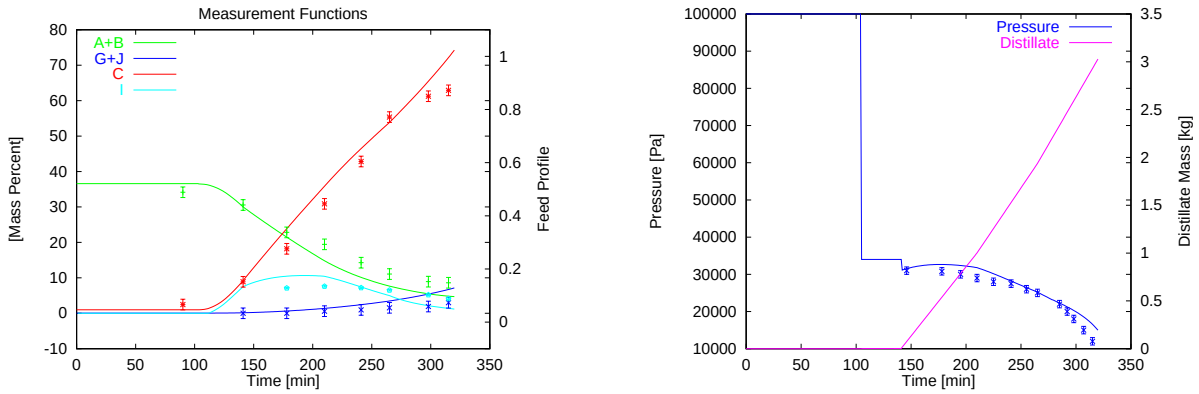
Für den Austrag der einzelnen Spezies aus der Gasphase gilt

$$\dot{n}_a = n \cdot \dot{out}.$$

Wir erhalten insgesamt ein nichtlineares DAE-Modell mit 13 Zustandsvariablen und sechs unbekannten Parametern:

- 3 Frequenzfaktoren k_{kat3}, k_{kat4}, k_{ref6} und
- 3 Aktivierungsenergien E_{a3}, E_{a4}, E_{a6}.

Experiment 1:



Experiment 2:

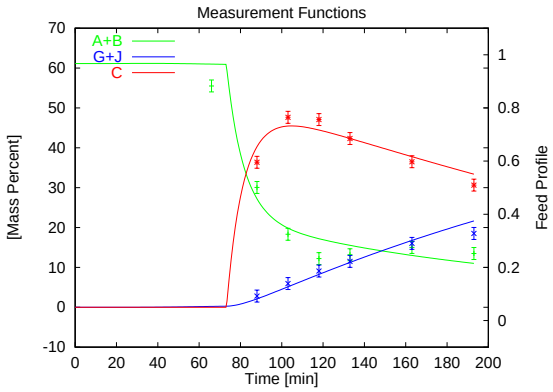


Abbildung 10.15: Parameterschätzung beim Veresterungsbeispiel: Anpassung der zwei Vorexperimente

10.3.3 Parameterschätzung

Im November 1998 wurden bei der BASF bereits zwei Experimente für dieses Reaktionssystem durchgeführt. Anhand der daraus gewonnenen Meßdaten berechneten wir eine Parameterschätzung. Abbildung 10.15 zeigt die Anpassung der Modellantwort an die Meßwerte.

Die Standardabweichungen der Parameter betrugen bei dieser Schätzung:

$$\begin{aligned} k_{kat3}: &\ \pm\ 25\% \\ k_{kat4}: &\ \pm\ 28\% \\ k_{ref6}: &\ \pm\ 4\% \\ E_{a3}: &\ \pm\ 14\% \\ E_{a4}: &\ \pm\ 35\% \\ E_{a6}: &\ \pm\ 5\% \end{aligned}$$

Um die Parameter zuverlässiger bestimmen zu können, ergab sich daraus der Bedarf, zusätzliche Experimente mittels optimaler Versuchsplanung zu entwerfen.

10.3.4 Versuchsraum

Jedes Experiment dauert $300\,min$.

Zu Beginn werden die Edukte im Reaktor vorgelegt.

In einer Aufheizphase wird das Reaktionsgemisch ausgehend von Raumtemperatur linear auf Reaktionstemperatur erhitzt. Diese soll nach $90\,min$ erreicht sein und zwischen $313\,K$ und $383\,K$ liegen. In der Aufheizphase wird der Druck im Reaktor auf Außendruck gehalten.

Nach $90\,min$ wird mit der Destillation begonnen, es wird von Drucksteuerung auf Austragssteuerung umgeschaltet. Die Temperatur wird von $90\,min$ bis $150\,min$ nach Reaktionsbeginn konstant gehalten, dann wird eine lineare Änderung mit Steigung zwischen $-0.2\,K/min$ und $0.2\,K/min$ zugelassen.

Die Stoffe im Zulauf können ab $89\,min$ nach Reaktionsbeginn zugegeben werden.

An die Steuergrößen sind Nebenbedingungen bezüglich der Molverhältnisse und des Anfangsvolumens sowie des Massenverhältnisses im Zulauf gestellt.

Durch eine Volumennebenbedingung wird gewährleistet, daß der Reaktor nicht überläuft:

$$V_{fest} + V_{fl} \leq V_{fuell}.$$

Pro Experiment sollen 10 HPLC-Messungen durchgeführt werden. Der Druck kann beliebig oft gemessen werden.

10.3.5 Optimale Versuchsplanung

Mit unserer Software VPLAN haben wir durch optimale Versuchsplanung drei zusätzliche Experimente entworfen unter Berücksichtigung der zwei Vorexperimente. Das Optimierungsproblem hatte dabei 189 Variablen. Bei der Optimierung hat sich die Zielfunktion, das A-Gütekriterium, von 0.02 auf 0.0006 verringert.

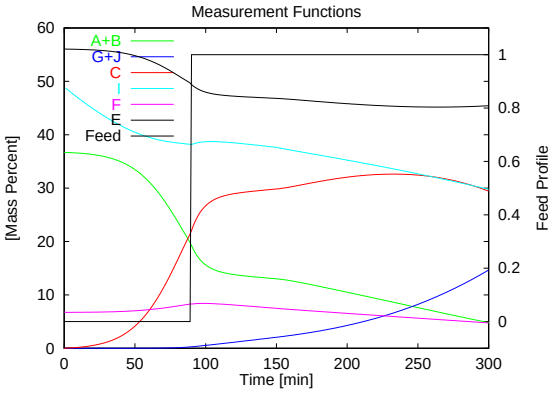
In Abbildung 10.16 sind die drei optimierten Experimente beschrieben.

Für alle fünf Experimente ergeben sich die folgenden Standardabweichungen der Parameter:

$$\begin{aligned} k_{kat3}: &\ \pm\ 1.1\% \\ k_{kat4}: &\ \pm\ 3.8\% \\ k_{ref6}: &\ \pm\ 2.1\% \\ E_{a3}: &\ \pm\ 0.8\% \\ E_{a4}: &\ \pm\ 2.8\% \\ E_{a6}: &\ \pm\ 3.0\% \end{aligned}$$

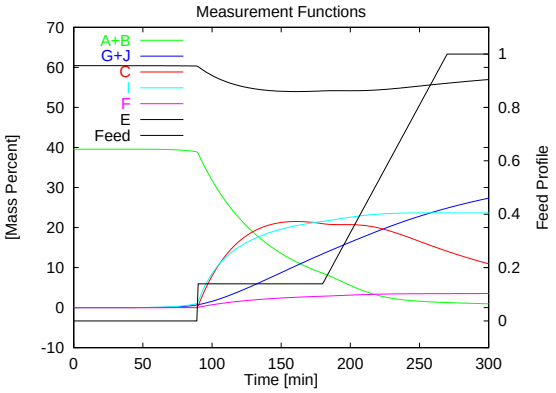
Dies bedeutet eine Verringerung des Schätzfehlers durch optimale Versuchsplanung um einen Faktor von ungefähr 10, alle Parameter können nun mit genügend hoher Genauigkeit bestimmt werden.

Experiment 3:



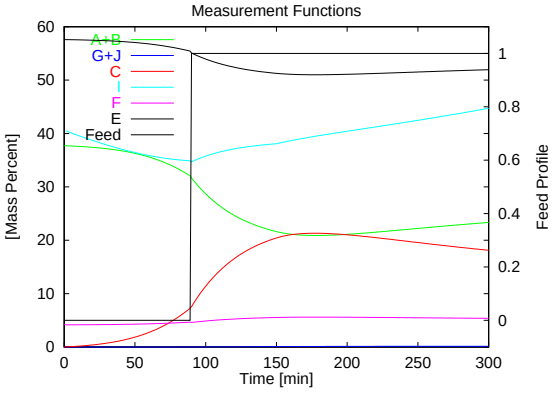
Temperatur:
$295\,K \rightarrow 360\,K \rightarrow 379\,K$
Destillation: $4.9\,g/min$
Messungen bei
130, 150, 170, 190, 210,
220, 230, 250, 275, 300

Experiment 4:



Temperatur:
$295\,K \rightarrow 383\,K \rightarrow 383\,K$
Destillation: $0\,g/min$
Messungen bei
100, 110, 120, 130, 150,
170, 190, 220, 230, 250

Experiment 5:



Temperatur:
$295\,K \rightarrow 331\,K \rightarrow 301\,K$
Destillation: $0.01\,g/min$
Messungen bei
80, 100, 110, 120, 130,
220, 230, 250, 275, 300

Abbildung 10.16: Beschreibung der drei durch optimale Versuchsplanung berechneten Experimente für das Veresterungsbeispiel

10.4 Eine Bimolekulare Katalyse: Die Diels-Alder-Reaktion

Anhand des Beispiels der Diels-Alder-Reaktion untersuchen wir nun die Methode der robusten Versuchsplanung unter Einbeziehung der Parameterabhängigkeit, siehe Abschnitt 8.2 in Kapitel 8. Die numerischen Ergebnisse zeigen, daß dieser Ansatz es ermöglicht, Versuchspläne zu berechnen, die deutlich weniger sensitiv von Unsicherheiten der Parameter abhängen als bei der nichtrobusten Vorgehensweise.

10.4.1 Das chemische Modell und der Prozeß

Die Diels-Alder-Reaktion ist ein wichtiges Beispiel einer Zykloadditionsreaktion, siehe [79]. α, β-ungesättigte Karbonylverbindungen reagieren dabei mit konjugierten Dienen. Dabei entsteht ein 6-Ring. Reaktionen dieses Typs finden mit verschiedenartigen Reaktanten statt. Ein konkretes Beispiel wird in Abbildung 10.17 dargestellt. Katalysator ist Aluminiumchlorid.

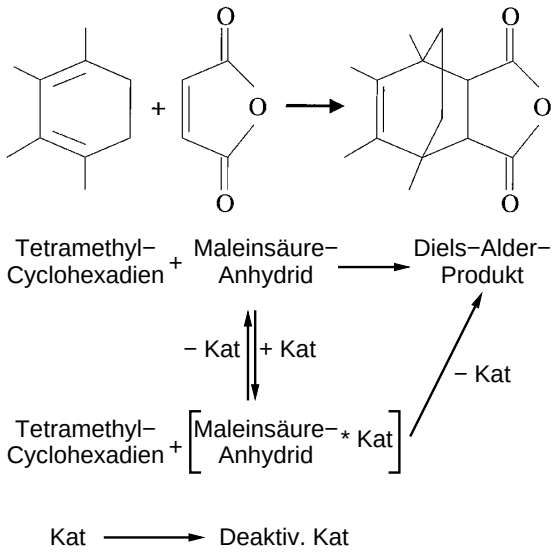


Abbildung 10.17: Reaktionsmechanismus einer Diels-Alder-Reaktion

Die Reaktionsgleichung der Gesamtreaktion lautet

$$A + B \longrightarrow C$$

(A: Tetramethyl-Cyclohexadien, B: Maleinsäure-Anhydrid, C: Diels-Alder-Produkt) mit Katalysator K in Lösungsmittel L.

Der Katalysator baut sich ab mit Abbaurate λ:

$$K \xrightarrow{\lambda} K'.$$

Die Reaktion wird in einem Rührkessel durchgeführt als isothermer Batchprozeß mit Einwaage der Edukte A und B im Lösungsmittel L mit Katalysator K.

Gemessen werden kann das Reaktionsprodukt C.

10.4.2 Das Mathematische Modell

Das Differentialgleichungssystem

Experimente ergeben, daß sich die Reaktionskinetik nach dem folgenden Zeitgesetz modellieren läßt:

$$\begin{aligned} \dot{n}_1 &= -k \cdot \frac{n_1 \cdot n_2}{m_{ges}} \\ \dot{n}_2 &= -k \cdot \frac{n_1 \cdot n_2}{m_{ges}} \\ \dot{n}_3 &= k \cdot \frac{n_1 \cdot n_2}{m_{ges}}. \end{aligned}$$

Anfangswerte der Molzahlen sind die Einwaagen:

$$\begin{aligned} n_1(0) &= n_{a1} \\ n_2(0) &= n_{a2} \\ n_3(0) &= 0. \end{aligned}$$

Die Geschwindigkeitskonstante k hat einen Anteil für den nichtkatalysierten und einen Anteil für den katalysierten Reaktionsweg. Als Geschwindigkeitsgesetz wird jeweils die Arrhenius-Beziehung angesetzt:

$$\begin{aligned} k &= k_1 \cdot \exp\left(-\frac{E_1}{R} \cdot \left(\frac{1}{T} - \frac{1}{T_{ref}}\right)\right) \\ &+ k_{kat} \cdot c_{kat} \cdot \exp\left(-\lambda \cdot t\right) \cdot \exp\left(-\frac{E_{kat}}{R} \cdot \left(\frac{1}{T} - \frac{1}{T_{ref}}\right)\right). \end{aligned}$$

Die Molzahl des Lösungsmittels ist

$$n_4 = n_{a4},$$

die Gesamtmasse

$$m_{ges} = n_1 \cdot M_1 + n_2 \cdot M_2 + n_3 \cdot M_3 + n_4 \cdot M_4$$

und die Kelvintemperatur

$$T = \vartheta + 273.$$

Auftretende Größen

Die folgenden Größen kommen im Differentialgleichungssystem vor:

- Zustandsgrößen: Molzahlen n_1, n_2, n_3,
- Parameter: Frequenzfaktoren k_1, k_{kat}, Aktivierungsenergien E_1, E_{kat}, Katalysatorabbaukoeffizient λ,
- Steuergrößen: Anfangsmolzahlen $n_{a1},\ n_{a2},\ n_{a4} \in [0.4, 9]$, Katalysatorkonzentration $c_{kat} \in [0, 6]$, Celsiustemperatur $\vartheta \in [20, 100]$,
- Konstanten: Molmassen $M_1 = 0.1362,\ M_2 = 0.09806,\ M_3 = 0.23426,\ M_4 = 0.236$, allgemeine Gaskonstante $R = 8.314$, Referenztemperatur $T_{ref} = 293$.

Das Meßmodell

Als Meßverfahren steht eine High Performance Liquid Chromatography (HPLC) [66] zur Verfügung, mit der Massenprozent des Reaktionsprodukts gemessen werden können:

$$h = \frac{n_3 \cdot M_3}{m_{ges}} \cdot 100.$$

Die Standardabweichung des Meßfehlers beträgt dabei

$$\varsigma = 1.$$

An den folgenden Zeitpunkten sind Messungen möglich:

$$t_j = j/3, \quad j = 1, \ldots, 15, \qquad t_j = j - 10, \quad j = 16, \ldots, 20.$$

Durch die Versuchsplanung sollen pro Experiment sechs Messungen aus den 20 möglichen ausgewählt werden.

Nebenbedingungen

An die Steuergrößen sind Nebenbedingungen gestellt, und zwar bezüglich der Masse der Einwaage

$$0.1 \leq n_{a1} \cdot M_1 + n_{a2} \cdot M_2 + n_{a4} \cdot M_4 \leq 10$$

und des Wirkstoffgehalts

$$10 \leq \frac{n_{a1} \cdot M_1 + n_{a2} \cdot M_2}{n_{a1} \cdot M_1 + n_{a2} \cdot M_2 + n_{a4} \cdot M_4} \leq 70.$$

10.4.3 Nichtrobuste und robuste optimale Versuchsplanung

Ziel der Untersuchungen anhand der Diels-Alder-Reaktion war der Vergleich zwischen nichtrobuster und robuster Versuchsplanung.

Für die Parameter wurden die folgenden Werte angenommen:

$$\begin{aligned} k_1 &= p_{0,1} \cdot 0.01, \\ E_1 &= p_{0,2} \cdot 60000, \\ k_{kat} &= p_{0,3} \cdot 0.10, \\ E_{kat} &= p_{0,4} \cdot 40000, \\ \lambda &= p_{0,5} \cdot 0.25, \end{aligned}$$

wobei p_0 auf eins skaliert ist: $p_{0,j} = 1$, $j = 1, \ldots, 5$.

Zunächst wurde mit nichtrobuster Versuchsplanung, also mit der Zielfunktion

$$\min_{\xi \in \Omega} \phi(C(\xi, p_0)),$$

ein Vierfachexperiment-Versuchsplan berechnet. Als Gütekriterium ϕ wurde das A-Kriterium verwendet.

Das Optimierungsproblem hatte dabei pro Experiment fünf Steuergrößen und 20 Gewichte zur Auswahl der Messungen, also insgesamt 100 Variablen.

In Abbildung 10.18 sind die Ergebnisse der Optimierung, also Steuergrößen und Meßzeitpunkte sowie die Trajektorien für die Molzahlen der vier Experimente, dargestellt. Für dieses Vierfachexperiment ergeben sich für die Parameter die Standardabweichungen

$$\begin{aligned} k_1 : &\pm 2.1\% \\ E_1 : &\pm 0.8\% \\ k_{kat} : &\pm 2.9\% \\ E_{kat} : &\pm 3.4\% \\ \lambda : &\pm 6.1\%, \end{aligned}$$

das A-Kriterium beträgt 0.00124372.

Ausgehend vom nichtrobusten Versuchsplan wurde dann mit der Zielfunktion

$$\min_{\xi \in \Omega} \phi(C(\xi, p_0)) + \gamma \left\| \frac{\mathrm{d}}{\mathrm{d}p} \phi(C(\xi, p_0)) \right\|_{2,\Sigma}$$

ein robuster Versuchsplan unter Berücksichtigung der Parameterunsicherheit berechnet, siehe Abschnitt 8.2 in Kapitel 8. Dabei wählten wir $\Sigma = I$ und $\gamma = \chi_5(1 - 0.95)$ für einen Konfidenzlevel von 95%. Zur Berechnung der Differenzenquotienten verwendeten wir $\Delta p_j = 1 \cdot 10^{-5}$, $j = 1, \ldots, 5$.

Auch hier hatte das Vierfachexperiment-Optimierungsproblem 100 Variablen.

Experiment	n_{a1}	n_{a2}	n_{a4}	c_{kat}	ϑ	Probenahmen bei
1	25.66	30.40	0.4	0	25.98	5, 6, 7, 8, 9, 10
2	41.50	43.37	0.4	1.71	20	0.66, 1, 1.33, 8, 9, 10
3	15.78	15.93	0.4	0.99	40.86	0.33, 0.66, 7, 8, 9, 10
4	10.73	10.73	0.4	0	88.60	0.33, 0.66, 1, 1.33, 1.66, 2

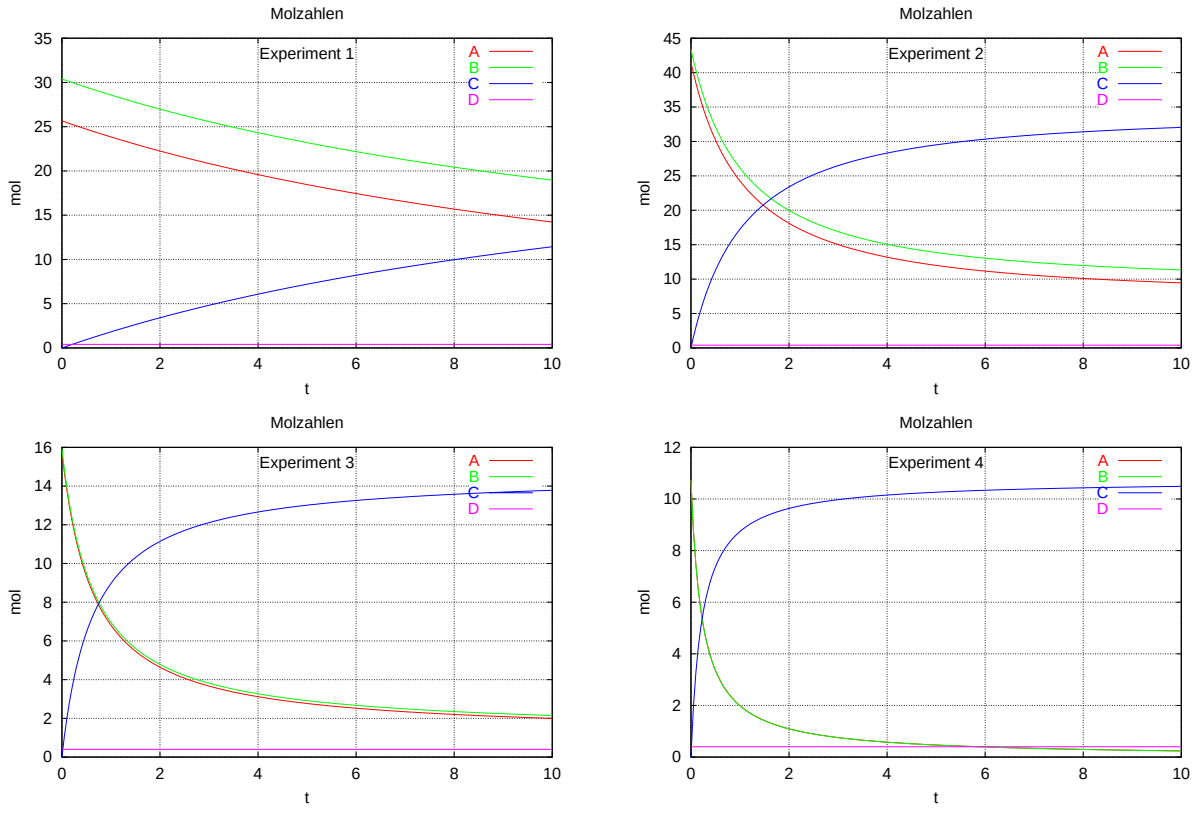


Abbildung 10.18: Steuergrößen, Messungen und Trajektorien des optimalen, nichtrobusten Vierfachexperiments für die Diels-Alder-Reaktion

Experiment	n_{a1}	n_{a2}	n_{a4}	c_{kat}	ϑ	Probenahmen bei
1	39.55	46.09	0.4	0	20	5, 6, 7, 8, 9, 10
2	41.45	43.45	0.4	1.90	20	0.33, 0.66, 1, 8, 9, 10
3	41.88	42.84	0.4	1.35	31.92	0.33, 0.66, 7, 8, 9, 10
4	40.54	44.71	0.4	0	32.21	0.66, 1, 1.33, 1.6, 2, 2.33

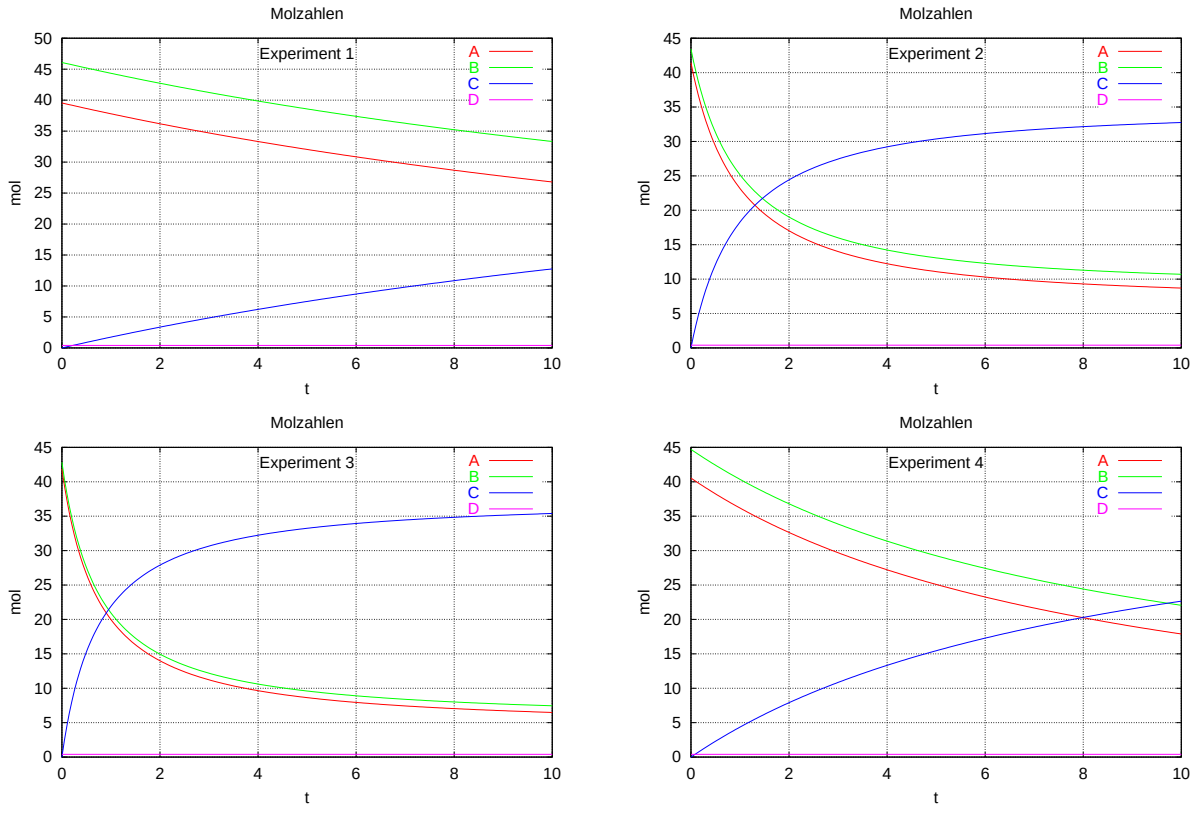


Abbildung 10.19: Steuergrößen, Messungen und Trajektorien des optimalen, robusten Vierfachexperiments für die Diels-Alder-Reaktion

Die Ergebnisse der robusten Optimierung sind in Abbildung 10.19 zusammengestellt. Für den Parametersatz p_0 ergeben sich daraus die Standardabweichungen

$$\begin{aligned} k_1 : &\pm 2.2\% \\ E_1 : &\pm 4.0\% \\ k_{kat} : &\pm 2.9\% \\ E_{kat} : &\pm 5.6\% \\ \lambda : &\pm 5.6\% \end{aligned}$$

und das A-Kriterium 0.00184572. Dies ist erwartungsgemäß schlechter als für den nichtrobusten Versuchsplan.

Zur Beurteilung der Robustheit dieses Versuchsplans werten wir die Standardabweichungen und das A-Kriterium jedoch nicht nur in p_0 aus, sondern auch für gestörte Werte in der Umgebung von p_0. Abbildung 10.20 zeigt hierbei exemplarisch anhand einiger Schnitte durch den (fünfdimensionalen) Parameterraum den Vergleich zwischen robustem und nichtrobustem Design. Die Schätzgüte beim robusten Versuchsplan ist dabei wesentlich unempfindlicher gegen Änderungen der Parameterwerte.

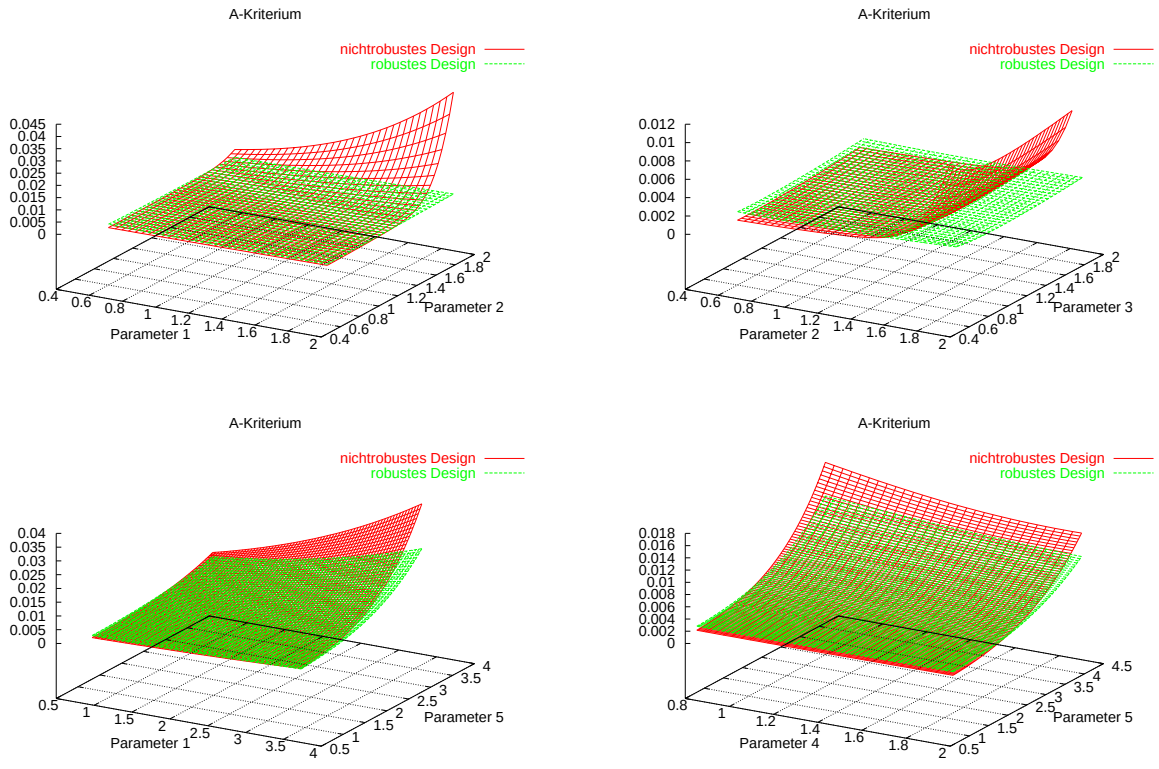


Abbildung 10.20: Vergleich zwischen robustem (grün) und nichtrobustem (rot) Design für die Diels-Alder-Reaktion. Durch die robuste Versuchsplanung wird die Empfindlichkeit des Gütekriteriums auf Parameteränderungen deutlich reduziert.

Kapitel 11

Schlußbemerkungen

Mit dieser Arbeit und der zugehörigen Software VPLAN stellen wir ein umfassendes Werkzeug zur Behandlung einer allgemeinen Problemklasse zur Verfügung.

Wir betrachten Prozesse, die durch nichtlineare Differentiell-Algebraische Gleichungssysteme modelliert werden. Zu deren Lösung verwenden wir BDF-Verfahren.

Zur Modellvalidierung lösen wir nichtlineare beschränkte Parameterschätzprobleme. Die DAE-Modellgleichungen treten dabei als Nebenbedingungen auf. Wir parametrisieren diese zum Beispiel mit der Mehrzielmethode und lösen die nichtlinearen beschränkten Least-Squares-Probleme mit Verallgemeinerten Gauß-Newton-Verfahren. Die Sensitivitätsanalyse in deren Lösungspunkt ergibt Näherungen der Konfidenzgebiete der Parameterschätzung. Ein Maß für die Signifikanz der Schätzung sind Gütekriterien auf der Kovarianzmatrix.

Indem wir diese minimieren, gelangen wir zur Formulierung Nichtlinerer Optimaler Versuchsplanungsprobleme mit Prozeßsteuerungen und Meßentwurfsgrößen als Variablen. Diese sind beschränkte Optimalsteuerungsprobleme, wiederum mit dem zugrundeliegenden DAE-System als Nebenbedingung. Zur Parametrisierung der Steuerfunktionen verwenden wir den Direkten Ansatz. Für die Optimierung setzen wir SQP-Verfahren ein, was für die Bereitstellung der benötigten Gradienten eine spezielle Ableitungserzeugung erforderlich macht, da die Zielfunktion bereits von Sensitivitäten der Lösung der Modellfunktionen nach den Parametern abhängt. Wir berechnen die Ableitungen durch Matrixableitungskalkül in Verbindung mit Interner Numerischer Differentiation und Automatischer Differentiation. Die Ganzzahligkeit der Meßentwurfsgrößen behandeln wir mit geeigneten Relaxierungen und Heuristiken.

Bei Parameterschätzung und Versuchsplanung treten häufig Mehrfachexperimentaufgaben auf. Wir nutzen die Strukturen, die sich durch diese Formulierung ergeben, effizient aus. Außerdem ermöglicht dieser Ansatz die Berücksichtigung von Vorexperimenten, die als festgehaltene Bestandteile eines Mehrfachexperimnts zwar in die Berechnung der Kovarianzmatrix, nicht aber in die Optimierung eingehen.

Zur Robusten Versuchsplanung, daß heißt der Berücksichtigung der Parameterstreuung, schlagen wir zwei Ansätze vor. Einerseits eine sequentielle Vorgehensweise aus Optimaler Versuchsplanung, Versuchsdurchführung und Parameterschätzung, andererseits einen

Worst-Case-Ansatz zur Berechnung des besten Versuchsplans unter gegebener Parameterverteilung.

Die in der Arbeit behandelten Methoden wurden im Softwarepaket VPLAN implementiert. Bei dessen Entwicklung wurde auf Problemunabhängigkeit und Bedienungsfreundlichkeit Wert gelegt.

Methodik und Software wurden auf verschiedene Probleme angewendet. Anhand vier ausgewählter Praxisbeispiele demonstrieren wir den erfolgreichen Einsatz. Als erstes Beispiel zeigt die Phosphinreaktion, daß mit sequentieller Versuchsplanung und Parameterschätzung Experimente bestimmt werden können, die wesentlich signifikantere Schätzungen erlauben als eine vergleichbare intuitive Vorgehensweise. Bei der Urethanreaktion, die ein wesentlich komplexeres Reaktionssystem und eine praxisnahe Modellierung beinhaltet, fällt dieser Vergleich noch viel drastischer zugunsten unserer Methodik aus. Zwei optimal geplante Experimente liefern hier wesentlich genauere Schätzergebnisse als fünfzehn intuitiv entworfene. Das Beispiel einer Veresterungsreaktion ist ein komplexes mehrphasiges Reaktionssystem. Hier kann ein Mehrfachexperiment-Versuchsplan berechnet werden, mit dem alle Modellparameter mit der in der Praxis geforderten Genauigkeit geschätzt werden können. Am Beispiel einer Bimolekularen Katalyse demonstrieren wir schließlich, daß mit Robuster Versuchsplanung Experimente entworfen werden können, die unempfindlich gegen die Parameterunsicherheiten sind.

Unsere Software VPLAN wird erfolgreich in der industriellen Praxis eingesetzt. Mit ihr liegt erstmals ein Werkzeug vor, mit dem aus einer Hand Modellierung, Simulation, Parameterschätzung und Optimale Versuchsplanung für nichtlineare DAE-Systeme durchgeführt werden können — ein Virtuelles Laboratorium für dynamische Prozesse.

Literaturverzeichnis

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide Third Edition*. SIAM, 1999.

[2] U. M. Ascher and L. R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. SIAM, 1998.

[3] S. P. Asprey and S. Macchietto. Statistical tools for optimal dynamic model building. *Computers and Chemical Engineering*, 24:1261–1267, 2000.

[4] P. W. Atkins. *Physikalische Chemie*. Verlag Chemie, 1996.

[5] A. C. Atkinson. Recent Developments in the Methods of Optimum and Related Experimental Designs. *International Statistical Review*, 56(2):99–115, 1988.

[6] A. C. Atkinson. The Usefulness of Optimum Experimental Design. *J. R. Statist. Soc. B*, 58(1):59–76, 1996.

[7] A. C. Atkinson and A. N. Donev. *Optimum Experimental Designs*. Oxford University Press, 1992.

[8] A. C. Atkinson and V. V. Fedorov. Optimal design: Experiments for discriminating between several models. *Biometrika*, 62(2):289, 1975.

[9] A. C. Atkinson and S. E. Fienberg, editors. *A Celebration of Statistics*, chapter 20: A. C. Atkinson: An Introduction to the Optimum Design of Experiments, pages 465–473. Springer, 1985.

[10] M. Baltes, R. Schneider, C. Sturm, and M. Reuss. Optimal Experimental Design for Parameter Estimation in Unstructured Growth Models. *Biotechnol. Prog.*, 10:480–488, 1994.

[11] Y. Bard. *Nonlinear Parameter Estimation*. Academic Press, Inc., 1974.

[12] G. M. Barrow. *Physikalische Chemie*. Bohmann Vieweg, 1984.

[13] D. M. Bates and D. G. Watts. *Nonlinear regression analysis and its applications*. Series in Probability and Mathematical Statistics. Wiley, 1988.

[14] I. Bauer. Numerische Verfahren zur Lösung von Anfangswertaufgaben und zur Generierung von ersten und zweiten Ableitungen mit Anwendungen in Chemie und Verfahrenstechnik. Preprint, SFB 359, Universität Heidelberg, 2001.

[15] I. Bauer, H. G. Bock, S. Körkel, and J. P. Schlöder. Numerical methods for initial value problems and derivative generation for DAE models with application to optimum experimental design of chemical processes. In F. Keil, W. Mackens, H. Voss, and J. Werther, editors, *Scientific Computing in Chemical Engineering II*, volume 2, pages 282–289, Berlin, Heidelberg, 1999. Springer-Verlag.

[16] I. Bauer, H. G. Bock, S. Körkel, and J. P. Schlöder. Numerical methods for optimum experimental design in DAE systems. *Journal of Computational and Applied Mathematics*, 120:1–25, 2000.

[17] I. Bauer, H. G. Bock, and J. P. Schlöder. DAESOL — a BDF code for the numerical solution of differential-algebraic equations. Preprint, IWR der Universität Heidelberg, SFB 359, November 1999.

[18] I. Bauer, M. Heilig, S. Körkel, A. Kud, A. Mayer, and O. Wörz. Versuchsplanung am Beispiel einer Phosphin- und Urethanreaktion. In Optimale Versuchsplanung für nichtlineare Prozesse [87].

[19] I. Bauer, S. Körkel, H. G. Bock, and J. P. Schlöder. Optimale Versuchsplanung für dynamische Systeme aus der chemischen Reaktionskinetik. In Optimale Versuchsplanung für nichtlineare Prozesse [87].

[20] I. Bauer, S. Körkel, H. G. Bock, and J. P. Schlöder. BMBF-Projekt: Optimale Versuchsplanung für nichtlineare Prozesse in Reaktionskinetik und chemischer Verfahrenstechnik: Entwicklung mathematischer Methoden und Algorithmen. Schlußbericht, IWR der Universität Heidelberg, 1999.

[21] C. Bischof, A. Carle, G. Corliss, A. Griewank, and P. Hovland. ADIFOR - generating derivative codes from fortran programs. *Scientific Computing*, 1(1):1–29, 1992.

[22] C. Bischof, A. Carle, P. Hovland, P. Khademi, and A. Mauer. ADIFOR 2.0 user's guide (revision D),. Technical Report CRPC-TR95516-S, Center for Research on Parallel Computation, Rice University, Houston, TX, 1995, revised 1998.

[23] C. Bischof, A. Carle, P. Khademi, and A. Mauer. The ADIFOR 2.0 system for the automatic differentiation of fortran 77 programs. Technical Report CRPC-TR94491, Center for Research on Parallel Computation, Rice University, Houston, TX, 1994.

[24] H. G. Bock. Numerical Treatment of Inverse Problems in Chemical Reaction Kinetics. In K. H. Ebert, P. Deuflhard, and W. Jäger, editors, *Modelling of Chemical Reaction Systems*, Springer Series in Chemical Physics, Heidelberg, 1981.

[25] H. G. Bock. Recent Advances in Parameteridentification Techniques for O.D.E. In P. Deuflhard and E. Hairer, editors, *Numerical Treatment of Inverse Problems in Differential and Integral Equations*, pages 95–121. Birkhäuser, 1983.

[26] H. G. Bock. Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen. *Bonner Mathematische Schriften 183*, 1987.

[27] H. G. Bock, E. Eich, and J. P. Schlöder. Numerical solution of constrained least squares boundary value problems in differential-algebraic equations. In K. Strehmel, editor, *Numerical Treatment of Differential Equations*. Teubner, Leipzig, 1988.

[28] H. G. Bock, J. P. Schlöder, and V. H. Schulz. Numerik großer Differentiell-Algebraischer Gleichungen, Simulation und Optimierung. In H. Schuler, editor, *Prozeß-Simulation*. Verlag Chemie, 1995.

[29] H. G. Bock, M. Zieße, J. Gallitzendörfer, and J. P. Schlöder. Parameter estimation in multispecies transport reaction systems using parallel algorithms. In J. Gottlieb and P. DuChateaux, editors, *Parameter Identification and Inverse Problems in Hydrology, Geology and Ecology*. Kluwer, 1996.

[30] P. T. Boggs, J. W. Tolle, and P. Wang. On the local convergence of quasi-Newton methods for constrained optimization. *SIAM J. Control and Optimization*, 20(2):161–171, 1982.

[31] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Classics in Applied Mathematics. SIAM, Philadelphia, PA, 1996.

[32] P. Chaudhuri and P. A. Mykland. Nonlinear Experiments: Optimal Design and Inference Based on Likelihood. *J. Am. Stat. Assoc.*, 88(422):538–546, 1993.

[33] D. Cook and V. Fedorov. Constrained Optimization Of Experimental Design. *Statistics*, 26:129–178, 1995.

[34] P. Deuflhard and F. Bornemann. *Numerische Mathematik II, Integration gewöhnlicher Differentialgleichungen*. de Gruyter Lehrbuch, Berlin, New York, 1994.

[35] A. Dieses. Numerische Verfahren zur Diskriminierung nichtlinearer Modelle für dynamische chemische Prozesse. Diplomarbeit, Interdisziplinäres Zentrum für Wissenschaftliches Rechnen der Universität Heidelberg, 1997.

[36] A. Dieses. *Numerical Methods for Optimization Problems in Water Flow and Reactive Solute Transport Processes of Xenobiotics on Soils*. Dissertation, Universität Heidelberg, Dezember 2000.

[37] V. G. Dovíˊ, A. P. Reverberi, and L. Acevedo-Duarte. New Procedure for Optimal Design of Sequential Experiments in Kinetic Models. *Ind. Eng. Chem. Res.*, 33:62–68, 1994.

[38] N. R. Draper and W. G. Hunter. The use of prior distributions in the design of experiments for parameter estimation in nonlinear situations. *Biometrika*, 54(1):147, 1967.

[39] N. R. Draper and F. Pukelsheim. An overview of design of experiments. *Statistical Papers, Springer-Verlag*, 37:1–32, 1996.

[40] N. R. Draper and H. Smith. *Applied Regression Analysis*. John Wiley, New York, 2nd edition, 1981.

[41] E. Eich. Numerische Behandlung semi-expliziter differentiell-algebraischer Gleichungssysteme vom Index 1 mit BDF-Verfahren. Diplomarbeit, Universität Bonn, 1987.

[42] M. Estler, A. S. Bommarius, H. Werner, H. Luft, E. Kostina, H. G. Bock, and J. P. Schlöder. Optimum experimental design for the determination of enzyme operating stability. In *Proc. 3rd European Congress of Chemical Engineering*, 2001.

[43] P. Cederqvist et al. *Version Management with CVS*. Signum Support AB, available online: http://www.cvshome.org/docs/manual/, 1993.

[44] V. V. Fedorov. *Theory of Optimal Experiments*. Probability And Mathematical Statistics. Academic Press, London, 1972.

[45] V. V. Fedorov and P. Hackl. Optimal Experimental Design: Spatial Sampling. *Calcutta Statistical Association Bulletin*, 44(173-174):57, 1994.

[46] V. V. Fedorov and P. Hackl. *Model-Oriented Design of Experiments*, volume 125 of *Lecture Notes in Statistics*. Springer, 1997.

[47] R. Fletcher. *Practical methods of optimization*. Wiley, 1988.

[48] C. A. Floudas. *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*. Oxford University Press, 1995.

[49] I. Ford, B. Torsney, and C. F. J. Wu. The Use of a Canonical Form in the Construction of Locally Optimal Designs for Non-linear Problems. *J. R. Statist. Soc. B*, pages 569–583, 1992.

[50] J. Frehse. Existence of optimal controls I. *Methods of Operations Research*, 30:26–59, 1979.

[51] K. C. Frisch and H. C. Vogt. Polyurethanes and Related Isocyanate Polymers. In E.M. Fettes, editor, *Chemical Reactions of Polymers*, volume 19 of *High Polymers Series*. Wiley Interscience, 1964.

[52] J. Gallitzendörfer. *Parallel Algorithms for optimization boundary value Problems in DAE*. Dissertation, Universität Heidelberg, 1994.

[53] A. Gifi. *Nonlinear Multivariate Analysis*. John Wiley, New York, 1990.

[54] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. Report NA 97-2, Dept. of Mathematics, University of California, San Diego, 1997.

[55] P. E. Gill, W. Murray, and M. A. Saunders. *User's Guide For SNOPT 5.3: A FORTRAN Package for Large-Scale Nonlinear Programming*, December 1998.

[56] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. Contrained nonlinear programming. In G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, editors, *Handbooks in Operations Research and Management Science: Optimization*, volume 1, chapter III, pages 171–210. North-Holland, 1989.

[57] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Academic Press, 1981.

[58] A. Griewank. *Evaluating Derivatives. Principles and Techniques of Algorithmic Differentiation.* Frontiers in Applied Mathematics. SIAM, 2000.

[59] A. Griewank, D. Juedes, H. Mitev, J. Utke, O. Vogel, and A. Walther. ADOL-C: A package for the automatic differentiation of algorithms written in C/C++. *ACM TOMS*, 2(22):131–167, 1996.

[60] L. M. Haines. Optimal Design For Nonlinear Regression Models. *Commun. Statist.-Theory Meth.*, 22(6):1613–1627, 1993.

[61] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I. Nonstiff Problems.* Springer Series in Computational Mathematics. Springer, 2nd edition, 1993.

[62] E. Hairer and G. Wanner. On the instability of the BDF formulas. *SIAM J. Numer. Anal.*, 20:1206–1209, 1983.

[63] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems.* Springer Series in Computational Mathematics. Springer, 2nd edition, 1996.

[64] R. Hettich and K. O. Kortanek. Semi-infinite programming: Theory, methods, and applications. *SIAM Review*, 35(3):380–429, 1993.

[65] K.-D. Hilf. *Optimale Versuchsplanung zur dynamischen Roboterkalibrierung*, volume 8 of *Fortschrittberichte.* VDI, 1996.

[66] High Performance Liquid Chromatography (HPLC): A Users Guide. Analytical Spectroscopy Group, University of Kentucky, Lexington, KY. Online verfügbar http://www.pharm.uky.edu/ASRG/HPLC/hplcmytry.html.

[67] J. Huber. Large-Scale SQP-Methods and Cross-Over-Techniques for Quadratic Programming. Diplomarbeit, Universität Heidelberg, 1998.

[68] S. Körkel, I. Bauer, H. G. Bock, and J. P. Schlöder. A sequential approach for nonlinear optimum experimental design in DAE systems. In F. Keil, W. Mackens, H. Voss, and J. Werther, editors, *Scientific Computing in Chemical Engineering II*, volume 2, pages 338–345, Berlin, Heidelberg, 1999. Springer-Verlag.

[69] E. Kostina. To the robust design. Persönliche Mitteilung, 2001.

[70] I. Leichsenring. Numerische Behandlung von Parameterschätzproblemen mit Mehrfachexperiment-Struktur in der Pharmakokinetik. Diplomarbeit, Universität Heidelberg, Fakultät für Mathematik, 1995.

[71] A. C. Ponce De Leon and A. C. Atkinson. Optimum experimental design for discriminating between two rival models in the presence of prior information. *Biometrika*, 78(3):601, 1991.

[72] J. Leonhard, O. Lade, and P. Hugo. Thermokinetische Auswertung von adiabatischen Semibatch-Messungen. Technical report, TU Berlin, Institut für Technische Chemie, 1997. Online verfügbar http://www.tu-berlin.de/ itc/hugo/asbr/asbr.htm.

[73] T. W. Lohmann. Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen, Modellbildung durch Analyse der Kovarianzmatrix der Parameterschätzung. Diplomarbeit, Universität Bonn, 1988.

[74] T. W. Lohmann. *Ein numerisches Verfahren zur Berechnung optimaler Versuchspläne für beschränkte Parameteridentifizierungsprobleme*. Reihe Informatik. Verlag Shaker, Aachen, 1993.

[75] T. W. Lohmann, H. G. Bock, and J. P. Schlöder. Numerical Methods for Parameter Estimation and Optimal Experiment Design in Chemical Reaction Systems. *Ind. Eng. Chem. Res.*, 31:54–57, 1992.

[76] The Math Works, Inc. *Using Matlab*, 1998.

[77] G. P. McCormick. *Nonlinear Programming*. John Wiley & Sons, 1983.

[78] R. Mead. *The design of experiments: statistical principles for practical application*. Cambridge University Press, 1988.

[79] R. T. Morrison and R. N. Boyd. *Organic Chemistry*. Allyn and Bacon, Inc., 4th edition, 1983.

[80] E. Müller-Erlwein. *Chemische Reaktionstechnik*. Teubner, 1998.

[81] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. Series in Discrete Mathematics and Optimization. Wiley-Interscience, 1988.

[82] R. Nishii. Optimality of experimental designs. *Discrete Mathematics*, 116:209–225, 1993.

[83] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer-Verlag, New York, 1999.

[84] The Numerical Algorithms Group Limited. *NAG Fortran Library Manual*.

[85] T. E. O'Brien and J. O. Rawlings. A Non-Sequential Design Procedure for Parameter Estimation and Model Discrimination in Nonlinear Regression Models. *Journal of Statistical Planning and Inference*, 1996.

[86] P. Oinas, I. Turunen, and H. Haario. Experimental Design With Steady-State And Dynamic Models Of Multiphase Reactors. *Chemical Engineering Science*, 47(13/14):3689–3696, 1992.

[87] Optimale Versuchsplanung für nichtlineare Prozesse. Schriftliche Projektpräsentation zum BMBF-Verbundvorhaben. Dechema e. V., Frankfurt am Main, 14. 10. 1998.

[88] J. M. Ortega and W. C. Rheinboldt. On discretization and differentiation of operators with application to Newton's method. *J. SIAM Numer. Anal.*, 3(1):143–156, 1966.

[89] R. Penrose. A generalized inverse for matrices. *Proc. Cambridge Philosoph. Soc.*, 51:406–413, 1955.

[90] L. R. Petzold. A description of DASSL: A differential/algebraic system solver. In R. S. Stepleman et al., editor, *Scientific Computing*, pages 65–68. North-Holland, Amsterdam, 1983.

[91] M. J. Pilling and P. W. Seakins. *Reaction Kinetics*. Oxford Science Publications. Oxford University Press, 1997.

[92] L. Pronzato and E. Walter. Robust experiment designs for nonlinear regression models. In V. Fedorov and H. Läuter, editors, *Model-Oriented Data Analysis*, Lecture Notes in Economics and Mathematical Systems, Eisenach, 1987. Springer.

[93] F. Pukelsheim. *Optimal Design of Experiments*. John Wiley & Sons, Inc., New York, 1993.

[94] F. Pukelsheim and S. Rieder. Efficient rounding of approximate designs. *Biometrika*, 79(4):763–770, 1992.

[95] Z. H. Qureshi, T. S. Ng, and G. C. Goodwin. Optimum experimental design for identification of distributed parameter systems. *Int. J. Control*, 31(1):21–29, 1980.

[96] D. Rasch. The robustness against parameter variation of exact locally optimum designs in nonlinear regression — a case study. *Computational Statistics & Data Analysis*, 20:441–453, 1995.

[97] A. Raum. Numerische Methoden für die Optimale Versuchsplanung bei nichtlinearen Problemen. Diplomarbeit, Universität Heidelberg, Fakultät für Mathematik, 1997.

[98] P. M. Reilly, R. Bajramovic, G. E. Blau, D. R. Branson, and M. W. Sauerhoff. Guidelines for the Optimal Design of Experiments to Estimate Parameters in First Order Kinetic Models. *The Canadian Journal of Chemical Engineering*, 55:614, 1977.

[99] H. Renon and J. M. Prausnitz. Local composition in thermodynamic, excess functions for liquid mixtures. *AIChE Journal*, 14(1):135–144, 1968.

[100] G. Rücker. Automatisches Differenzieren mit Anwendung in der Optimierung bei chemischen Reaktionssystemen. Diplomarbeit, Universität Heidelberg, Dezember 1999.

[101] P. E. Rudolph and G. Herrendörfer. Optimal Experimental Design and Accuracy of Parameter Estimation for Nonlinear Regression Models Used in Long-term Selection. *Biom. J.*, 37(2):183–190, 1995.

[102] A. Schäfer, H. G. Bock, J. P. Schlöder, and D. B. Leineweber. An exact Hessian SQP method for multistage Optimal Control Problems. Preprint, IWR der Universität Heidelberg, 2002.

[103] W. E. Schiesser. *The Numerical Method of Lines, Integration of Partial Differential Equations*. Academic Press, 1991.

[104] J. P. Schlöder. *Numerische Methoden zur Behandlung hochdimensionaler Aufgaben der Parameteridentifizierung*. Dissertation, Hohe Mathematisch-Naturwissenschaftliche Fakultät der Rheinischen Friedrich-Wilhelms-Universität zu Bonn, 1987.

[105] V. H. Schulz. Ein effizientes Kollokationsverfahren zur numerischen Behandlung von Mehrpunktrandwertaufgaben in der Parameteridentifizierung und Optimalen Steuerung. Diplomarbeit, Universität Augsburg, 1990.

[106] V. H. Schulz, H. G. Bock, and M. C. Steinbach. Exploiting invariants in the numerical solution of multipoint boundary value problems for DAE. *SIAM Journal on Scientific Computing*, 19(2):440–467, 1998.

[107] G. A. F. Seber and C. J. Wild. *Nonlinear Regression*. John Wiley & Sons, Inc., 1989.

[108] P. Spellucci. *Numerische Verfahren der nichtlinearen Optimierung*. ISNM Lehrbuch. Birkhäuser, 1993.

[109] J. Stoer and R. Bulirsch. *Numerische Mathemtik 2*. Springer, 1973.

[110] Bjarne Stroustrup. *The C++ Programming Language (3rd Edition)*. Addison-Wesley, 1997.

[111] H. C. van Ness, S. M. Byer, and R. E. Gibbs. Vapor-liquid equilibrium. *AIChE Journal*, 19(2):238–244, 1973.

[112] M. von Löwis and N. Fischbeck. *Das Python-Buch*. Addison-Wesley, 1997.

[113] M. von Schwerin. *Numerische Methoden zur Schätzung von Reaktionsgeschwindigkeiten bei der katalytischen Methankonversion und Optimierung von Essigsäure- und Methanprozessen*. Dissertation, Universität Heidelberg, 1998.

[114] R. von Schwerin. *Numerical Methods, Algorithms, and Software for Higher Index Differential-Algebraic Equations in MultiBody System SIMulation*. Dissertation, IWR der Universität Heidelberg, 1997.

[115] K. Walrath and M. Campione. *The JFC Swing Tutorial: A Guide to Constructing GUIs*. Addison-Wesley, 1999.

[116] G. Wedler. *Lehrbuch der Physikalischen Chemie*. Wiley-VCH, 4. Auflage, 1997.

[117] T. Williams and C. Kelley. *gnuplot — An Interactive Plotting Program*. Online Documentation http://www.ucc.ie/gnuplot/gnuplot.html, 1998.

[118] A. Vande Wouver, Ph. Suacez, and W. E. Schiesser, editors. *Adaptive Method of Lines*. Chapman & Hall/CRC, 2001.

[119] S. J. Wright. *Primal-Dual Interior-Point Methods*. siam, 1997.

Abbildungsverzeichnis

Tabellenverzeichnis