

**Objekt-orientierte Modellierung und Simulation dynamischer
Elektrodenprozesse in der molekularen Elektrochemie.
Eine Anwendung der adaptiven
Multilevel-Finite-Elemente-Methode**

DISSERTATION

der Fakultät für Chemie und Pharmazie
der Eberhard-Karls-Universität Tübingen

zur Erlangung des Grades eines Doktors
der Naturwissenschaften

2006

vorgelegt von

Kai Ludwig

Die vorliegende Arbeit wurde von Dezember 2002 bis Juli 2006 am Institut für Organische Chemie der Universität Tübingen unter der Leitung von Prof. Dr. B. Speiser angefertigt.

Danksagung

Ich möchte besonders Herrn Prof. Dr. B. Speiser danken für die Vergabe des interdisziplinären Themas dieser Doktorarbeit, für die vielen zielführenden Fachgespräche und für das Vertrauen, welches er mir auch in Bezug auf eine zuletzt freiere Mitarbeit in seinem Arbeitskreis entgegengebracht hat. Sein wissenschaftlicher Stil ist mir ein Vorbild.

Ein herzlicher Dank gilt Dr. Martin Bogdan und Dr. Elena P. Sapozhnikova vom Wilhelm-Schickard-Institut für Informatik, Tübingen sowie Dr. Isabel Morales Moreno aus der Grupo de Investigación de Electroquímica Teórica y Aplicada de la Universidad de Murcia für die bereichernde Zusammenarbeit im Rahmen des EChem++-Projekts.

Ein besonderer Dank gilt auch den Mitgliedern des Arbeitskreises Speiser: Patric Beuter, Dr. Wolfgang Märkle, Dr. Filip Novak, Nicolas Plumeré, Elena Popa, Britta Rochier, Bernhard Sandig, Carsten Tittel für die vielen nützlichen Gespräche und für die offene und freundliche Arbeitsatmosphäre. Viel Glück Euch allen!!

Ich möchte mich besonders bei meiner Lebenspartnerin Jasmina Orman und meinen Eltern Isolde und Manfred Ludwig bedanken, die mir stets zur Seite stehen.

Ich danke dem Land Baden-Württemberg und der Universität Tübingen sowie der Deutschen Forschungsgemeinschaft (DFG) für die finanzielle Unterstützung im Rahmen eines Promotionsstipendiums nach dem Landesgraduiertenförderungsgesetz (LGFG) und der Forschergruppe *“Analysis and modeling of diffusion/dispersion-limited reactions in porous media”*.

Teile dieser Arbeit wurden bereits publiziert:

EChem++ - An Object-Oriented Problem Solving Environment for Electrochemistry. 4. Adaptive Multilevel Finite Elements Applied to Electrochemical Models. Algorithm and Benchmark Calculations

K. Ludwig, B. Speiser, *J. Electroanal. Chem.* **2006**, 588, 74 – 87

EChem++ - An Object-Oriented Problem Solving Environment for Electrochemistry. 2. The Kinetic Facilities of Ecco - A Compiler for (Electro-)Chemistry

K. Ludwig, B. Speiser, *J. Chem. Inf. Comput. Sci.* **2004**, 44(6), 2051 – 2060

EChem++ - An Object-Oriented Problem Solving Environment for Electrochemistry. 1. A C++ Class Collection for Electrochemical Excitation Functions

K. Ludwig, L. Rajendran, B. Speiser, *J. Electroanal. Chem.* **2004**, 568, 203 – 214

Tag der mündlichen Prüfung:

23. Oktober 2006

Dekan:

Prof. Dr. L. Wesemann

1. Berichterstatter:

Prof. Dr. B. Speiser

2. Berichterstatter:

Prof. Dr. Ch. Ochsenfeld

3. Berichterstatter:

Prof. Dr. D. Britz

Inhaltsverzeichnis

1. Einleitung	1
2. Zielsetzung	5
1. Modellierung	7
3. Modellanalyse	9
3.1. Modellgeometrie	10
3.2. Anregungsfunktionen	11
3.3. Reaktions-Diffusions-Gleichungen	12
3.4. Die Stöchiometriematrix	14
3.5. Homogene Reaktionen	17
3.6. Heterogene Elektronentransfers	18
3.7. Adsorptionen und Oberflächenelektronentransfers	24
3.8. Oberflächenreaktionen	28
3.9. Dimensioniertes Modell	29
3.10. Dimensionsloses Modell	30
3.11. Strom, Potential und Massenfluss	35
3.12. Objekt-orientierte Modellierung	37
4. Ecco	41
4.1. Eingabesprache	43
4.2. Reaktionsterme und Randgleichungen	49
4.3. Objekt-orientiertes Design und Algorithmus	52
4.3.1. Grammatik	53
4.3.2. Der Kinetik-Compiler	54
4.3.3. Das <i>ReactionNetwork</i> -Objekt	59
4.3.4. Die <i>Law</i> -Objekte	61
4.3.5. Das <i>CurrentPotentialResidual</i> -Objekt	64
4.3.6. Das <i>CurrentResponse</i> -Objekt	64
4.3.7. Konsistenzprüfung	67
4.3.8. Wiederverwendung und Erweiterbarkeit	72

5. Die Adapter-Bibliothek	75
II. Simulation	77
6. Ein voll-adaptiver Algorithmus für Elektrodenprozesse	79
6.1. Die partielle differentiell-algebraische Formulierung	82
6.2. Rosenbrock-Methoden	84
6.2.1. Der ε -eingebettete Ansatz	85
6.2.2. Adaptive Schrittweitensteuerung	87
6.3. Adaptive finite Elemente	88
6.3.1. Variationelle Formulierung und FE-Interpolation	89
6.3.2. Der hierarchische Fehlerschätzer	94
6.4. Fehlerskalierung	96
6.5. Gitterrevolution	99
6.6. Objekt-orientiertes Design und Implementierung	100
6.6.1. Das Strategie-Muster	101
6.6.2. Kopplung an Ecco	104
7. Anwendungen	107
7.1. Potentialkontrollierte Experimente	108
7.1.1. <i>Beispiel 1:</i> Chronoamperometrie — Einelektronentransfer	108
7.1.2. <i>Beispiel 2:</i> Doppel- und Multi-Puls-Chronoamperometrie — Ein- elektronentransfer	116
7.1.3. <i>Beispiel 3:</i> Chronoamperometrie — Das Reinert-Berg-System	120
7.1.4. <i>Beispiel 4:</i> Cyclische Voltammetrie — Einelektronentransfer	121
7.1.5. <i>Beispiel 5:</i> Linear Sweep Voltammetry — Katalytische Folgereak- tion pseudo-erster Ordnung	125
7.1.6. <i>Beispiel 6:</i> Cyclische Voltammetrie — Zweielektronentransfer mit Disproportionierungsreaktion	128
7.1.7. <i>Beispiel 7:</i> Cyclische Voltammetrie — Elektroaktive Adsorptions- schicht	130
7.1.8. <i>Beispiel 8:</i> Cyclische Voltammetrie — Adsorptionssystem ohne und mit Folgereaktionen auf der Elektrodenoberfläche	132
7.1.9. <i>Beispiel 9:</i> Cyclische Voltammetrie in Dünnschichtsystemen mit einer einzelnen Arbeitselektrode	137
7.1.10. <i>Beispiel 10:</i> Chronoamperometrie in einer Dünnschichtzelle mit zwei Arbeitselektroden	142
7.2. Stromkontrollierte Experimente	144
7.2.1. <i>Beispiel 11:</i> Chronopotentiometrie mit konstanter und komplexen Anregungsfunktionen — Einelektronentransfer	146

7.2.2. <i>Beispiel 12</i> : Chronopotentiometrie mit konstanter Anregungsfunktion — Eielektronentransfer mit Folgereaktion erster Ordnung . . .	152
8. Zusammenfassung	157
8.1. Teil I — Modellierung	157
8.2. Teil II — Simulation	159
A. Anhang	163
A.1. Erläuterung zu Gleichung (3.19)	163
A.2. Ein generisches Beobachter-Muster	164
A.3. Die <i>definition</i> -Klasse	167
A.4. Technische Anmerkungen	173
Literaturverzeichnis	175
Abbildungsverzeichnis	187
Tabellenverzeichnis	189
Abkürzungsverzeichnis	191

1. Einleitung

In der Grundlagenforschung sowie in industrieller Forschung und Entwicklung ist eine zuverlässige Simulation physikalischer und chemischer Prozesse von entscheidender Bedeutung. Simulationen dienen der Verifikation experimenteller Daten, zu Fehleranalysen, Optimierungen sowie in Machbarkeitsstudien zur kostengünstigen Entwicklung neuer Technologien. Auch in der elektrochemischen Grundlagenforschung besitzt die Modellierung und Simulation von Elektrodenprozessen eine lange Tradition[1–4].

Elektrochemische Prozesse treten in verschiedenen Anwendungsgebieten auf, wie z.B. in der Entwicklung von Brennstoffzellen, Batterien und elektrochemischen (Bio-)Sensoren[5–13]. Auch bei der Entwicklung neuer Katalysemethoden spielt das elektrochemische Verhalten von Molekülen eine wichtige Rolle[14, 15]. Eine grundlegende experimentelle Untersuchung insbesondere in Verbindung mit der Simulation dynamischer Elektrodenvorgänge führt daher oft zu anwendungsrelevanten Ergebnissen.

In der molekularen Elektrochemie werden Elektronentransferprozesse zwischen Elektroden und gelösten oder an Elektrodenoberflächen adsorbierten chemischen Substanzen initiiert und analysiert. Die elektrochemisch generierten Produkte sind oft reaktive Teilchen, die in unterschiedlichen Reaktionsmechanismen weiterreagieren können. Der Transportprozess (Diffusion, Konvektion, Migration) der chemischen Substanzen zwischen Lösung und Elektrodenoberfläche wird durch die elektrochemische Analysemethode sowie die Zellgeometrie bestimmt[16].

Klassische elektrochemische Analyseverfahren, wie z.B. die cyclische Voltammetrie, liefern mechanistische, thermodynamische sowie insbesondere kinetische Aussagen über Elektrodenprozesse. In Kombination mit spektroskopischen Methoden sind strukturelle Informationen über die elektrochemisch generierten Teilchen zugänglich (Spektroelektrochemie)[17]. Zur Untersuchung von Grenzschichten sowie technologisch interessanten elektroaktiven Oberflächen, wie Katalysatormaterialien oder Biosensoren, stellt die SECM¹-Technologie heute ein modernes Werkzeug dar[18, 19].

Die Komplexität der elektrochemischen Modelle führte bisher zu der Praxis, spezielle Computerprogramme zur Simulation einzelner Probleme zu entwickeln. Ein Beispiel ist das Programm SIMSENSOR[20–22], welches ausschließlich die Simulation enzymkatalytischer Elektrodenreaktionen unterstützt. Einen allgemeineren Ansatz beschreiben Softwarepakete wie EASI[23], DigiSim[24], DigiElch[25] und ELSIM[26]. Die Erweiterung dieser Programme wird zum einen durch den Gebrauch unflexibler, zum Teil veralteter Programmiermethoden erschwert. Zum anderen wird der Quellcode der Software nur

¹Scanning Electrochemical Microscopy

selten der wissenschaftlichen Öffentlichkeit zur Verfügung gestellt, was eine Erweiterung des Programms sowie eine Prüfung der zugrunde liegenden Algorithmen verhindert. Beispiele proprietärer Softwarepakete sind DigiSim[24], DigiElch[25] und ELSIM[26].

Komplizierte Zellgeometrien, dynamische Adsorptions- und Oberflächenprozesse sowie elektrochemische Experimente, die über die “Standardmethoden” hinausgehen, können mit diesen Programmen nicht oder nur auf umständliche Weise simuliert werden. Selbst die Kinetik chemischer Reaktionen in Lösung wird meist nur sehr vereinfacht, in manchen Fällen sogar fehlerhaft behandelt. Eine Erweiterung oder Anpassung der Programme durch den Anwender ist jedoch insbesondere aufgrund des proprietären Lizenzcharakters nicht möglich.

Im Gegensatz dazu bieten sich heute für die Entwicklung “dynamischer” Softwarelösungen im Wesentlichen zwei Konzepte an: (1) Ein objekt-orientiertes Design (OOD) und (2) die Öffnung des Quellcodes (*Open-Source-Software*, OSS).

In Verbindung gebracht erleichtern beide Konzepte die Entwicklung wiederverwendbarer Softwarebausteine[27], garantieren die Erweiterbarkeit und erlauben einen angemessenen Umgang mit wissenschaftlichen Resultaten (z.B. Verifizierbarkeit und Wiederverwendung)[28].

In Anlehnung an einen Vorschlag von Bieniasz[29], wurde mit EChem++[30] die OSS-Entwicklung einer sogenannten “Problemlösungsumgebung” (*problem solving environment*, PSE) für die Elektrochemie begonnen. Das langfristige Ziel der PSE ist es, die Steuerung elektrochemischer Experimente, die Analyse experimenteller Daten sowie die Modellierung und Simulation der Experimente in einem erweiterbaren Anwenderprogramm als OSS zur Verfügung zu stellen. Während das Gesamtziel sicherlich nur in einzelnen Etappen erreicht werden kann, wurde mit EChem++ bereits heute eine Infrastruktur geschaffen, die es erlaubt, einzelne Teilkomponenten der Software zu veröffentlichen[31].

Dem Chemiker wird somit die Möglichkeit gegeben sowohl als *Anwender* der Softwarekomponenten, z.B. in experimentellen Studien, von der Entwicklung zu profitieren, als auch als *aktiver Teilnehmer* zur Erweiterung und Qualitätsverbesserung der Software beizutragen.

In der vorliegenden Arbeit stehen Modellierungs- und Simulationen von EChem++ im Mittelpunkt. Sie besteht aus zwei Teilen.

Eine allgemeine Vorgehensweise zur computerunterstützten *Modellierung*² elektrochemischer Reaktionsmechanismen wird in Teil I vorgestellt. Dort werden die grundlegenden Konzepte der in dieser Arbeit entwickelten C++-Bibliothek, Ecco, beschrieben. Ecco enthält einen sogenannten *Kinetik-Compiler*, welcher einen nahezu beliebigen elektrochemischen Reaktionsmechanismus in kinetische Gleichungen übersetzt. Die generierten Terme können zur Simulation von reaktiven Transportvorgängen in elektrochemischen Zellen mit einer oder mehreren Arbeitselektroden verwendet werden. Die Modelle bein-

²Als *Modellierung* soll im Folgenden die Formulierung eines physikalisch-chemischen Modells mittels mathematischer Gleichungen verstanden werden.

halten heterogene Elektronentransfers zwischen Elektroden und gelösten sowie adsorbierten Teilchen, homogene Reaktionen in Lösung, Adsorptionsprozesse sowie Oberflächenreaktionen.

Teil I beginnt mit einer sogenannten *Modellanalyse* (Kapitel 3), in der die einzelnen Grundbausteine eines elektrochemischen, kinetischen Modells identifiziert, voneinander getrennt betrachtet und schließlich zu dem Gesamtmodell zusammengefügt werden (auch bekannt als *objekt-orientierte Analyse*, OOA[32]). Auf Basis der OOA wird anschließend in Kapitel 4 die Ecco-Bibliothek im Detail vorgestellt. Eine weitere Bibliothek, welche die Umrechnung zwischen dimensionierten und dimensionlosen Modellgleichungen leistet wird kurz in Kapitel 5 beschrieben.

Teil II beschäftigt sich dann mit der numerischen *Simulation*³ von eindimensionalen Reaktions-Diffusions-Modellen durch voll-adaptive Algorithmen. Die Algorithmen werden darin an die Reaktions- und Randterme des Kinetik-Compilers gekoppelt, sodass sämtliche Elektrodenprozesse aus Teil I simuliert werden können.

Teil II beinhaltet zudem einen neuen, voll-adaptiven Ansatz zur Simulation elektrochemischer Modelle. Auf der Basis adaptiver Zeitintegratoren, in Kombination mit einer modernen, adaptiven Finite-Elemente-Methode wird ein Konzept vorgestellt, welches eine Fehlerkontrolle in den Konzentrationen gelöster und adsorbierter Teilchen *sowie* in der jeweiligen Observablen eines elektrochemischen Experiments (Strom oder Elektrodenpotential) durchführt.

Die Eigenschaften des Verfahrens werden anhand der Simulation diverser elektrochemischer Experimente analysiert. In insgesamt 12 Beispielen werden Systeme betrachtet, welche zum einen die vielfältige Einsatzmöglichkeit von Ecco, zum anderen die Zuverlässigkeit des Algorithmus zeigen.

Neben klassischen Testsystemen werden chronopotentiometrische Experimente mit verschiedenen Anregungsfunktionen, Dünnschichtgeometrien sowie kompliziertere Adsorptionssysteme mit Folgereaktionen auf der Elektrodenoberfläche simuliert.

Eine ausführliche Übersicht über den aktuellen Stand der Forschung in beiden Teilen wird jeweils in den Einleitungen zu Kapitel 4 und Kapitel 6 gegeben.

³Als *Simulation* verstehen wir die numerische Lösung der mathematischen Modellgleichungen.

2. Zielsetzung

Das Ziel dieser Arbeit ist die Entwicklung eines möglichst allgemeinen Konzepts zur computerunterstützten Modellierung und Simulation elektrochemischer Elektrodenvorgänge. Unter Verwendung der Programmiersprache C++ sollen Softwarekomponenten geschaffen werden, die es ermöglichen, aus einem gegebenen elektrochemischen Reaktionsmechanismus die zugrunde liegenden mathematischen Modellgleichungen zu erzeugen (Kinetik-Compiler, Abbildung 2.1). Die Mechanismen sollen in einer dem Chemiker intuitiven Formulierung heterogene Elektronentransfers, homogene Reaktionen, Adsorptionsprozesse und Oberflächenreaktionen beinhalten. Neben der zu integrierenden, in der Literatur bekannten Reaktionskinetik soll es dem Chemiker ermöglicht werden, beliebig komplexe kinetische Zusammenhänge einzubringen.

Die Komponenten sollen ein objekt-orientiertes Design vorweisen, welches die Wiederverwendung und zukünftige Erweiterungen erleichtert. Zudem sollen sie als Bestandteil der OSS EChem++ der Öffentlichkeit zugänglich gemacht werden.

In einem zweiten Teil dieser Arbeit soll ein allgemeines Verfahren zur numerischen Simulation elektrochemischer Modelle gefunden, objekt-orientiert strukturiert und in der Programmiersprache C++ implementiert werden. Das Verfahren soll an die mathematischen Gleichungen des Compilers gekoppelt werden, sodass die obige Auswahl an Elektrodenprozessen simuliert werden kann.

Im Laufe der Arbeit ergaben sich weitere Ziele. In der Tradition moderner, adaptiver Finite-Elemente-Methoden sollte ein Konzept gefunden werden, welches eine Fehlerkontrolle nicht nur in den Konzentrationen der Substanzen, sondern auch in der jeweiligen *Observablen* (Strom oder Elektrodenpotential) des zu simulierenden Experiments

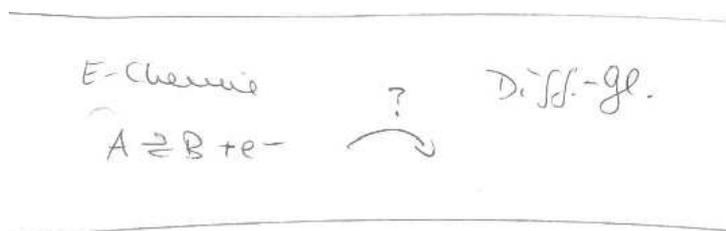


Abbildung 2.1: Skizze von Prof. Dr. B. Speiser in einer Vorbesprechung zur Zielsetzung dieser Doktorarbeit.

2. Zielsetzung

durchführt. Des Weiteren sollte die allgemeine Anwendbarkeit und Zuverlässigkeit des Verfahrens anhand anspruchsvoller, elektrochemischer Testsysteme überprüft werden. Die Softwarekomponenten sollen schließlich innerhalb eines anwenderfreundlichen Modellierungs- und Simulationsprogramm der wissenschaftlichen Öffentlichkeit als OSS zur Verfügung gestellt werden.

Teil I.

Modellierung

3. Modellanalyse

Die Modellierung von Experimenten in der molekularen Elektrochemie beinhaltet traditionell physikalisch-chemische Prozesse, die mittels Elektroden initiiert werden. Sie beschreibt elektrochemische Zellen, in denen chemische Stoffe durch Transportprozesse in die Nähe einer Elektrode gelangen und dort, in Abhängigkeit vom Elektrodenpotential, Elektronentransferreaktionen eingehen. Neben der Konzentrationsverteilung der chemischen Substanzen ist insbesondere der resultierende Stromfluss durch die Arbeitselektrode oder die Veränderung des Elektrodenpotentials von Interesse.

Die Modellierung elektrochemischer Zellgeometrien umfasst heute einzelne Arbeitselektroden, Dünnschichtzellen sowie komplexe Mehrelektrodenanordnungen.

Die Art des elektrochemischen Experiments hängt unter anderem von der physikalischen Größe ab, welche an den Elektroden der Zelle kontrolliert wird. Man unterscheidet zwischen *potentialkontrollierten* und *stromkontrollierten* Experimenten. Im potentialkontrollierten Fall wird das Elektrodenpotential von außen vorgegeben und der resultierende Stromfluss durch die Arbeitselektrode gemessen. Im stromkontrollierten Fall hingegen bewirkt ein definierter Stromfluss die Veränderung des Elektrodenpotentials, welches hier die Messgröße darstellt. In beiden Fällen beschreibt eine *Anregungsfunktion* die zeitliche Änderung der kontrollierten Größe.

Die gemessenen Signale hängen von der Konzentrationsverteilung der Spezies an der Elektrodenoberfläche ab, welche wiederum durch die an der Elektrode ablaufenden Vorgänge beeinflusst werden. Abbildung 3.1 skizziert einige solcher Elektrodenprozesse.

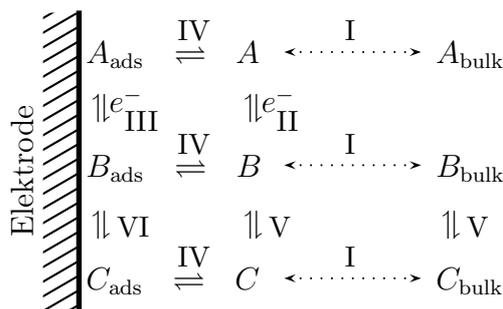


Abbildung 3.1: Übersicht über elektrochemische Prozesse an einer Elektrodenoberfläche (Erklärung siehe Text).

Der Stofftransport (I) der Spezies zwischen Lösung und Elektrodenoberfläche findet je nach experimentellen Bedingungen durch Diffusions-, Konvektions-, und Migrationsvorgänge statt. Die Prozesse II und III beschreiben heterogene Elektronentransfers zwischen der Elektrode und gelösten oder an der Elektrodenoberfläche gebundenen chemischen Substanzen. Letztere können sich durch Adsorption an der Oberfläche anlagern (IV). Die elektrochemisch generierten Substanzen stellen meist reaktive Teilchen dar, die in einer Vielzahl an unterschiedlichen Reaktionsmechanismen in der Lösung und der Diffusionsschicht (V) sowie auf der Elektrodenoberfläche (VI) weiterreagieren können.

In diesem Kapitel werden die einzelnen Modellbestandteile näher beleuchtet und sukzessive ein allgemeines Rahmenwerk zur Modellierung elektrochemischer Elektrodenvorgänge aufgebaut. In der Analyse werden die *Grundbausteine* der Modelle identifiziert, die sich später, in Kapitel 4, im objekt-orientierten Design der Ecco-Bibliothek wiederfinden. Die folgenden Abschnitte können somit als *objekt-orientierte Analyse* eines Softwareentwurfs betrachtet werden[32].

Beginnend mit der Modellgeometrie und den elektrochemischen Anregungsfunktionen (Abschnitte 3.1 und 3.2) werden Reaktions- und Transportprozesse behandelt (Abschnitte 3.3 – 3.5). Es folgt die Einbindung von Elektronentransfers, Adsorptionen und Oberflächenreaktionen (Abschnitte 3.6 – 3.8). Insgesamt erhält man zunächst dimensionierte Modellgleichungssysteme (Abschnitt 3.9), die anschließend in ein dimensionloses Modell transformiert werden (Abschnitt 3.10). Strategien zur Berechnung der Observablen aus den dimensionslosen Gleichungen werden in Abschnitt 3.11 näher behandelt. Schließlich werden in Abschnitt 3.12 die Elektrodenprozesse mit Hilfe einer objekt-orientierten Sprache beschrieben.

3.1. Modellgeometrie

In der Vergangenheit wurden zumeist eindimensionale Modellgeometrien zur Simulation elektroanalytischer Zellanordnungen eingesetzt[2, 26, 33, 34]. Komplexere Systeme schließen heute zwei- und dreidimensionale Modelle zur Simulation von Mikroelektroden[35–39], SECM-Elektrodentips[40, 41], Kanalzellen[42–48] und *Wall Tube*-Elektroden[49] ein.

Mathematisch lässt sich die Zellgeometrie allgemein durch ein Gebiet $\Omega = \bigcup \Omega_k \subset \mathbb{R}^d$, $d = 1, 2, 3$ beschreiben, welches von einem Rand Γ umgeben ist. Der Rand kann in nummerierte Randsegmente Γ_n ($n = 1, 2, \dots$) unterteilt werden (Abbildung 3.2(a)), um z.B. zwischen Elektrodenoberflächen, inerten Wänden oder Phasengrenzen (siehe[50]) in der Zelle zu unterscheiden.

Im Falle linearer Diffusion und planarer Elektroden lässt sich die Zellgeometrie durch ein eindimensionales (1D-) Modell darstellen (Abbildung 3.2(b)), in dem die x -Achse den Diffusionsraum senkrecht zur Elektrodenoberfläche beschreibt[51]. Die Ränder in Abbildung 3.2(b) stehen für Elektrodenoberflächen, für das Ende der Diffusionsschicht zur

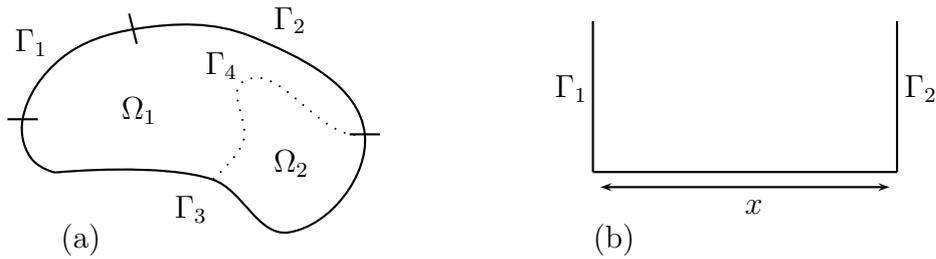


Abbildung 3.2: Modellgeometrie: allgemeines Modell, $\Omega = \bigcup \Omega_k \subset \mathbb{R}^d$, $d = 1, 2, 3$ (a) mit Rändern Γ_n (n Randindex) und 1D-Modell (b).

Lösung hin oder für eine inerte Wand. Somit lassen sich mit dem 1D-Modell klassische elektroanalytische Methoden an einer einzelnen Arbeitselektrode mit halb-begrenztem (*semi-infinitem*) oder begrenztem (*finitem*) Diffusionsraum, oder Dünnschichtzellen mit zwei sich parallel gegenüberstehenden Arbeitselektroden modellieren.

Wir werden uns in dieser Arbeit auf die Modelle der in Abbildung 3.2(b) dargestellten 1D-Geometrie konzentrieren. Es sei jedoch betont, dass die meisten der folgenden Modellgleichungen ohne oder nur mit geringem Aufwand auf nicht-lineare Diffusionsprozesse (siehe z.B. Tabelle 4.4.2 in[51]) oder höherdimensionale Zellgeometrien übertragen werden können.

3.2. Anregungsfunktionen

Die Unterscheidung zwischen potential- und stromkontrollierten Experimenten, sowie zwischen den speziellen Typen der elektroanalytischen Methoden liefert die Art der Anregungsfunktionen an den Elektrodenoberflächen der Zellgeometrie. Die objekt-orientierte Modellierung solcher elektrochemischer Anregungsfunktionen wurde in[52] ausführlich beschrieben. Der Quellcode steht der Öffentlichkeit als *Excitation Function Template Library* zur Verfügung[30]. In diesem Abschnitt wollen wir lediglich die Grundkonzepte erläutern.

Im potentialkontrollierten Fall geben die Anregungsfunktionen die Veränderung des Potentials

$$E = f_E(t) \quad (3.1)$$

an einer Arbeitselektrode vor, während der Strom, der dabei durch die Elektrode fließt, gemessen wird. Umgekehrt wird im stromkontrollierten Fall der elektrische Strom, der durch die Arbeitselektrode fließt, gesteuert

$$i = f_i(t) \quad (3.2)$$

und das Elektrodenpotential gemessen. Die rechten Seiten $f(t)$ in den Gleichungen (3.1) und (3.2) beschreiben stückweise stetige Funktionen, die sich auf natürliche Weise aus

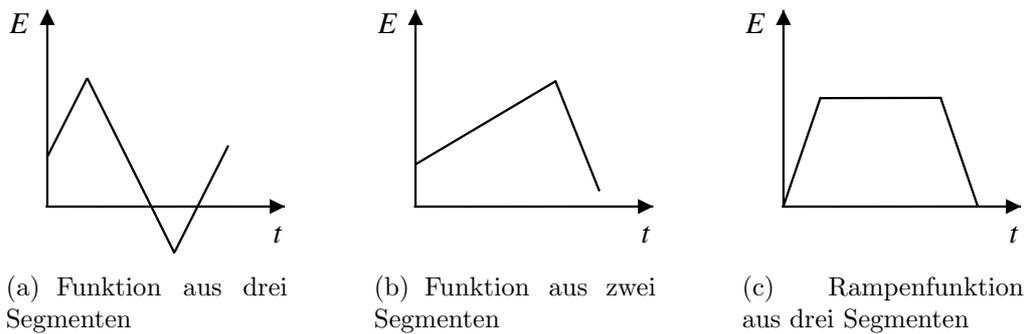


Abbildung 3.3: Beispiele für Potential-Zeit-Anregungsfunktionen.

Segmenten zusammensetzen. Die Segmente werden auf bestimmten Zeitintervallen definiert.

Abbildung 3.3 zeigt einige Beispiele, die aus linearen und konstanten Potential-Zeit-Segmenten aufgebaut sind. Die ersten beiden Abbildungen stehen für Anregungsfunktionen der cyclischen Voltammetrie (CV) mit gleichmäßiger (a) und variierender Spannungsvorschubgeschwindigkeit (b) (siehe z.B.[53]). Ein Rampenexperiment wird durch Abbildung 3.3(c) beschrieben[54]. Weitere funktionale Beziehungen, die in der Elektrochemie verwendet werden, sind in Tabelle 3.1 zusammengestellt. Als ein einfaches Beispiel sei die Chronoamperometrie (CA) erwähnt, deren Anregungsfunktion aus konstanten Segmenten besteht. Etwas komplexere Segmente werden in der *Cyclic Reciprocal Derivative Power Chronopotentiometry* (CRDPC)[55], in Impedanz-Experimenten[51] und in der *Cyclic Reciprocal Derivative Exponential Chronopotentiometry* (CRDEC)[56] benötigt.

Potential-Zeit- bzw. Strom-Zeit-Anregungsfunktionen dienen während der Simulation (siehe Teil II) zur Berechnung der potentialabhängigen Randterme des Modells.

3.3. Reaktions-Diffusions-Gleichungen

Reaktions-Diffusions-Gleichungen beschreiben Prozesse in sehr unterschiedlichen Bereichen der Naturwissenschaften. Typische Anwendungen sind Wärmetransportvorgänge, z.B. in der Modellierung der regionalen Hyperthermie zur Krebsbekämpfung[60], propagierende Flammenfronten[60], invasive Ausbreitungen von Ameisen oder Krankheitskeimen[61] oder die Dynamik von Plankton- und Fischbeständen[62]. Es sei betont, dass zahlreiche weitere naturwissenschaftliche Phänomene durch Reaktions-Diffusions-Gleichungen modelliert werden können.

In der molekularen Elektrochemie sind insbesondere reaktive Transportvorgänge von Molekülen in der Lösung vor einer Elektrode von zentraler Bedeutung (Abbildung 3.1, Prozesse I und V). Elektrochemische Reaktions- und Transportmodelle bilden ein System

Tabelle 3.1: Funktionale Zusammenhänge in den Segmenten elektrochemischer Anregungsfunktionen.

Beziehung	Ausdruck	Variablen ^{a)}	Methoden	
konstant	$d = d_{\text{const}}$	d_{const}	CA	[57]
linear	$d = d_{\text{start}} + (d_{\text{end}} - d_{\text{start}})/i_{\text{max}} * i$	$d_{\text{start}}, d_{\text{end}}, i_{\text{max}}$	CV	[58]
polynomiell	$d = d_0 + d_1 * i^u$	$d_0, d_1, u > -1.0$	CRDPC ^{b)}	[55]
sinusoidal	$d = d_0 + d_{\text{max}} \sin(f * i + \phi)$	$d_0, d_{\text{max}}, f, \phi$	Impedanz	[51](S.368ff)
exponentiell	$d = d_0 + d_1 * e^{\omega i}$	d_0, d_1, ω	CRDEC ^{c)}	[56]
polynomiell-exponentiell	$d = d_0 + d_1 * i^u e^{\omega i}$	$d_0, d_1, \omega, u \geq -1/2$		[59]

^{a)}Variablen, die zur Definition benötigt werden; i steht für die unabhängige, d für die abhängige Variable.

^{b)}*Cyclic Reciprocal Derivative Power Chronopotentiometry*; in[55]: $d_0 = 0$

^{c)}*Cyclic Reciprocal Derivative Exponential Chronopotentiometry*; in[56]: $d_0 = 0$

aus partiellen Differentialgleichungen (PDEs)[51]

$$\begin{aligned}
 \begin{pmatrix} \partial_t c_0 \\ \partial_t c_1 \\ \vdots \\ \partial_t c_{N_s-1} \end{pmatrix} &= \begin{pmatrix} D_0 \Delta c_0 \\ D_1 \Delta c_1 \\ \vdots \\ D_{N_s-1} \Delta c_{N_s-1} \end{pmatrix} + \frac{F}{RT} \begin{pmatrix} z_0 D_0 \nabla^T (c_0 \nabla \phi) \\ z_1 D_1 \nabla^T (c_1 \nabla \phi) \\ \vdots \\ z_{N_s-1} D_{N_s-1} \nabla^T (c_{N_s-1} \nabla \phi) \end{pmatrix} \\
 &\quad - \begin{pmatrix} \nabla^T c_0 \cdot \nu \\ \nabla^T c_1 \cdot \nu \\ \vdots \\ \nabla^T c_{N_s-1} \cdot \nu \end{pmatrix} + \begin{pmatrix} f_0(c_0, c_1, \dots, c_{N_s-1}) \\ f_1(c_0, c_1, \dots, c_{N_s-1}) \\ \vdots \\ f_{N_s-1}(c_0, c_1, \dots, c_{N_s-1}) \end{pmatrix} \quad (3.3)
 \end{aligned}$$

vom parabolischen Typ[63](S.78), wobei die räumlichen Variablen sowie die Form der Nabla- (∇) und Laplace-Operatoren ($\Delta = \nabla^T \nabla$) von der gewählten Geometrie abhängen können[51](S.150). Für kartesische Koordinaten $x = (x_1, x_2, x_3)$ gilt $\nabla^T = (\partial_{x_1}, \partial_{x_2}, \partial_{x_3})$. Die rechte Seite des PDE-Systems setzt sich sukzessiv aus Diffusions-, Migrations-, Konvektions- und Reaktionstermen zusammen. Der Diffusionskoeffizient und die Ionenladung einer Spezies mit Index k werden mit D_k bzw. z_k beschrieben. Die Symbole F , R und T stehen für die Faraday-Konstante, Gaskonstante und die Temperatur, ϕ für das elektrische Potential und $\nu = (\nu_{x_1}, \nu_{x_2}, \nu_{x_3})^T$ für den Geschwindigkeitsvektor der Strömung. Das Strömungsfeld hydrodynamischer Systeme liefern die Kontinuitäts- und Navier-Stokes-Gleichungen. Die Potentialverteilung $\phi(t, x)$ in der Lösung wird durch die Poissongleichung beschrieben[51]. Die Lösung des PDE-Systems (3.3) beschreibt den

Konzentrationsverlauf der einzelnen Substanzen in Abhängigkeit von der Zeit und den Ortskoordinaten, $c_k = c_k(t, x)$. Sie ist im Allgemeinen nur durch eine numerische Behandlung der Gleichungen zugänglich.

Oft werden im Falle elektroanalytischer Methoden die Reaktionsbedingungen so gewählt, dass der Transport der chemischen Substanzen vom Diffusionsterm dominiert wird (hohe Leitsalzkonzentration, ruhende Lösung). In diesem Fall können die Migrations- und Konvektionsterme vernachlässigt werden. Man erhält ein System aus Reaktions-Diffusions-Gleichungen

$$\begin{pmatrix} \partial_t c_0 \\ \partial_t c_1 \\ \vdots \\ \partial_t c_{N_s-1} \end{pmatrix} = \begin{pmatrix} D_0 \Delta c_0 \\ D_1 \Delta c_1 \\ \vdots \\ D_{N_s-1} \Delta c_{N_s-1} \end{pmatrix} + \begin{pmatrix} f_0(c_0, c_1, \dots, c_{N_s-1}) \\ f_1(c_0, c_1, \dots, c_{N_s-1}) \\ \vdots \\ f_{N_s-1}(c_0, c_1, \dots, c_{N_s-1}) \end{pmatrix} \quad (3.4)$$

deren Lösung allgemein ebenfalls nur numerisch berechnet werden kann. Die Diffusionsterme $D_k \Delta c_k$ setzen sich aus dem Diffusionskoeffizienten D_k und dem Laplace-Operator ($\Delta = \sum_{i=1}^d \partial_{x_i x_i}$, d Raumdimension) angewandt auf c_k zusammen ($\Delta c_k = \sum_{i=1}^d \partial_{x_i x_i} c_k$). Die Diffusionskoeffizienten werden als von den Ortskoordinaten unabhängig angenommen.¹

Im Folgenden sollen Migrations- und Konvektionsprozesse stets vernachlässigt werden. Mit $c = (c_0, c_1, \dots, c_{N_s-1})^T$ wählen wir für das System (3.4) fortan die kompaktere Vektorschreibweise

$$\partial_t c = D \partial_{xx} c + R(c) \quad (3.5)$$

mit der Diagonalmatrix der Diffusionskoeffizienten $D = \text{diag}(D_0, D_1, \dots, D_{N_s-1})$ und dem Symbol ∂_{xx} als Laplace-Operator, welcher komponentenweise auf c angewendet wird. Die Reaktionsterme werden von der Funktion

$$R(c) = (f_0(c), f_1(c), \dots, f_{N_s-1}(c))^T \quad (3.6)$$

repräsentiert. Die Einträge $f_k(c)$ setzen sich aus den Geschwindigkeitsgesetzen der einzelnen homogenen Reaktionen zusammen, an denen die Spezies beteiligt sind. Um in späteren Abschnitten den Aufbau der Terme möglichst elegant durchführen zu können, wollen wir uns zunächst auf die sogenannte *Stöchiometriematrix* konzentrieren.

3.4. Die Stöchiometriematrix

Die kompakte Beschreibung der Stöchiometrie komplexer Reaktionsmechanismen (Reaktionsnetzwerke) durch eine Stöchiometriematrix findet heute Anwendung in verschiedenen Bereichen der chemischen und elektrochemischen Reaktionskinetik. So bildet die

¹Die Lösung vor der Elektrode wird hier als isotrope, homogene Umgebung aufgefasst[33, 64]. Richtungsabhängige Diffusionsprozesse treten hingegen in anisotropen Medien auf (z.B. in Flüssigkristallen[65] oder porösen Medien[64]).

Stöchiometriematrix z.B. die Grundlage für stöchiometrische Netzwerkanalysen[66–69], in der Modellierung chemischer und elektrochemischer Reaktionskinetik[70–73] oder dient zur Aufklärung von Abhängigkeiten zwischen thermodynamischen Parametern einzelner Reaktionen[74]. Auch in der vorliegenden Arbeit stellt sie ein wichtiges Konzept zur Beschreibung und zur Analyse elektrochemischer Reaktionsmechanismen dar. Die Erzeugung der Matrix gehört zu den zentralen Aufgaben der Modellierungskomponente Ecco (Kapitel 4).

Wie schon in früheren Arbeiten (z.B.[70–73]) werden die Reaktionsmechanismen durch ein System aus N_r chemischen Reaktionen beschrieben, an denen insgesamt N_s verschiedene Reaktanden A_k beteiligt sind.

$$\sum_{k=0}^{N_s-1} e_{ik} \cdot A_k \rightleftharpoons \sum_{k=0}^{N_s-1} p_{ik} \cdot A_k \quad i = 0, 1, \dots, N_r - 1 \quad (3.7)$$

Zu den Reaktanden des Mechanismus sollen adsorbierte und gelöste Teilchen sowie Elektronen zählen. Die Reaktionsstöchiometrie ist durch die Koeffizienten $e_{ik} \geq 0$ (für Edukte) und $p_{ik} \geq 0$ (für Produkte) gegeben, welche die Matrizen E und P

$$E = \begin{pmatrix} e_{00} & e_{01} & \dots & e_{0M} \\ e_{10} & e_{11} & \dots & e_{1M} \\ \vdots & \vdots & \ddots & \vdots \\ e_{N0} & e_{N1} & \dots & e_{NM} \end{pmatrix} \quad P = \begin{pmatrix} p_{00} & p_{01} & \dots & p_{0M} \\ p_{10} & p_{11} & \dots & p_{1M} \\ \vdots & \vdots & \ddots & \vdots \\ p_{N0} & p_{N1} & \dots & p_{NM} \end{pmatrix} \quad (3.8)$$

bilden (mit $M = N_s - 1$ und $N = N_r - 1$). Die *Stöchiometriematrix* S des Mechanismus (3.7) wird durch die Differenz zwischen P und E definiert

$$S = P - E = \begin{pmatrix} \nu_{00} & \nu_{01} & \dots & \nu_{0M} \\ \nu_{10} & \nu_{11} & \dots & \nu_{1M} \\ \vdots & \vdots & \ddots & \vdots \\ \nu_{N0} & \nu_{N1} & \dots & \nu_{NM} \end{pmatrix} \quad (3.9)$$

Die Einträge $\nu_{ik} = p_{ik} - e_{ik}$ werden *Stöchiometrikoeffizienten* genannt, es gilt

$$\nu_{ik} = \begin{cases} \nu_{ik} < 0 & k \text{ Edukt} \\ \nu_{ik} > 0 & k \text{ Produkt} \end{cases} \quad (3.10)$$

Die Zeilen der Matrix bilden die *Stöchiometrievektoren* der einzelnen Reaktionen. Weiterhin soll hier speziell gelten:

Die Spalten von S enthalten zuerst die Stöchiometrikoeffizienten der adsorbierten Spezies, gefolgt von den Koeffizienten der gelösten Teilchen und der Elektronen. Adsorbierte Formen einer Spezies sowie Elektronen werden aufgrund ihres Auftretens an verschiedenen Rändern der Modellgeometrie unterschieden.

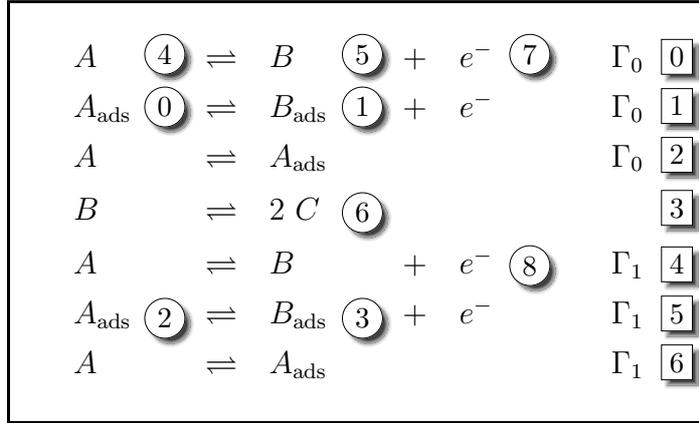


Abbildung 3.4: Nummerierung von Reaktionen und Spezies; Reaktionsindizes sind in eckigen Boxen, Speziesindizes in Kreisen dargestellt. Zuerst werden adsorbierte Teilchen gezählt. Es folgen die gelösten Spezies und schließlich die Elektronen. Bei adsorbierten Substanzen und Elektronen muss zwischen verschiedenen Oberflächen (Γ_n) unterschieden werden.

Es ergeben sich beispielsweise die in Abbildung 3.4 dargestellten Indizierungen. Das Beispiel beschreibt einen hypothetischen Reaktionsmechanismus aus heterogenen Elektronentransfers (Indizes 0, 1, 4, 5) und Adsorptionen (Indizes 2, 6) an zwei Elektrodenoberflächen (Γ_n , $n = 0, 1$; siehe Abschnitt 3.1) sowie einer homogenen Reaktion (Index 3). Die dazugehörige Stöchiometriematrix lautet

$$S = \left(\begin{array}{cccc|ccc|cc} 0 & 0 & 0 & 0 & -1 & 1 & 0 & 1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \end{array} \right) \quad (3.11)$$

Die ersten vier Spalten enthalten die Stöchiometrikoeffizienten der Formen $A_{\text{ads},0}$, $B_{\text{ads},0}$, $A_{\text{ads},1}$ und $B_{\text{ads},1}$. Die mittleren Spalten stehen für A , B und C und die letzten beiden Spalten für die Elektronen e_0^- und e_1^- . Die tiefgestellten Zahlen in den Namen ermöglichen eine Unterscheidung zwischen adsorbierten Teilchen und zwischen Elektronen, die an verschiedenen Oberflächen auftreten, beziehungsweise übertragen werden.

3.5. Homogene Reaktionen

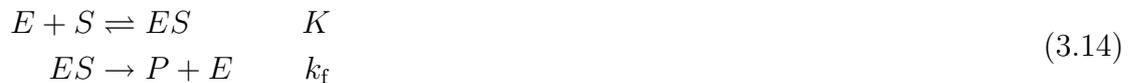
Die Form der $f_k(c)$ -Terme in Gleichung (3.6) hängt vom kinetischen Modell der homogenen Reaktionen in Lösung ab (Abbildung 3.1, Prozess V). Für die Kinetik einer homogenen Reaktion wird üblicherweise ein *Potenzgesetz* angenommen[72, 75].

$$r_{\text{P}_{\text{rev}}} = k_{\text{f}} \prod_k c_k^{\alpha_k} - k_{\text{b}} \prod_k c_k^{\beta_k} \quad (3.12)$$

Es sei bemerkt, dass die einzelnen Reaktionsordnungen der Edukte (α_k) und Produkte (β_k) in Gleichung (3.12) *nicht* den Stöchiometrikoeffizienten ν_{ik} der Substanzen entsprechen (siehe z.B.[75] S.65). Für irreversible Reaktionen ($k_{\text{b}} = 0$) gilt

$$r_{\text{P}_{\text{irr}}} = k_{\text{f}} \prod_k c_k^{\alpha_k} \quad (3.13)$$

In speziellen Fällen lassen sich experimentell kinetische Zusammenhänge beobachten, die nicht den Formen (3.12) oder (3.13) gehorchen. Ein Beispiel ist eine irreversibel verlaufende Enzymkatalyse



in der die Kinetik durch die Michaelis-Menten-Gleichung wiedergegeben wird. Für die Bildung des Produkts P ergibt sich der Ausdruck[76](S.84)

$$r = k_{\text{f}} \cdot \frac{[E]_0[S]}{1/K + [S]} \quad (3.15)$$

Gleichung (3.15) lässt sich nicht in die Form eines der Potenzgesetze (3.12) oder (3.13) überführen.

Ein Geschwindigkeitsgesetz, welches nicht als eines der Potenzgesetze darstellbar ist, soll im Folgenden als *generisches Gesetz* bezeichnet werden

$$r_{\text{G}} = f(c) \quad (3.16)$$

wobei $f(c)$ eine beliebige mathematische Funktion der Konzentrationen darstellt. Die Gleichungen (3.12), (3.13) und (3.16) stellen die Bausteine dar, aus denen für jeden Reaktanden eines Reaktionsmechanismus ein Reaktionsterm $f_k(c)$ gebildet werden kann.

Dazu verwenden wir die Stöchiometriematrix S (siehe Gleichung (3.9)).

Sei $\mathbf{r}_{\text{hom}} = (r_0(c), r_1(c), \dots)$ ein Vektor, der die Geschwindigkeitsgesetze der einzelnen Reaktionen $r_{\text{P}_{\text{rev}}}$, $r_{\text{P}_{\text{irr}}}$ oder r_{G} enthält, dann ergeben sich die Reaktionsterme (vgl. Gleichung (3.6)) durch

$$R(c) = S_{\text{hom}}^T \cdot \mathbf{r}_{\text{hom}} \quad (3.17)$$

wobei S_{hom} die Zeilen der Stöchiometriematrix enthält, die den homogenen Reaktionen entsprechen. Die Spalten der adsorbierten Teilchen und der Elektronen werden in S_{hom} nicht berücksichtigt.

Zu Beginn eines elektrochemischen Experiments befindet sich jede homogene Reaktion in einem Gleichgewichtszustand, der durch das Massenwirkungsgesetz gegeben ist

$$(K_i)^{s_i} = \frac{k_{\text{f},i}}{k_{\text{b},i}} = \prod_k c_k^{\beta_{ik} - \alpha_{ik}} \quad (3.18)$$

wobei K_i die Gleichgewichtskonstante der Reaktion beschreibt. Zwischen den Stöchiometrikoeffizienten und den einzelnen Reaktionsordnungen gilt die Beziehung[77]

$$s_i = \frac{\beta_{ik} - \alpha_{ik}}{\nu_{ik}} \quad (3.19)$$

für ein festes $s_i \in \mathbb{R}^+$ (Details siehe Anhang A.1).

Der Gleichgewichtszustand bestimmt die *Anfangswerte* c_k^0 des Modells. Neben homogenen Reaktionen können am Gleichgewichtszustand auch Adsorptionen und Oberflächenreaktionen beteiligt sein (siehe Abschnitte 3.7 und 3.8).

3.6. Heterogene Elektronentransfers

Heterogene Elektronentransfers (ETs) finden zwischen gelösten oder adsorbierten Molekülen und einer Elektrode statt (Abbildung 3.1, Prozesse II und III). Die Kinetik der Elektronentransferprozesse wird durch die *Randwerte* in das Modell eingebracht.

Die Modellgleichungen der Elektronentransfers auf der Elektrodenoberfläche (Prozess III) hängen vom gewählten Adsorptionsmodell ab[78]. Sie werden daher gemeinsam mit Adsorptionsvorgängen in Abschnitt 3.7 behandelt. Wir werden im Folgenden den Begriff *Volumenelektronentransfer (VET)* für heterogene Elektronentransfers zwischen einer Elektrode und einer gelösten Spezies verwenden. Der Transfer zwischen einem adsorbierten Teilchen und Elektrode wird als *Oberflächenelektronentransfer (OET)* bezeichnet.

Um Verwechslungen vorzubeugen sei an dieser Stelle betont, dass VETs und OETs stets an der Elektrodenoberfläche eintreten. "Homogene" Elektronentransfers, die zwischen gelösten oder zwischen adsorbierten Teilchen eintreten, werden in dieser Arbeit hingegen nicht gesondert behandelt und zu den homogenen Reaktionen (Abschnitt 3.5)

beziehungsweise zu den Oberflächenreaktionen (Abschnitt 3.8) gezählt.

Die Grundgleichungen für VETs finden sich in den Büchern von Bard und Faulkner[51] und Britz[33]. Sie sollen im Folgenden möglichst allgemein formuliert werden.

Betrachtet man einen VET an einer Elektrode Γ_e



so erfolgt dieser für gewöhnlich mit einer Gesamtreaktionsrate

$$r = r_f - r_b = k_f c_A(t, x) - k_b c_B(t, x) \quad x \in \Gamma_e \quad (3.21)$$

die sich aus den Raten der Hin- (r_f) und der Rückreaktion (r_b) zusammensetzt.² Die Partialströme der Hin- (i_f) und Rückreaktion (i_b) sind proportional zu den einzelnen Reaktionsraten, wodurch sich für den aus Reaktion (3.20) resultierenden Strom an einer Elektrode mit Oberfläche A_{Γ_e}

$$i = i_f - i_b = \nu_e F A_{\Gamma_e} [k_f c_A(t, x) - k_b c_B(t, x)] \quad x \in \Gamma_e \quad (3.22)$$

ergibt. Es sei darauf hingewiesen, dass Gleichung (3.22) durch das Vorzeichen des Stöchiometriekoeffizienten des Elektrons ($\nu_e < 0$ für eine Reduktion, $\nu_e > 0$ für eine Oxidation) bereits die IUPAC-Konvention für das Vorzeichen des Stroms erfüllt[79]. Des Weiteren stellen die Gleichungen

$$\begin{aligned} -D_A \partial_x c_A(t, x) &= -\frac{i}{\nu_e F A_{\Gamma_e}} & x \in \Gamma_e \\ -D_B \partial_x c_B(t, x) &= \frac{i}{\nu_e F A_{\Gamma_e}} & x \in \Gamma_e \end{aligned} \quad (3.23)$$

eine Beziehung zwischen dem Massenfluss auf der einen und der elektrischen Stromdichte auf der anderen Seite her. Setzt man Gleichung (3.22) in (3.23) ein, ergeben sich für die ersten Ableitungen der Volumenkonzentrationen an der Elektrodenoberfläche sogenannte Neumann-Randbedingungen

$$\begin{aligned} -D_A \partial_x c_A(t, x) &= -[k_f c_A(t, x) - k_b c_B(t, x)] & x \in \Gamma_e \\ -D_B \partial_x c_B(t, x) &= [k_f c_A(t, x) - k_b c_B(t, x)] & x \in \Gamma_e \end{aligned} \quad (3.24)$$

die zur Lösung der im vorherigen Abschnitt beschriebenen Reaktions-Diffusions-Gleichungen verwendet werden können.

Ein wichtiger Unterschied zur homogenen Reaktionskinetik besteht darin, dass die Geschwindigkeitskonstanten k_f und k_b in den obigen Gleichungen vom Potential der Elektrode abhängen, was durch das *Butler-Volmer-Modell* mit Hilfe der *Theorie des aktivierten Komplexes* beschrieben werden kann[51](Kapitel 3). Dies soll im Folgenden in

²Die Notation $x \in \Gamma_e$ beschreibt x -Werte auf dem Rand der Modellgeometrie. Für 1D-Modelle mit einer Arbeitselektrode gilt üblicherweise $x = 0$ (vgl. Γ_1 in Abbildung 3.2(b)).

3. Modellanalyse

Grundzügen wiedergegeben werden. Ein kurzer Vergleich zu der moderneren *Marcus-Theorie* wird am Ende des Abschnittes gegeben.

Für die heterogenen Geschwindigkeitskonstanten werden Arrhenius-Gleichungen

$$k_f = A_f \exp(-\Delta G_f^\ddagger/RT) \quad (3.25)$$

$$k_b = A_b \exp(-\Delta G_b^\ddagger/RT) \quad (3.26)$$

angenommen, jeweils mit der freien Aktivierungsenergie ΔG^\ddagger und präexponentiellem Faktor A . Das Butler-Volmer-Modell besagt, dass die freien Aktivierungsenergien gemäß

$$\Delta G_f^\ddagger = \Delta G_{of}^\ddagger + \alpha|\nu_e|F(E - E^0) \quad (3.27)$$

$$\Delta G_b^\ddagger = \Delta G_{ob}^\ddagger - (1 - \alpha)|\nu_e|F(E - E^0) \quad (3.28)$$

jeweils *linear* vom Potential abhängen, wobei ΔG_0^\ddagger die freie Standard-Aktivierungsenergie der Teilreaktionen im Gleichgewichtszustand mit Formalpotential E^0 beschreibt. Der *Transferkoeffizient* $0 < \alpha < 1$ gibt das Verhältnis wieder, um das die Energie des Übergangszustandes aus dem Gleichgewichtszustand durch die Veränderung des Elektrodenpotentials E abgesenkt wird. Aus den Gleichungen (3.22) und (3.25) – (3.28) folgt für einen VET

$$i = \nu_e F A_{\Gamma_e} k_h \left[e^{-\alpha|\nu_e|\frac{F}{RT}(E-E^0)} c_A(t, x) - e^{(1-\alpha)|\nu_e|\frac{F}{RT}(E-E^0)} c_B(t, x) \right] \quad (3.29)$$

mit der heterogenen (Standard-)Geschwindigkeitskonstanten $k_h/\text{m} \cdot \text{s}^{-1}$ und $x \in \Gamma_e$. Setzt man Gleichung (3.29) in Gleichung (3.23) ein, erhält man die Randbedingungen in Abhängigkeit vom Elektrodenpotential.

$$\begin{aligned} -D_A \partial_x c_A(t, x) &= -k_h \left[e^{-\alpha|\nu_e|\frac{F}{RT}(E-E^0)} c_A(t, x) - e^{(1-\alpha)|\nu_e|\frac{F}{RT}(E-E^0)} c_B(t, x) \right] \\ -D_B \partial_x c_B(t, x) &= k_h \left[e^{-\alpha|\nu_e|\frac{F}{RT}(E-E^0)} c_A(t, x) - e^{(1-\alpha)|\nu_e|\frac{F}{RT}(E-E^0)} c_B(t, x) \right] \end{aligned} \quad (3.30)$$

für $x \in \Gamma_e$. Die Gleichungen (3.29) und (3.30) beschreiben die Beziehung zwischen Strom, Potential und Massenfluss an einer Elektrodenoberfläche für eine Reduktion der Form (3.20). Es können jedoch auch Oxidationen, sowie parallele und gekoppelte Elektronentransfers auftreten. Im Falle einer Oxidation



erhält man für die Aktivierungsenergien

$$\Delta G_f^\ddagger = \Delta G_{of}^\ddagger - \alpha|\nu_e|F(E - E^0) \quad (3.32)$$

$$\Delta G_b^\ddagger = \Delta G_{ob}^\ddagger + (1 - \alpha)|\nu_e|F(E - E^0) \quad (3.33)$$

durch Vorzeichenwechsel in den Gleichungen (3.27) und (3.28). Die Beziehungen (3.21) – (3.26) bleiben hingegen unverändert. Um von der Formulierung eines Reaktionsmechanismus unabhängige Ausdrücke zu erhalten, kann der Vorzeichenwechsel durch die

Stöchiometrikoeffizienten der Komponenten eingebracht werden. So erhält man die für Reduktionen und Oxidationen gültigen Beziehungen

$$\Delta G_f^\ddagger = \Delta G_{0f}^\ddagger - \alpha \nu_e F (E - E^0) \quad (3.34)$$

$$\Delta G_b^\ddagger = \Delta G_{0b}^\ddagger + (1 - \alpha) \nu_e F (E - E^0) \quad (3.35)$$

Es folgen die allgemeinen Gleichungen der Strom-Potential-Charakteristik

$$i = \nu_e F A_{\Gamma_e} k_h \left[e^{\alpha \nu_e \frac{F}{RT} (E - E^0)} c_A(t, x) - e^{-(1-\alpha) \nu_e \frac{F}{RT} (E - E^0)} c_B(t, x) \right] \quad (3.36)$$

und der Fluss-Potential-Beziehung (siehe auch [72, 73])

$$\begin{aligned} -D_A \partial_x c_A(t, x) &= \nu_A k_h \left[e^{\alpha \nu_e \frac{F}{RT} (E - E^0)} c_A(t, x) - e^{-(1-\alpha) \nu_e \frac{F}{RT} (E - E^0)} c_B(t, x) \right] \\ -D_B \partial_x c_B(t, x) &= \nu_B k_h \left[e^{\alpha \nu_e \frac{F}{RT} (E - E^0)} c_A(t, x) - e^{-(1-\alpha) \nu_e \frac{F}{RT} (E - E^0)} c_B(t, x) \right] \end{aligned} \quad (3.37)$$

für $x \in \Gamma_e$. Der Grundbaustein der Gleichungen (3.36) und (3.37) stellt das für einen VET charakteristische *Butler-Volmer-Gesetz*

$$r_{\text{BV}} = k_h \left[e^{\alpha \nu_e \frac{F}{RT} (E - E^0)} c_{\text{Edukt}} - e^{-(1-\alpha) \nu_e \frac{F}{RT} (E - E^0)} c_{\text{Produkt}} \right] \quad (3.38)$$

dar, wobei c_{Edukt} und c_{Produkt} die Volumenkonzentrationen der Reaktionspartner an der Elektrodenoberfläche bezeichnen. Mit Hilfe dieses Bausteines können im Falle komplexer Reaktionsmechanismen allgemeine Gleichungen für die Strom- und Randwert-Berechnung abgeleitet werden.

Als Beispiel betrachten wir ein System aus gekoppelten und parallelen Elektronentransfers am linken Rand der Modellgeometrie aus Abbildung 3.2(b)

$$\begin{aligned} |\nu_{00}| \cdot A &\rightleftharpoons |\nu_{01}| \cdot B + |\nu_{06}| \cdot e^- && \Gamma_1 \\ |\nu_{11}| \cdot B &\rightleftharpoons |\nu_{12}| \cdot C + |\nu_{16}| \cdot e^- && \Gamma_1 \\ |\nu_{23}| \cdot E + |\nu_{26}| \cdot e^- &\rightleftharpoons |\nu_{24}| \cdot F && \Gamma_1 \\ |\nu_{34}| \cdot F + |\nu_{36}| \cdot e^- &\rightleftharpoons |\nu_{35}| \cdot G && \Gamma_1 \end{aligned} \quad (3.39)$$

Die Stöchiometriematrix lautet

$$S = \begin{pmatrix} \nu_{00} & \nu_{01} & 0 & 0 & 0 & 0 & \nu_{06} \\ 0 & \nu_{11} & \nu_{12} & 0 & 0 & 0 & \nu_{16} \\ 0 & 0 & 0 & \nu_{23} & \nu_{24} & 0 & \nu_{26} \\ 0 & 0 & 0 & 0 & \nu_{34} & \nu_{35} & \nu_{36} \end{pmatrix} \quad (3.40)$$

Man erkennt zwei voneinander unabhängige Blöcke, die den beiden parallelen VET-Systemen entsprechen. Sei

$$r_{\text{BV},j} = k_{h,j} \left[e^{\alpha_j \nu_{j,e} \frac{F}{RT} (E - E_j^0)} c_{\text{Edukt},j} - e^{-(1-\alpha_j) \nu_{j,e} \frac{F}{RT} (E - E_j^0)} c_{\text{Produkt},i} \right] \quad (3.41)$$

3. Modellanalyse

das Butler-Volmer-Gesetz des VETs mit Index j und $\nu_{j,e} = \nu_{j6}$. Dann setzen sich die Randbedingungen gemäß

$$\begin{aligned}
 -D_0 \partial_x c_0(t, x) &= \nu_{00} \cdot r_{\text{BV},0} \\
 -D_1 \partial_x c_1(t, x) &= \nu_{01} \cdot r_{\text{BV},0} + \nu_{11} \cdot r_{\text{BV},1} \\
 -D_2 \partial_x c_2(t, x) &= \nu_{12} \cdot r_{\text{BV},1} \\
 -D_3 \partial_x c_3(t, x) &= \nu_{23} \cdot r_{\text{BV},2} \\
 -D_4 \partial_x c_4(t, x) &= \nu_{24} \cdot r_{\text{BV},2} + \nu_{34} \cdot r_{\text{BV},3} \\
 -D_5 \partial_x c_5(t, x) &= \nu_{35} \cdot r_{\text{BV},3} \\
 \frac{i_{\text{tot}}}{FA_{\Gamma_e}} &= \nu_{06} \cdot r_{\text{BV},0} + \nu_{16} \cdot r_{\text{BV},1} + \nu_{26} \cdot r_{\text{BV},2} + \nu_{36} \cdot r_{\text{BV},3}
 \end{aligned} \tag{3.42}$$

zusammen. Die letzte Gleichung beschreibt die Gesamtstromdichte des Systems als Summe der einzelnen Teilströme i_j (siehe auch Gleichung (6) in [73])

$$\frac{i_{\text{tot}}}{FA_{\Gamma_e}} = \sum_{j=0}^{N_{\text{ET}}-1} i_j \tag{3.43}$$

wobei N_{ET} die Anzahl der im Mechanismus vertretenen ETs beschreibt (hier $N_{\text{ET}} = 4$). Für die Teilströme gilt

$$i_j = \nu_{j,e} \cdot r_{\text{BV},j} \tag{3.44}$$

(hier $\nu_{j,e} = \nu_{j6}$). Mit $c = (c_0(t, x), c_1(t, x), \dots)^T$ lassen sich die Randbedingungen der Substanzkonzentrationen in (3.42) allgemein durch die Matrix-Vektor-Multiplikation

$$-D \cdot \partial_x c = S_{\text{ET}}^T \cdot \mathbf{r}_{\text{ET}} \tag{3.45}$$

ausdrücken, wobei die Matrix S_{ET} die Zeilen aus S enthält, die den VETs entsprechen. Die Spalten der adsorbierten Teilchen und der Elektronen (v_e) werden in S_{ET} nicht berücksichtigt. Letztere dienen jedoch zur Berechnung der Gesamtstromdichte (3.43).

$$\frac{i_{\text{tot}}}{FA_{\Gamma_e}} = v_e^T \cdot \mathbf{r}_{\text{ET}} \tag{3.46}$$

Der Vektor $\mathbf{r}_{\text{ET}} = (r_{\text{BV},0}, r_{\text{BV},1}, \dots)^T$ enthält die einzelnen Geschwindigkeitsgesetze der VETs (hier: r_{BV}). Für Systeme aus OETs gelten analoge Beziehungen (Abschnitt 3.7).

Das auf makroskopischen Konzepten basierende Butler-Volmer-Modell beschreibt die Kinetik der Elektronentransfers mit Hilfe der phänomenologischen Parameter k_h und α . Durch den Vergleich mit experimentellen Daten kann das Modell dadurch eine Aussage

über die Geschwindigkeit eines Elektronentransfers liefern, nicht jedoch über die mikroskopischen Eigenschaften der beteiligten Moleküle. Der Einfluss auf die Kinetik eines ETs, der durch die Veränderung in der Molekülstruktur sowie einer veränderten Molekülumgebung resultiert, wird durch das Modell von Marcus (*Marcus-Theorie*³) behandelt[83–89]. In der Theorie des aktivierten Komplexes werden mikroskopische Moleküleigenschaften in der sogenannten Reaktionskoordinate zusammengefasst. Die Marcus-Theorie liefert den Zusammenhang zwischen der Reaktionskoordinate (Molekülstruktur, Solvatisierung) und der freien Aktivierungsenergie ΔG^\ddagger eines ETs. So erhält man z.B. für die Hinreaktion einer Reduktion an einer Elektrode die Beziehung[87, 89] (vgl. Gleichung (3.6.11b) in[51])

$$\Delta G_f^\ddagger = \frac{\lambda}{4} \left(1 - \frac{\nu_e F (E - E^0) + (w_A - w_B)}{\lambda} \right)^2 \quad (3.47)$$

wobei sich die sogenannte *Reorganisationsenergie* λ

$$\lambda = \lambda_i + \lambda_o \quad (3.48)$$

aus der inneren λ_i (beschreibt Veränderungen in Bindungslängen, -winkel) und äußeren λ_o (beschreibt Veränderungen der Solvathülle) Reorganisationsenergie zusammensetzt. Die Arbeitsterme w stehen für weitere Energiebeträge, die z.B. die elektrostatischen Wechselwirkungen zwischen der Elektrode und den Redoxmolekülen beinhalten.

Im Vergleich zum Butler-Volmer-Modell folgt aus dem mikroskopischen Modell von Marcus eine Beziehung zwischen Transferkoeffizient α und dem Elektrodenpotential. Aus Gleichung (3.34) erhält man durch partielle Differentiation nach dem Potential die Beziehung

$$\alpha = -\frac{1}{\nu_e F} \cdot \frac{\partial \Delta G_f^\ddagger}{\partial E} \quad (3.49)$$

sodass mit Gleichung (3.47)

$$\alpha = \frac{1}{2} - \frac{\nu_e F (E - E^0) + (w_A - w_B)}{2\lambda} \quad (3.50)$$

folgt (vgl. Gleichung (3.6.15b) in[51]). Während der phänomenologische Wert $\alpha \approx 0.5$ bestätigt wird, prognostiziert Gleichung (3.50) zudem eine Abhängigkeit des Transferkoeffizienten vom Elektrodenpotential.

Die Herleitung allgemeiner Modellgleichungen auf Basis der Gleichungen (3.38) und (3.50) wurde in dieser Arbeit nicht geleistet. Das Einbringen der Marcus-Kinetik in Analogie zu Gleichung (3.38) als zusätzlicher Baustein des Gesamtmodells sollte jedoch ohne größeren Aufwand möglich sein (siehe dazu auch die Diskussion in Abschnitt 4.3.8).

³Marcus erhielt "Für Forschungen zum Übergang von Elektronen und die damit verbundenen Änderungen des Energiezustandes" 1992 den Nobelpreis für Chemie[80–82].

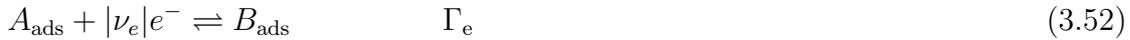
3.7. Adsorptionen und Oberflächenelektronentransfers

Elektrochemische Analysen können durch das Auftreten von Adsorptionsvorgängen des Typs



an der Elektrodenoberfläche beeinflusst werden (Abbildung 3.1, Prozess IV). Bei konstanter Temperatur liefern Adsorptionsisothermen den Zusammenhang zwischen den Konzentrationen der Substanz auf der Oberfläche (A_{ads}) und in der Volumenphase (A). Einen Überblick über Adsorptionsisothermen gibt Volke[90,91].

Im Zusammenhang mit OETs (Prozess III)



ist die thermodynamische Kompatibilität zwischen den Adsorptionsisothermen und dem kinetischen Elektronentransfermodell zu beachten[78,92] (siehe auch Abschnitt 4.3.7). Zahlreiche Publikationen beschreiben die Modellierung und Simulation von an Elektronentransfers gekoppelten Adsorptionsprozessen, berücksichtigen dabei jedoch stets nur spezielle Reaktionsmechanismen. Ein allgemeines Modellierungskonzept fehlt bis heute. Die Geschwindigkeit der Hin- (Adsorption) und Rückreaktion (Desorption) von (3.51) wird in den meisten elektrochemischen Arbeiten als ausreichend schnell angenommen, sodass für die mathematische Beschreibung bisher überwiegend die Gleichgewichtsformen der Isothermen verwendet wurden[78,93–106]. Die am häufigsten angenommenen Ausdrücke sind die der Langmuir-[93–97,105,106], der Frumkin-[98–101], sowie die der Henryisotherme[102–105].

Eine allgemeinere Behandlung von Adsorptionsvorgängen sowie Analyseverfahren mit hoher Zeitauflösung verlangen hingegen nach einer kinetischen Beschreibung. Dynamische Adsorptionsprozesse wurden in der Elektrochemie bisher entweder nur konzeptionell[33,107,108] oder für spezielle Systeme behandelt[109–112]. Ein Simulationsmodell, welches die dynamische Adsorption eines Moleküls und dessen OET enthält (Abbildung 3.1, ohne die Prozesse II, V und VI), wurde von Chevallier et al.[112] beschrieben. Weitere Modelle, die z.B. VETs (Prozess II) sowie Folgereaktionen zwischen Teilchen in Lösung (Prozess V) und auf der Elektrodenoberfläche (Prozess VI) enthalten, wurden teilweise in den Arbeiten von Laviron[95,106] und Mirčeski und Lovrić[104] behandelt. Im Folgenden sollen all diese Vorgänge durch ein allgemeines Rahmenwerk wiedergegeben werden.

Für einen Adsorptionsvorgang des Typs (3.51) ergibt sich die Bildung der adsorbierten Substanz wie üblich aus den Geschwindigkeiten der Hin- und Rückreaktion

$$\partial_t c_{\text{ads}} = r_f - r_b \quad (3.53)$$

wobei die Form der Terme r_f und r_b durch die jeweilige Adsorptionsisotherme bestimmt wird. Wir wollen uns an dieser Stelle auf die dynamische Form der Langmuir-Isotherme

beschränken, weisen jedoch darauf hin, dass auch andere Modelle (z.B. Henry-, Temkin-, Frumkin-Isothermen) eingebracht werden können. Die Adsorptions- und Desorptionsgeschwindigkeitskonstanten werden als potentialunabhängig angenommen.⁴ Doppelschichteffekte sowie Kapazitätsströme sollen vernachlässigt werden[95, 98, 104, 106, 112]. Weitere Voraussetzungen des Langmuirmodells werden z.B. in[78, 90, 91] diskutiert. Bezieht man co-adsorbierte Moleküle in die Betrachtung ein, erhält man nach dem dynamischen Modell von Langmuir den Ausdruck[75]

$$\partial_t c_{\text{ads},k}(t) = k_f \left[c_{\text{sat}} - \sum_i c_{\text{ads},i}(t) \right] \cdot c_j(t) - k_b \cdot c_{\text{ads},k}(t) \quad (3.54)$$

der den zeitlichen Verlauf der Oberflächenkonzentration c_{ads} der adsorbierten Substanz mit Index k beschreibt. Die rechte Seite setzt sich aus den Geschwindigkeitskonstanten k_f und k_b , der Sättigungskonzentration c_{sat} , der Summe der Konzentrationen aller auf derselben Oberfläche adsorbierten Teilchen und der Volumenkonzentration c_j der gelösten Form an der Oberfläche Γ_e zusammen. Wir definieren für eine Adsorption der Form (3.51) das *Langmuir-Gesetz*

$$r_L = k_f \left[c_{\text{sat}} - \sum_i c_{\text{ads},i}(t) \right] \cdot c_A(t) - k_b \cdot c_{\text{ads},A}(t) \quad (3.55)$$

Befindet sich die Adsorption (3.51) im Gleichgewicht ($\partial_t c_{\text{ads},k} = 0$ in Gleichung (3.54)), so erhält man die Langmuir-Isotherme

$$K = \frac{k_f}{k_b} = \frac{c_{\text{ads},k}}{\left[c_{\text{sat}} - \sum_i c_{\text{ads},i} \right] \cdot c_j} \quad (3.56)$$

Betrachtet man beispielsweise zwei an einen VET gekoppelte Adsorptionsreaktionen



gelten statt der Beziehung (3.23) zwischen Massenfluss und elektrischer Stromdichte die Gleichungen (vgl. Gleichung (14.3.1) in[51])

$$\begin{array}{ll} -D_A \partial_x c_A(t, x) = -\frac{i}{\nu_e F A_{\Gamma_e}} + \nu_{12} \cdot r_{L,1} & x \in \Gamma_e \\ -D_B \partial_x c_B(t, x) = \frac{i}{\nu_e F A_{\Gamma_e}} + \nu_{23} \cdot r_{L,2} & x \in \Gamma_e \end{array} \quad (3.58)$$

⁴Ein Modell, welches potentialabhängige Geschwindigkeitskonstanten beinhaltet, findet man z.B. in[109].

Die Randgleichungen lauten allgemein

$$\begin{aligned}
 \partial_t c_{\text{ads},0}(t) &= \nu_{10} \cdot r_{L,1} \\
 \partial_t c_{\text{ads},1}(t) &= \nu_{21} \cdot r_{L,2} \\
 -D_2 \partial_x c_2(t, x) &= \nu_{02} \cdot r_{\text{BV},0} + \nu_{12} \cdot r_{L,1} \\
 -D_3 \partial_x c_3(t, x) &= \nu_{03} \cdot r_{\text{BV},0} + \nu_{23} \cdot r_{L,2} \\
 i_{\text{tot}} / (FA_{\Gamma_e}) &= \nu_{04} \cdot r_{\text{BV},0}
 \end{aligned} \tag{3.59}$$

wobei $r_{\text{BV},0}$ das Butler-Volmer-Gesetz (3.38) der ersten Reaktion und $r_{L,1}$ bzw. $r_{L,2}$ die Langmuir-Gesetze (3.55) der beiden Adsorptionen beschreiben. Es sei darauf hingewiesen, dass die Terme r_{BV} und r_L spezielle Modellgleichungen für VETs und Adsorptionen darstellen. Der allgemeine Aufbau der Gleichungen (3.59) bleibt jedoch auch für andere Modelltypen gültig.

Wie schon bei der Behandlung von VETs (Abschnitt 3.6) gelten für einen einfachen OET weiterhin die Gleichungen (3.21) und (3.22), wobei die Volumenkonzentrationen $c_A(t, x)$ und $c_B(t, x)$ durch die Oberflächenkonzentrationen $c_{A_{\text{ads}}}(t, x)$ und $c_{B_{\text{ads}}}(t, x)$ der elektroaktiven Teilchen ersetzt werden. Des Weiteren gilt für den aus dem OET resultierenden Strom i_a

$$\begin{aligned}
 \partial_t c_{A_{\text{ads}}}(t, x) &= -\frac{i_a}{\nu_e FA_{\Gamma_e}} & x \in \Gamma_e \\
 \partial_t c_{B_{\text{ads}}}(t, x) &= \frac{i_a}{\nu_e FA_{\Gamma_e}} & x \in \Gamma_e
 \end{aligned} \tag{3.60}$$

woraus mit Gleichung (3.22)

$$\begin{aligned}
 \partial_t c_{A_{\text{ads}}}(t, x) &= -[k_f c_{A_{\text{ads}}}(t, x) - k_b c_{B_{\text{ads}}}(t, x)] & x \in \Gamma_e \\
 \partial_t c_{B_{\text{ads}}}(t, x) &= [k_f c_{A_{\text{ads}}}(t, x) - k_b c_{B_{\text{ads}}}(t, x)] & x \in \Gamma_e
 \end{aligned} \tag{3.61}$$

folgt. Eine zu einer Langmuir-Isothermen konforme Strom-Potential-Charakteristik lautet [78, 92]

$$i_a = \nu_e FA_{\Gamma_e} k_s \left[e^{\alpha \nu_e \frac{F}{RT}(E-E^0)} c_{A_{\text{ads}}}(t, x) - e^{-(1-\alpha)\nu_e \frac{F}{RT}(E-E^0)} c_{B_{\text{ads}}}(t, x) \right] \tag{3.62}$$

für $x \in \Gamma_e$ und mit der heterogenen Geschwindigkeitskonstante des OETs, k_s/s^{-1} . Bis auf die Einheiten von k_s und der Konzentrationen entspricht (3.62) der Gleichung (3.36) des Butler-Volmer-Modells. Für die Oberflächenkonzentrationen folgt

$$\begin{aligned}
 \partial_t c_{A_{\text{ads}}}(t, x) &= \nu_A k_s \left[e^{\alpha \nu_e \frac{F}{RT}(E-E^0)} c_{A_{\text{ads}}}(t, x) - e^{-(1-\alpha)\nu_e \frac{F}{RT}(E-E^0)} c_{B_{\text{ads}}}(t, x) \right] \\
 \partial_t c_{B_{\text{ads}}}(t, x) &= \nu_B k_s \left[e^{\alpha \nu_e \frac{F}{RT}(E-E^0)} c_{A_{\text{ads}}}(t, x) - e^{-(1-\alpha)\nu_e \frac{F}{RT}(E-E^0)} c_{B_{\text{ads}}}(t, x) \right]
 \end{aligned} \tag{3.63}$$

für $x \in \Gamma_e$, wobei das Vorzeichen durch die Stöchiometrikoeffizienten ν_A und ν_B eingebracht wird. Analog zu (3.38) definieren wir das Butler-Volmer-Gesetz eines OETs

$$r_{\text{BV}} = k_s \left[e^{\alpha \nu_e \frac{F}{RT} (E - E^0)} c_{\text{Edukt}} - e^{-(1-\alpha) \nu_e \frac{F}{RT} (E - E^0)} c_{\text{Produkt}} \right] \quad (3.64)$$

mit den Oberflächenkonzentrationen der Reaktionspartner c_{Edukt} und c_{Produkt} .

Betrachten wir nun ein System, das Adsorptionen, Oberflächen- und Volumenelektronentransfers verbindet

$$\begin{aligned} |\nu_{02}| \cdot A + |\nu_{04}| \cdot e^- &\rightleftharpoons |\nu_{03}| \cdot B && \Gamma_e \\ |\nu_{10}| \cdot A_{\text{ads}} + |\nu_{14}| \cdot e^- &\rightleftharpoons |\nu_{11}| \cdot B_{\text{ads}} && \Gamma_e \\ |\nu_{22}| \cdot A &\rightleftharpoons |\nu_{20}| \cdot A_{\text{ads}} && \Gamma_e \\ |\nu_{33}| \cdot B &\rightleftharpoons |\nu_{31}| \cdot B_{\text{ads}} && \Gamma_e \end{aligned} \quad (3.65)$$

so ergibt sich z.B für die Konzentration der Spezies A_{ads} die gewöhnliche Differentialgleichung (ODE)

$$\partial_t c_{\text{ads},0}(t) = \nu_{10} \cdot r_{\text{BV},1} + \nu_{20} \cdot r_{\text{L},2} \quad (3.66)$$

Insgesamt folgt das System

$$\begin{aligned} \partial_t c_{\text{ads},0}(t) &= \nu_{10} \cdot r_{\text{BV},1} + \nu_{20} \cdot r_{\text{L},2} \\ \partial_t c_{\text{ads},1}(t) &= \nu_{11} \cdot r_{\text{BV},1} + \nu_{31} \cdot r_{\text{L},3} \\ -D_2 \partial_x c_2(t, x) &= \nu_{02} \cdot r_{\text{BV},0} + \nu_{22} \cdot r_{\text{L},2} \\ -D_3 \partial_x c_3(t, x) &= \nu_{03} \cdot r_{\text{BV},0} + \nu_{33} \cdot r_{\text{L},3} \\ i_{\text{tot}} / (F A \Gamma_e) &= \nu_{04} \cdot r_{\text{BV},0} + \nu_{14} \cdot r_{\text{BV},1} \end{aligned} \quad (3.67)$$

wobei $r_{\text{BV},0}$ das Butler-Volmer-Gesetz (3.38) und $r_{\text{BV},1}$ das Gesetz (3.64) beschreiben. Für gekoppelte und parallele Elektronentransfers sollen weiterhin die Beziehungen des vorherigen Abschnittes gelten. Die Randgleichungen komplexer Reaktionsmechanismen lassen sich erneut mittels geeigneter Ausschnitte der Stöchiometriematrix erzeugen.

Sei S_{ads} die Matrix, welche aus den Stöchiometrievektoren der Reaktionen besteht, an denen adsorbierte Teilchen eines Randes beteiligt sind. Die Spalten, die den gelösten Substanzen oder den Elektronen entsprechen, sollen in S_{ads} nicht auftreten. Das System aus ODEs, das den Verlauf der Oberflächenkonzentrationen eines Randes beschreibt, ergibt sich somit nach

$$\partial_t c_{\text{ads}} = S_{\text{ads}}^T \cdot \mathbf{r}_{\text{ads}} \quad (3.68)$$

mit $c_{\text{ads}} = (c_{\text{ads},0}(t), c_{\text{ads},1}(t), \dots)^T$ und Vektor $\mathbf{r}_{\text{ads}} = (r_0, r_1, \dots)^T$, der die einzelnen Geschwindigkeitsgesetze der Reaktionen enthält, an denen adsorbierte Teilchen beteiligt sind. Die Randwerte für die Konzentrationen der gelösten Spezies setzen sich nach

$$-D \cdot \partial_x c = S_{\text{ET}}^T \cdot \mathbf{r}_{\text{ET}} + \tilde{S}_{\text{ads}}^T \cdot \tilde{\mathbf{r}}_{\text{ads}} \quad (3.69)$$

zusammen, wobei Vektor c die Volumenkonzentrationen und Matrix S_{ET} die Stöchiometrie der VETs repräsentieren. Matrix \tilde{S}_{ads} enthält die Stöchiometrievektoren der Adsorptionen eines Randes. Die Spalten, die den adsorbierten Substanzen oder den Elektronen entsprechen, sollen in \tilde{S}_{ads} nicht auftreten. Der Vektor $\tilde{\mathbf{r}}_{\text{ads}}$ beinhaltet die Geschwindigkeitsgesetze der Adsorptionen.

Die Gesamtstromdichte an einer Elektrode ergibt sich analog zum vorherigen Abschnitt aus der Summe der Teilströme aller ETs einer Elektrode, wobei in Gleichung (3.46) der Vektor \mathbf{r}_{ET} die Geschwindigkeitsgesetze der VETs und OETs enthält.

3.8. Oberflächenreaktionen

Einen weiteren Bestandteil elektrochemischer Elektrodenprozesse stellen Oberflächenreaktionen dar. Im Allgemeinen lassen sich zwei verschiedene Reaktionstypen unterscheiden. So laufen *homogene Oberflächenreaktionen* zwischen adsorbierten Molekülen ab, z.B.



während *heterogene Oberflächenreaktionen* zwischen adsorbierten und gelösten Teilchen eintreten, z.B.



(Abbildung 3.1, Prozess VI). Im elektrochemischen Kontext wurde die Modellierung und Simulation solcher Prozesse erst vereinzelt begonnen[20, 21, 104, 106, 113]. In dieser Arbeit sollen für die Kinetik der Oberflächenreaktionen die zu den Geschwindigkeitsgesetzen (3.12), (3.13) und (3.16) homogener Reaktionen äquivalenten Ausdrücke verwendet werden. Es sei darauf hingewiesen, dass sich lediglich die physikalischen Einheiten der Größen in den Ausdrücken ändern. So besitzt z.B. für die Reaktion (3.70) die Geschwindigkeitskonstante des Potenzgesetzes der Hinreaktion die Einheit $\text{cm}^2\text{mol}^{-1}\text{s}^{-1}$, während für (3.71) die Einheit $\text{cm}^3\text{mol}^{-1}\text{s}^{-1}$ resultiert.

Die Beschreibung der Reaktionsterme für adsorbierte Spezies folgt auf natürliche Weise, indem Vektor \mathbf{r}_{ads} in Gleichung (3.68) zusätzlich die Reaktionsraten für Oberflächenreaktionen $r_{\text{P}_{\text{rev}}}$, $r_{\text{P}_{\text{irr}}}$ bzw. r_{G} enthalten soll.

Auf analoge Weise werden die Randgleichungen (3.69) der gelösten Spezies durch die Terme der heterogenen Oberflächenreaktionen erweitert. So werden in Matrix \tilde{S}_{ads} die Stöchiometrievektoren der heterogenen Oberflächenreaktionen berücksichtigt und Vektor $\tilde{\mathbf{r}}_{\text{ads}}$ beinhaltet zusätzlich die jeweiligen Geschwindigkeitsgesetze.

Für die Berechnung des Gleichgewichtszustandes werden Gleichungen der Form (3.18) verwendet.

3.9. Dimensioniertes Modell

In den Abschnitten 3.1 – 3.8 wurde ein Konzept zur allgemeinen Formulierung der Modellgleichungen für Diffusionsprozesse, homogene Reaktionen, heterogene Elektronentransfers, Adsorptionen und Oberflächenreaktionen in elektrochemischen Zellgeometrien vorgestellt. Eine zentrale Rolle spielte dabei die Stöchiometriematrix des Reaktionsmechanismus.

Im Inneren der elektrochemischen Zelle (Ω) beschreiben die Reaktions-Diffusions-Gleichungen (3.5) den zeitlichen Verlauf der Konzentrationen, wobei sich die Reaktionsterme $R(c)$ gemäß (3.17) ergeben. OETs und Adsorptionen sowie Oberflächenreaktionen bestimmen das Aussehen der ODEs für *adsorbierte* Teilchen (Gleichung (3.68)). Die *Randwerte* für *gelöste* Teilchen setzen sich aus den Geschwindigkeitsgesetzen der VETs, der Adsorptionen und der heterogenen Oberflächenreaktionen zusammen (Gleichung (3.69)). An inerten Wänden und dem Ende der Diffusionsschicht zur Lösung hin sowie für Teilchen, die weder adsorbieren noch elektroaktiv sind, werden homogene Neumann-Randbedingungen

$$\partial_x c(t, x) = 0 \quad x \in \Gamma \quad (3.72)$$

angenommen. Die Gesamtstromdichte an einer Elektrode ($x \in \Gamma_e$) berechnet sich aus der Summe der Teilströme aller ETs (Gleichung (3.43)).

Zu Beginn eines elektrochemischen Experiments wird angenommen, dass sich das System im Gleichgewichtszustand befindet. Die *Gleichgewichtskonzentrationen* für adsorbierte (c_{ads}^0) und gelöste (c^0) Spezies ergeben sich näherungsweise als Lösung des ODE-Systems

$$\left\{ \begin{array}{l} \partial_t c(t) = S_{\text{hom}}^T \cdot \mathbf{r}_{\text{hom}} \\ \partial_t c_{\text{ads}}(t) = S_{\text{ads}}^T \cdot \mathbf{r}_{\text{ads}} \\ c(0) = \tilde{c}^0 \\ c_{\text{ads}}(0) = \tilde{c}_{\text{ads}}^0 \end{array} \right. \quad t > 0 \quad (3.73)$$

sobald sämtliche Massenwirkungsgesetze erfüllt sind. Die Anfangswerte in (3.73) entsprechen den Einwaagekonzentrationen \tilde{c}^0 der Spezies.⁵

Insgesamt erhält man je nach der Komplexität des elektrochemischen Reaktionsmechanismus ein System (3.74) aus gekoppelten, nicht-linearen PDEs. Die Randgleichungen beschreiben zudem ein System aus Neumann-Bedingungen (enthalten den trivialen Fall

⁵Ein thermodynamischer Ansatz zur Berechnung der Gleichgewichtskonzentrationen wird in [114] beschrieben und mit der hier gewählten Methode verglichen.

$\partial_x c(t, x) = 0$) in Verbindung mit einem System aus ODEs.

$$\left\{ \begin{array}{ll} \partial_t c(t, x) = D \cdot \partial_{xx} c(t, x) + S_{\text{hom}}^T \cdot \mathbf{r}_{\text{hom}} & \text{in } (0, t_{\text{max}}] \times \Omega \\ -D \cdot \partial_x c(t, x) = S_{\text{ET}}^T \cdot \mathbf{r}_{\text{ET}} + \tilde{S}_{\text{ads}}^T \cdot \tilde{\mathbf{r}}_{\text{ads}} & \text{auf } (0, t_{\text{max}}] \times \Gamma \\ \partial_t c_{\text{ads}}(t, x) = S_{\text{ads}}^T \cdot \mathbf{r}_{\text{ads}} & \text{auf } (0, t_{\text{max}}] \times \Gamma \\ c(0, x) = c^0 & \text{auf } \bar{\Omega} \\ c_{\text{ads}}(0, x) = c_{\text{ads}}^0 & \text{auf } \Gamma \end{array} \right. \quad (3.74)$$

Die Berechnung der Observablen aus den Konzentrationen erfolgt mit Hilfe der Gleichung (3.43) durch das *Strom-Potential-Residuum*⁶

$$0 = \frac{i_{\text{tot}}}{FA_{\Gamma_e}} - \sum_{j=0}^{N_{\text{ET}}-1} \nu_{j,e} \cdot r_{\text{ET},j} \quad \text{auf } (0, t_{\text{max}}] \times \Gamma \quad (3.75)$$

und liefert im Falle eines potentialkontrollierten Experiments den elektrischen Strom i_{tot} an einer Elektrode. Für stromkontrollierte Experimente muss Gleichung (3.75) nach dem Potential E gelöst werden, welches implizit durch die Geschwindigkeitsgesetze r_{ET} der Elektronentransfers (z.B. die Butler-Volmer-Terme r_{BV} (3.38) und (3.64)) gegeben ist.

3.10. Dimensionsloses Modell

Die Modellgleichungen (3.74) und (3.75) beinhalten mehrere Größen (Zeit, Länge, Volumenkonzentrationen, Diffusionskoeffizienten, Oberflächenkonzentrationen, Strom, Potential, ...) in unterschiedlichen physikalischen Einheiten. Im Hinblick auf eine numerische Behandlung des Modells stellt sich die Frage nach einer geeigneten Skalierung, um große Unterschiede in den numerischen Werten der Größen zu vermeiden. Ein erster Schritt ist die Transformation des Modells in eine *dimensionslose* Form, in der sämtliche Größen ohne ihre physikalischen Einheiten erscheinen. Oft bietet sich dazu eine *Normalisierung* mittels sogenannter Referenzwerte an. Tabelle 3.2 gibt einen Überblick über die dimensionslosen und normalisierten Größen des hier verwendeten Modells.

Die Zeit wird mit der Dauer des elektrochemischen Experiments \tilde{t}_{max} normalisiert. Die

⁶Die Bezeichnung *Strom-Potential-Residuum* deutet auf die Möglichkeit hin, den Rest, um den Gleichung (3.75) aufgrund numerischer Approximationsfehler nicht erfüllt wird, durch eine geeignete Strategie möglichst klein zu halten (siehe Kapitel 6).

resultierende dimensionslose Zeitgröße soll im Folgenden weiterhin mit t bezeichnet werden. Die Referenzlänge L_{ref} beschreibt im Falle eines semi-infiniten Diffusionsraums die Dicke der Nernstschen Diffusionsschicht[51]

$$L_{\text{ref}} = \zeta \cdot \sqrt{D_{\text{max}} \tilde{t}_{\text{max}}} \quad (3.76)$$

wobei ζ einen Sicherheitsfaktor (hier wird $\zeta = 3$ verwendet) und D_{max} den größten Diffusionskoeffizienten beschreiben. Im Falle eines finiten Diffusionsraum steht L_{ref} für den Abstand zweier Elektroden bzw. zwischen der Elektrode und einer inerten Wand. Die resultierende dimensionslose Größe soll weiterhin mit x bezeichnet werden.

Die Volumenkonzentrationen werden mit der maximalen Konzentration c_{max} , die Diffusionskoeffizienten mit D_{max} normalisiert.

Für die Transformation der Oberflächenkonzentrationen wird der Quotient aus der Sättigungskonzentration c_s und

$$\varphi = \frac{c_s L_{\text{ref}}}{\tilde{t}_{\text{max}} D_{\text{max}} c_{\text{max}}} \quad (3.77)$$

verwendet. Die Einbringung des Faktors φ ist durch die Transformation der Randbedingungen (3.69) motiviert und erleichtert eine konsistente Beschreibung der dimensionslosen Modellgleichungen.⁷

Aus den in Tabelle 3.2 angegebenen Definitionen ergeben sich die Reaktions-Diffusions-Gleichungen in dimensionsloser Form

$$\partial_t u = \rho \cdot \beta \cdot \partial_{xx} u + R(u) \quad (3.78)$$

mit dem Faktor

$$\rho = \frac{D_{\text{max}} \tilde{t}_{\text{max}}}{L_{\text{ref}}^2} \quad (3.79)$$

⁷Verwendet man die Definitionen aus Tabelle 3.2 zur Transformation von Gleichung (3.69), erhält man durch die Definition von φ konsistente dimensionslose Randterme.

Tabelle 3.2: Dimensionslose Größen.

dimensionierte Größe	Symbol	SI-Einheit	Referenzwert	normalisierte Größe
Zeit	\tilde{t}	s	t_{max}	$t = \tilde{t}/\tilde{t}_{\text{max}}$
Länge	\tilde{x}	m	L_{ref}	$x = \tilde{x}/L_{\text{ref}}$
Volumenkonzentration	c	$\text{mol} \cdot \text{m}^{-3}$	c_{max}	$u = c/c_{\text{max}}$
Diffusionskoeffizient	D	$\text{m}^2 \text{s}^{-1}$	D_{max}	$\beta = D/D_{\text{max}}$
Oberflächenkonzentration	c_{ads}	$\text{mol} \cdot \text{m}^{-2}$	c_s/φ	$w = \varphi \cdot c_{\text{ads}}/c_s$ ^{a)}

^{a)}Zur Bedeutung des Faktors φ siehe Gleichung (3.77).

3. Modellanalyse

und $\beta = \text{diag}(\beta_0, \beta_1, \dots)$. Zwischen dimensionslosen und dimensionierten Reaktionstermen gilt die Beziehung

$$R(u) = \frac{\tilde{t}_{\max}}{c_{\max}} \cdot R(c) \quad (3.80)$$

Aus den Potenzgesetzen (3.12) und (3.13) und mit der Beziehung $c = u \cdot c_{\max}$ folgen somit die Ausdrücke für reversible Reaktionen

$$r_{\text{P}_{\text{rev}}} = \kappa_{\text{f}} \prod_k u_k^{\alpha_k} - \kappa_{\text{b}} \prod_k u_k^{\beta_k} \quad (3.81)$$

und für irreversible Reaktionen

$$r_{\text{P}_{\text{irr}}} = \kappa_{\text{f}} \prod_k u_k^{\alpha_k} \quad (3.82)$$

mit den Geschwindigkeitskonstanten

$$\begin{aligned} \kappa_{\text{f}} &= k_{\text{f}} \cdot \tilde{t}_{\max} \cdot c_{\max}^{(\sum_k \alpha_k) - 1} \\ \kappa_{\text{b}} &= k_{\text{b}} \cdot \tilde{t}_{\max} \cdot c_{\max}^{(\sum_k \beta_k) - 1} \end{aligned} \quad (3.83)$$

Das generische Gesetz der Form (3.16) wird zu

$$r_{\text{G}} = f(u) \quad (3.84)$$

wobei die Transformationen in die rechte Seite $f(u)$ eingehen. Die Formen der Butler-Volmer-Gesetze (3.38) und (3.64) lauten für VETs

$$r_{\text{BV}} = \psi_{\text{h}} \left[e^{\alpha \nu_e (p - p^0)} u_{\text{Edukt}} - e^{-(1 - \alpha) \nu_e (p - p^0)} u_{\text{Produkt}} \right] \quad (3.85)$$

und für OETs

$$r_{\text{BV}} = \psi_{\text{s}} \left[e^{\alpha \nu_e (p - p^0)} w_{\text{Edukt}} - e^{-(1 - \alpha) \nu_e (p - p^0)} w_{\text{Produkt}} \right] \quad (3.86)$$

mit dimensionslosem Potential

$$p = \frac{F}{RT} \cdot E \quad (3.87)$$

und den heterogenen Geschwindigkeitskonstanten

$$\begin{aligned} \psi_h &= k_h \cdot \frac{L_{\max}}{D_{\max}} \\ \psi_s &= k_s \cdot \tilde{t}_{\max} \end{aligned} \quad (3.88)$$

Des Weiteren folgt für das Langmuir-Gesetz (3.55) die Form

$$r_L = \kappa_{\text{ads}} \left[\varphi - \sum_i w_i \right] \cdot u_A - \kappa_{\text{des}} \cdot w_A \quad (3.89)$$

mit den Geschwindigkeitskonstanten

$$\begin{aligned} \kappa_{\text{ads}} &= k_f \cdot \tilde{t}_{\max} \cdot c_{\max} \\ \kappa_{\text{des}} &= k_b \cdot \tilde{t}_{\max} \end{aligned} \quad (3.90)$$

Für die Potenzgesetze der homogenen ($v = w$) und heterogenen ($v = w$ oder $v = u$) Oberflächenreaktionen ergeben sich die Gleichungen

$$r_{\text{P}_{\text{rev}}} = \kappa_f \prod_k v_k^{\alpha_k} - \kappa_b \prod_k v_k^{\beta_k} \quad (3.91)$$

bzw.

$$r_{\text{P}_{\text{irr}}} = \kappa_f \prod_k v_k^{\alpha_k} \quad (3.92)$$

mit den Geschwindigkeitskonstanten für homogene

$$\begin{aligned} \kappa_f &= k_f \cdot \tilde{t}_{\max} \cdot (c_s/\varphi)^{(\sum_k \alpha_k)-1} \\ \kappa_b &= k_b \cdot \tilde{t}_{\max} \cdot (c_s/\varphi)^{(\sum_k \beta_k)-1} \end{aligned} \quad (3.93)$$

und heterogene Oberflächenreaktionen

$$\begin{aligned} \kappa_f &= k_f \cdot \tilde{t}_{\max} \cdot (c_s/\varphi)^{(\sum_k \alpha_k)-1} \cdot c_{\max}^{(\sum_j \alpha_j)} \\ \kappa_b &= k_b \cdot \tilde{t}_{\max} \cdot (c_s/\varphi)^{(\sum_k \beta_k)-1} \cdot c_{\max}^{(\sum_j \beta_j)} \end{aligned} \quad (3.94)$$

Die generischen Gesetze auf Oberflächen werden analog zu Gleichung (3.84)

$$\boxed{r_G = f(u, w)} \quad (3.95)$$

behandelt. Dabei können sowohl Volumen- als auch Oberflächenkonzentrationen eingesetzt werden.

Insgesamt lassen sich nun die Differentialgleichungssysteme (3.73) und (3.74) in den Größen u und w ausdrücken. Die Berechnung der Anfangswerte erfolgt durch die Lösungen u^0 und w^0 des ODE-Systems

$$\boxed{\begin{cases} \partial_t u(t) = S_{\text{hom}}^T \cdot \mathbf{r}_{\text{hom}} \\ \partial_t w(t) = S_{\text{ads}}^T \cdot \mathbf{r}_{\text{ads}} \\ u(0) = \tilde{u}^0 \\ w(0) = \tilde{w}^0 \end{cases} \quad t > 0} \quad (3.96)$$

Die dimensionslosen elektrochemischen Modellgleichungen lauten

$$\boxed{\begin{cases} \partial_t u(t, x) = \rho \cdot \beta \cdot \partial_{xx} u(t, x) + S_{\text{hom}}^T \cdot \mathbf{r}_{\text{hom}} & \text{in } (0, 1] \times \mathcal{I} \\ -\beta \cdot \partial_x u(t, x) = S_{\text{ET}}^T \cdot \mathbf{r}_{\text{ET}} + \tilde{S}_{\text{ads}}^T \cdot \tilde{\mathbf{r}}_{\text{ads}} & \text{auf } (0, 1] \times \Gamma \\ \partial_t w(t, x) = S_{\text{ads}}^T \cdot \mathbf{r}_{\text{ads}} & \text{auf } (0, 1] \times \Gamma \\ u(0, x) = u^0 & \text{auf } \bar{\mathcal{I}} \\ w(0, x) = w^0 & \text{auf } \Gamma \end{cases}} \quad (3.97)$$

mit dem Intervall $\mathcal{I} = (0, 1)$. Die Vektoren \mathbf{r}_{hom} , \mathbf{r}_{ET} , \mathbf{r}_{ads} und $\tilde{\mathbf{r}}_{\text{ads}}$ enthalten die jeweiligen dimensionslosen Formen. Die Berechnung der Observablen erfolgt mit Hilfe der Reaktionsraten der Elektronentransfers

$$\boxed{0 = \chi - \sum_{j=0}^{N_{\text{ET}}-1} \nu_{j,e} \cdot r_{\text{ET},j}} \quad (3.98)$$

mit dimensionslosem Strom

$$\chi = \frac{L_{\text{ref}}}{F A_{\Gamma_e} D_{\text{max}} c_{\text{max}}} \cdot i_{\text{tot}} \quad (3.99)$$

Es sei betont, dass die Gleichungen (3.97) zusammen mit (3.98) als ein partiell-differentiell-algebraisches System (PDAE-System) in den Unbekannten u , w und χ bzw. p betrachtet werden können. Sie bestehen aus einem parabolischen PDE-System im Inneren der Zelle, kombiniert mit differentiell-algebraischen Randgleichungen (DAEs) an den Elektrodenoberflächen. Die DAEs setzen sich aus dem ODE-System der adsorbierten Spezies, den Randwerten der gelösten Formen (siehe Randgleichungen in (3.97)) sowie der algebraischen Gleichung (3.98) zusammen.⁸ Die Betrachtung der Gleichungen als PDAE-System führt zu einer verallgemeinerten Formulierung des Simulationsmodells, welches in Kapitel 6 vorgestellt wird.

3.11. Strom, Potential und Massenfluss

Im Falle eines potentialkontrollierten Experiments ist die gesuchte Größe der durch eine Arbeitselektrode fließende Strom. Zur Berechnung des Stroms wurde bisher in der Literatur stets der Massenfluss elektroaktiver Partikel an der Elektrode betrachtet[1, 34, 36, 42, 98, 115–118]. Für einen einfachen VET (3.20) liefert Gleichung (3.23)

$$\frac{i}{\nu_e F A_{\Gamma_e}} = D_A \partial_x c_A(t, x) \quad x \in \Gamma_e \quad (3.100)$$

den Zusammenhang zwischen Strom auf der einen und Massenfluss auf der anderen Seite. Die wohl am häufigsten angewandte Strategie zur numerischen Berechnung des Stroms stellt die Approximation des Gradienten $\partial_x c$ aus den zuvor berechneten diskreten Konzentrationsprofilen dar[33, 34, 42, 98, 115–117]. Andere Arbeiten verwenden die Strom-Massenfluss-Beziehung zur Herleitung alternativer Ansätze[36, 118].

Während Gleichung (3.100) noch sehr einfach ist, müssen für komplexere Mechanismen weitaus kompliziertere Beziehungen abgeleitet werden[33, 98]. Eine allgemeingültige Formulierung, die neben Elektronentransfers auch Adsorptionsvorgänge und Oberflächenreaktionen berücksichtigt, fehlt bis heute. Wir wollen daher in dieser Arbeit einen allgemeineren Ansatz wählen.

Den Zusammenhang zwischen Strom, Potential und Oberflächenkonzentrationen liefert für jeden Reaktionsmechanismus das *Strom-Potential-Residuum* (3.75) bzw. (3.98). Bei bekanntem Potential und bekannten Oberflächenkonzentrationen lässt sich der gesuchte Strom aus dem Strom-Potential-Residuum direkt berechnen. In diesem Fall kann auf eine Approximation der Gradienten vollständig verzichtet werden. Es sei an dieser Stelle betont, dass Gleichung (3.75) (bzw. (3.98)) auch im stromkontrollierten Fall zur Berechnung des gesuchten Elektrodenpotentials dient. Sie stellt somit die Grundlage zur Modellierung und Simulation potential- und stromkontrollierter elektrochemischer Experimente dar.

Auf der Basis des Strom-Potential-Residuums und den Randgleichungen aus (3.97) lassen

⁸Eine ähnliche Klassifizierung elektrochemischer Modellgleichungen findet man in[73].

sich zudem allgemeine Ausdrücke herleiten, die es ermöglichen für sämtliche Reaktionsmechanismen eine Beziehung zwischen Strom, dem Massenfluss gelöster Teilchen und den Zeitableitungen der Oberflächenkonzentrationen adsorbierter Formen aufzubauen. Die Ausdrücke können somit für potentialkontrollierte Experimente zur Stromberechnung alternativ verwendet werden.⁹

Dazu stellen wir uns einen beliebigen Reaktionsmechanismus aus Elektronentransfers, Adsorptionen und Oberflächenreaktionen vor (homogene Reaktionen spielen in der Herleitung der Randgleichung keine Rolle und können somit vernachlässigt werden). Nach Gleichung (3.98) ergibt sich der Strom aus der Summe der Reaktionsraten aller im Mechanismus vorkommender VETs und OETs:

$$\chi = \sum_i \nu_{i,e} \cdot r_{\text{VET},i} + \sum_j \nu_{j,e} \cdot r_{\text{OET},j} \quad (3.101)$$

Sei $f_k = D_k \partial_x c_k$ der Massenfluss einer Spezies mit Index k , dann lautet die Randgleichung der Spezies

$$-f_k = \sum_i \nu_{ik} \cdot r_{\text{VET},i} + \sum_l \nu_{lk} \cdot r_{\text{ADS},l} + \sum_m \nu_{mk} \cdot r_{\text{OFR},m} \quad (3.102)$$

wobei sich die Summen aus den Raten der VETs, der Adsorptionen (r_{ADS}) und der Oberflächenreaktionen (r_{OFR}) bilden. Für die Zeitableitung der Oberflächenkonzentration einer adsorbierten Form gilt

$$\partial_t w_k = \sum_j \nu_{jk} \cdot r_{\text{OET},j} + \sum_l \nu_{lk} \cdot r_{\text{ADS},l} + \sum_m \nu_{mk} \cdot r_{\text{OFR},m} \quad (3.103)$$

Aus Gleichung (3.102) leiten wir zunächst die Ausdrücke

$$\begin{aligned} r_{\text{VET},i} &= \frac{1}{-\nu_{ik}} f_k + \frac{1}{-\nu_{ik}} \left(\sum_{n,n \neq i} \nu_{nk} \cdot r_{\text{VET},n} + \sum_l \nu_{lk} \cdot r_{\text{ADS},l} + \sum_m \nu_{mk} \cdot r_{\text{OFR},m} \right) \\ r_{\text{ADS},l} &= \frac{1}{-\nu_{lk}} f_k + \frac{1}{-\nu_{lk}} \left(\sum_i \nu_{ik} \cdot r_{\text{VET},i} + \sum_{n,n \neq l} \nu_{nk} \cdot r_{\text{ADS},n} + \sum_m \nu_{mk} \cdot r_{\text{OFR},m} \right) \\ r_{\text{OFR},m} &= \frac{1}{-\nu_{mk}} f_k + \frac{1}{-\nu_{mk}} \left(\sum_i \nu_{ik} \cdot r_{\text{VET},i} + \sum_l \nu_{lk} \cdot r_{\text{ADS},l} + \sum_{n,n \neq m} \nu_{nk} \cdot r_{\text{OFR},n} \right) \end{aligned} \quad (3.104)$$

⁹Im stromkontrollierten Fall ist diese Variante nicht möglich. Die Berechnung des Elektrodenpotentials erfolgt dabei stets über die Lösung einer nicht-linearen Gleichung (hier Gleichung (3.98)).

ab, indem jeweils nach den einzelnen Reaktionsraten r_{VET} , r_{ADS} und r_{OFR} aufgelöst wird. Analog folgt aus Gleichung (3.103)

$$\begin{aligned}
 r_{\text{OET},j} &= \frac{1}{-\nu_{jk}}(-\partial_t w_k) + \frac{1}{-\nu_{jk}} \left(\sum_{n,n \neq j} \nu_{nk} \cdot r_{\text{OET},n} + \sum_l \nu_{lk} \cdot r_{\text{ADS},l} + \sum_m \nu_{mk} \cdot r_{\text{OFR},m} \right) \\
 r_{\text{ADS},l} &= \frac{1}{-\nu_{lk}}(-\partial_t w_k) + \frac{1}{-\nu_{lk}} \left(\sum_j \nu_{jk} \cdot r_{\text{OET},j} + \sum_{n,n \neq l} \nu_{nk} \cdot r_{\text{ADS},n} + \sum_m \nu_{mk} \cdot r_{\text{OFR},m} \right) \\
 r_{\text{OFR},m} &= \frac{1}{-\nu_{mk}}(-\partial_t w_k) + \frac{1}{-\nu_{mk}} \left(\sum_j \nu_{jk} \cdot r_{\text{OET},j} + \sum_l \nu_{lk} \cdot r_{\text{ADS},l} + \sum_{n,n \neq m} \nu_{nk} \cdot r_{\text{OFR},n} \right)
 \end{aligned} \tag{3.105}$$

Die Gleichungen (3.104) und (3.105) stellen rekursive Beziehungen in den jeweiligen Reaktionsraten dar, welche zusammen mit (3.101) zum Aufbau einer allgemeinen Gleichung zur Berechnung des Stroms aus den Gradienten und Zeitableitungen verwendet werden können. Ein Algorithmus, der dieses leistet, ist in der Modellierungskomponente Ecco implementiert (siehe Abschnitt 4.3.6).

3.12. Objekt-orientierte Modellierung

In den vorherigen Abschnitten wurden die theoretischen Grundgleichungen zur mathematischen Modellierung elektrochemischer Systeme betrachtet. Die Berechnung des Modells durch ein Computerprogramm verlangt nach einem Konzept, welches dem Computer die mathematischen Gleichungen in Form einer Programmiersprache zugänglich macht. Die *Objekt-Orientierung* stellt heute eine sehr erfolgreiche Methode zur Abbildung komplexer Zusammenhänge in Computercode dar. In diesem Abschnitt werden daher die mathematischen Modellgleichungen objekt-orientiert behandelt.

Die Grundidee der objekt-orientierten Modellierung besteht darin, ein komplexes Gesamtproblem in kleinere, voneinander möglichst unabhängige Bestandteile aufzuteilen. Die kleinsten Komponenten eines solchen Entwurfes werden Objekte genannt. Sie stellen Softwarecode mit einem klar definierten Aufgabenbereich dar.

Die umrahmten Geschwindigkeitsgesetze (3.81) – (3.95) bilden als die Einträge der Vektoren \mathbf{r}_{hom} , \mathbf{r}_{ET} , \mathbf{r}_{ads} und $\tilde{\mathbf{r}}_{\text{ads}}$ auf der rechten Seite der Gleichungen (3.96) und (3.97) sowie in Gleichung (3.98) die Grundbausteine des elektrochemischen Modells. In der folgenden objekt-orientierten Betrachtung werden sie somit von sogenannten *Law-Objekten* verkörpert. Als ein nächster Schritt des objekt-orientierten Entwurfs werden die gemeinsamen Eigenschaften sowie die Unterschiede der einzelnen Law-Objekte analysiert.

Man erkennt leicht, dass die Funktionen (3.81) – (3.95) explizit von mindestens einer der Variablen w , u oder p abhängen, also im Allgemeinen der Beziehung

$$r = f(v, p) \tag{3.106}$$

gehörten, wobei die dimensionslosen Oberflächen- und Volumenkonzentrationen in den Vektor $v = (w, u)$ zusammengefasst werden. Aus mathematischer Sicht beschreibt Gleichung (3.106) eine reellwertige Funktion von mehreren Veränderlichen $r : \mathbb{R}^{N_s+1} \rightarrow \mathbb{R}$. Im Hinblick auf die numerische Lösung von Differentialgleichungen der Formen (3.96) und (3.97) sind, neben den Funktionswerten, insbesondere die Werte der partiellen Ableitungen von (3.106) nach den Oberflächen- oder Volumenkonzentrationen der einzelnen Spezies

$$\partial_{v_k} r(v, p), \quad k = 0, \dots, N_s - 1 \quad (3.107)$$

von Interesse. Diese werden von numerischen Algorithmen zur Berechnung der Jacobi-Matrix benötigt (siehe z.B. die Verfahren in Abschnitt 6.2).

Für potentialabhängige Funktionen kann eine im Allgemeinen von Null verschiedene Ableitung nach dem Potential

$$\partial_p r(v, p) \quad (3.108)$$

berechnet werden. Für alle anderen Law-Objekte gilt

$$\partial_p r(v, p) = 0 \quad (3.109)$$

Neben den unterschiedlichen funktionalen Zusammenhängen der Reaktionsraten (3.81) – (3.95) fallen zudem zumeist verschiedene Reaktionsparameter in den Ausdrücken auf. Die Reaktionsparameter stellen innerhalb der Software einen weiteren Typ an Objekten dar. Jedes Law-Objekt besitzt somit ein eigenes Reaktionsparameter-Objekt.

Als nächstes wollen wir den Kontext betrachten, in dem die einzelnen Geschwindigkeitsgesetze verwendet werden.

In den Gleichungen (3.96) – (3.98) werden Summen gebildet, deren Summanden aus einem Produkt zwischen Stöchiometriekoeffizient und Geschwindigkeitsgesetz bestehen

$$s_k(v, p) = \sum_i \nu_{ik} \cdot r_i(v, p) \quad (3.110)$$

Ausdruck (3.110) beschreibt wiederum eine reellwertige Funktion in den Variablen v und p , deren partielle Ableitungen sich aus den Ableitungen der einzelnen r -Terme berechnen.

$$\partial s_k(v, p) = \sum_i \nu_{ik} \cdot \partial r_i(v, p) \quad (3.111)$$

Insgesamt muss das objekt-orientierte Design zur Beschreibung der Reaktions- und Randterme einerseits die Komposition der einzelnen Gesetze (3.81) – (3.95) zu Summen der Form (3.110) ermöglichen, andererseits einen effizienten Zugriff auf die Funktionswerte und die Werte der partiellen Ableitungen der Terme anbieten. Ersteres wird auf elegante Weise durch das *Kompositum*-Entwurfsmuster[27] erreicht. Die Anwendung

des Musters sowie die Klassenschnittstelle der Reaktions- und Randterme wird in Abschnitt 4.3.4 besprochen. Dort findet sich auch eine graphische Darstellung des objekt-orientierten Designs in Form eines sogenannten Klassendiagramms.

Das Strom-Potential-Residuum (3.98) beschreibt einen weiteren Bestandteil des Modells

$$h(v, p, \chi) = \chi - \sum_j \nu_{j,e} \cdot r_j(v, p) \quad (3.112)$$

Zusätzlich zu den partiellen Konzentrations- und Potentialableitungen lässt sich die Ableitung nach dem Strom bilden. Es gilt

$$\partial_\chi h = 1 \quad (3.113)$$

Integriert man das Strom-Potential-Residuum in die Modellgleichungen (3.97), wird zur numerischen Berechnung des Stroms die Ableitung (3.113) benötigt (siehe Abschnitt 6.2). Das zugehörige *CurrentPotentialResidual*-Objekt wird in Abschnitt 4.3.5 näher besprochen.

Für die (alternative) Berechnung des Stroms gemäß den Beziehungen (3.101) – (3.105) existiert zudem ein weiteres Objekt, welches eine Funktion

$$\chi = f(\partial_x u, \partial_t w) \quad (3.114)$$

verkörpert (*CurrentResponse*-Objekt, siehe Abschnitt 4.3.6). Als Variablen treten hier die Oberflächengradienten der gelösten Spezies $\partial_x u = (\partial_x u_0, \partial_x u_1, \dots)$ und die Zeitableitungen der adsorbierten Formen $\partial_t w = (\partial_t w_0, \partial_t w_1, \dots)$ auf.

Die Erzeugung der dimensionslosen Law-Objekte (3.81) – (3.95), deren Komposition zu den Reaktionstermen und Randgleichungen gemäß (3.110), (3.112) und (3.114), sowie die Verwaltung von dimensionslosen Parameterobjekten für beliebige elektrochemische Reaktionsmechanismen zählen zu den Aufgaben des Kinetik-Compilers von Ecco (Kapitel 4).

Die Hin- und Rücktransformation zwischen den in den Termen enthaltenen dimensionslosen Parametern und deren dimensionierten Formen wird durch eine Adapter-Bibliothek geleistet (Kapitel 5).

4. Ecco

In Kapitel 3 wurden Modellgleichungen behandelt, die verschiedene elektrochemische Prozesse beschreiben. Die Gleichungen stellen den Ausgangspunkt für numerische Simulationen dar und müssen von einer anwenderfreundlichen Simulationssoftware selbst erzeugt werden. Dazu wurde in der Form einer C++-Klassenbibliothek eine Modellierungskomponente entwickelt und unter dem Namen Ecco (**E**lectro**C**hemical **C**ompiler) implementiert[119].

Eine der Hauptaufgaben des Compilers ist es, ausgehend von einer dem Chemiker vertrauten Eingabesprache, die Reaktions- und Randterme der Modellgleichungen (3.96) – (3.98) zu generieren.

Damit gehört Ecco zu den sogenannten *Kinetic-Compilern* (Synonyme: *Reaktions-Compiler* und *chemische Compiler*), die insbesondere in der chemischen Reaktionskinetik bekannt sind (siehe z.B.[70, 71, 120] und darin enthaltene Referenzen). Compiler zur Erzeugung elektrochemischer Modelle sind dagegen weniger häufig[24–26, 121]. Die bekanntesten Kinetik-Compiler der Elektrochemie sind in den proprietären Softwarepaketen DigiSim[24] und ELSIM[26] enthalten.

Der Compiler des DigiSim-Paketes bietet eine einfache Eingabesprache, welche sich für Elektronentransferprozesse sowie für daran gekoppelte homogene chemische Reaktionen erster und zweiter Ordnung eignet. Die Elektronentransferprozesse können jedoch nur als Einelektronen-Reduktionen formuliert werden. Unter der Voraussetzung, dass die Stöchiometrie¹ des eingegebenen Reaktionsmechanismus erfüllt ist, werden *thermodynamisch überflüssige Reaktionen*[74] (TSRs) erkannt, um konsistente Parameterwerte zu berechnen (siehe Abschnitt 4.3.7). Die Stöchiometrie des Reaktionsmechanismus wird jedoch zuvor nicht nachgeprüft, sodass fehlerhafte Simulationsmodelle formuliert werden können. Homogene Reaktionen müssen als reversible Reaktionen formuliert werden. Irreversible Vorgänge können nur durch die Wahl extremer Parameterwerte für Geschwindigkeits- und Gleichgewichtskonstanten angenähert werden. Darüber hinaus wird bei der Erzeugung der zugehörigen Geschwindigkeitsgesetze angenommen, dass die Reaktionsordnung dem Stöchiometriekoeffizienten des Reaktanden entspricht. Im Allgemeinen ist diese Annahme jedoch unzulässig. So werden bei vielen Mechanismen experimentell gebrochenrationale Reaktionsordnungen beobachtet[75]. Die Gleichsetzung von Stöchiometriekoeffizient und Reaktionsordnung führt dazu, dass bei der Verwendung von DigiSim nur unter erheblichem Aufwand, z.B. durch die Formulierung einer

¹Die Stöchiometrie eines Reaktionsmechanismus bezeichnet hier allein das stöchiometrische Verhältnis von Edukt- und Produktmolekülen, ohne Berücksichtigung ihrer atomaren Zusammensetzung.

Serie gekoppelter Elementarreaktionen, Modelle mit komplexer Kinetik simuliert werden können. Dies erschwert nicht nur die Eingabe des Mechanismus für den Anwender, sondern verursacht zudem größere Simulationsgleichungssysteme und folglich einen erhöhten Rechenaufwand. Adsorptionsprozesse, Reaktionen in einem finiten Diffusionsraum oder höher dimensionierte Zellgeometrien können mit DigiSim nicht modelliert werden. Insgesamt beschränkt sich DigiSim auf die Simulation der cyclischen Voltammetrie. Andere elektroanalytische Verfahren werden nicht unterstützt. Neben weiteren Einschränkungen[120] verhindert der kommerzielle Charakter der DigiSim-Software die Wiederverwendung und die Erweiterung des darin enthaltenen Compilers sowie eine objektive Verifizierung des internen Übersetzungsalgorithmus.

Der Compiler des ELSIM-Paketes wurde in verschiedenen Publikationen ausführlich beschrieben[26, 72, 73, 120]. Er unterstützt eine große Anzahl an elektrochemischen Modellen. Unter anderem können Adsorptionsgleichgewichte an einer Elektrode modelliert werden. Die Gesamtzahl der im Reaktionsmechanismus auftretenden Reaktanden ist aus technischen Gründen auf 20 beschränkt[72]. Die Stöchiometriebilanz des Reaktionsmechanismus (siehe Abschnitt 4.3.7) wird geprüft und TSRs werden erkannt. Die Kinetik homogener Reaktionen wird standardgemäß durch Potenzgesetze (3.12) wiedergegeben. Der Übersetzungsprozess von ELSIM verläuft in zwei getrennten Schritten. Zuerst generiert ein Reaktions-Compiler aus dem gegebenen Reaktionsmechanismus eine Textform der entsprechenden Modellgleichungen. Es werden lesbare ASCII-Dateien erzeugt, die dem Anwender dabei helfen sollen, den zugrunde liegenden mathematischen Formalismus zu verstehen[120]. Unter der Annahme, dass der Anwender eine ausreichende Kenntnis über die elektrochemische Modellierung besitzt, kann die Textform der Modellgleichungen von Hand verändert werden. Dadurch lassen sich auch komplexere Systeme einführen. Ohne das explizite Eingreifen des Anwenders in den Übersetzungsprozess ist dies jedoch nicht möglich.

In einem zweiten Schritt werden die Modellgleichungen durch einen Interpreter an den internen Simulationsalgorithmus angebunden. Die Effizienz der gesamten Simulation hängt somit unter anderem von der Geschwindigkeit des Interpreters ab[26]. Sowohl Reaktions-Compiler als auch der Formel-Interpreter sind in der Programmiersprache C implementiert. Der Quellcode von ELSIM ist nicht zugänglich, sodass auch hier keine objektive Diskussion über das Software-Design und somit über die Erweiterbarkeit des Compilers möglich ist. Aus der Verwendung der Programmiersprache C, die eine objekt-orientierte Strukturierung nicht direkt unterstützt, lässt sich jedoch vermuten, dass der Compiler im Rahmen zukünftiger Entwicklungen nur unter erheblichen Aufwand modifiziert und erweitert werden kann.

Im Vergleich zu DigiSim erscheint die chemische Eingabesprache des ELSIM-Compilers wenig intuitiv[120]. Wie DigiSim beschränkt sich ELSIM auf die Modellierung von elektrochemischen Prozessen an einer einzelnen Elektrode. Reaktionen an mehreren Elektroden oder in höher dimensionierten Zellgeometrien werden in der Eingabesprache nicht berücksichtigt.

```
Reaktion; <Option>;  
Reaktion; <Option>, <Option>, ... ;  
...
```

Abbildung 4.1: Ecco's Eingabesprache: Grundstruktur eines Reaktionsmechanismus. Reaktionen und Optionen werden durch Semikolons abgeschlossen. Einzelne Optionen werden durch Kommas getrennt.

Mit Ecco hingegen soll eine Softwarekomponente geschaffen werden, die eine flexible, möglichst allgemeine und erweiterbare Modellierung elektrochemischer Vorgänge unterstützt. Heterogene Volumen- und Oberflächenelektronentransfers (VET/OET), Adsorptionen sowie homogene Reaktionen und Oberflächenreaktionen sollen in Kombination mit einer unbeschränkten Anzahl an Geschwindigkeitsgesetzen zugänglich sein. Die Eingabesprache des Compilers soll so einfach wie möglich sein und aus chemisch-intuitiven Elementen bestehen. Ecco soll ein allgemeines Konzept anbieten, welches die Modellierung von Reaktionen in Elektrodenanordnungen mit mehreren Elektroden sowie in zwei- oder dreidimensionalen Zellgeometrien unterstützt. Die Stöchiometriebilanz des Reaktionsmechanismus soll geprüft werden. Im Rahmen des TSR-Konzeptes sollen automatisch physikalisch konsistente Parameterwerte berechnet werden können. Dafür muss das Konzept durch Oberflächenreaktionen und Adsorptionssysteme erweitert werden. Die erzeugten Modellgleichungen sollen eine effiziente Auswertung durch ein beliebiges numerisches Simulationsverfahren ermöglichen. Sowohl die Wiederverwendung als auch die Erweiterung von Ecco sollen durch ein klar strukturiertes, objekt-orientiertes Design erleichtert werden.

In den nächsten Abschnitten wird die chemische Eingabesprache des Compilers beschrieben. Danach folgt die Diskussion des objekt-orientierten Designs des Compilers und seines Übersetzungsalgorithmus.

4.1. Eingabesprache

Chemische Reaktionsmechanismen setzen sich üblicherweise aus einer oder mehreren Reaktionen zusammen. Die Reaktionen wiederum werden als Komposition aus Stöchiometriekoeffizienten, chemischen Spezies, Elektronen und Reaktionspfeilen betrachtet. Diese Grundstruktur eines Reaktionsmechanismus wird auch von der Eingabesprache des Compilers angeboten, wobei zusätzliche geometrische und kinetische Informationen einer Reaktion als Optionen hinzugefügt werden können (Abbildung 4.1). Beispiele der

Tabelle 4.1: Ecco's Eingabesprache; chemische Komponenten einer Reaktion; die Namen der Substanzen sind beispielhaft angegeben.

Komponente	chemische Notation	Eingabesymbol
Elektron	e^-	e-
gelöste, neutrale Spezies	A	A
adsorbierte, neutrale Spezies	A_{ads}	A{ads}
gelöstes Ion	A^{1+}	A^(1+)
adsorbiertes Ion	A_{ads}^{1+}	A^(1+){ads}
reversible Reaktion	\rightleftharpoons	<=>
irreversible Reaktion	\rightarrow	-->
Gleichgewichtsreaktion	\rightleftharpoons	=

für die Komponenten einer Reaktion verwendeten Symbole sind in Tabelle 4.1 dargestellt.

Ecco unterscheidet zwischen Elektronen und chemischen Spezies als den Basiskomponenten eines elektrochemischen Reaktionsmechanismus. Während für ein Elektron das Symbol 'e-' reserviert ist, können die Namen der chemischen Spezies nahezu frei gewählt werden. Sie beginnen mit einem Buchstaben, einer Zahl oder mit einem der Zeichen '(' und '['. Anschließend kann der Name durch eine beliebige Kombination aus Buchstaben, Zahlen und den Zeichen ']', ')', '-', '-' vervollständigt werden. Adsorbierte Spezies werden durch den Zusatz '{ads}' gekennzeichnet. Ionenladungen können durch das Symbol '^(' und einer Zahl, gefolgt von '+' oder '-' und der abschließenden Klammer ')' formuliert werden.

Die Kinetik einer Reaktion wird u.a. durch verschiedene Reaktionspfeile bestimmt. So werden reversible Reaktionen, deren Geschwindigkeitskonstanten endliche Werte annehmen, mit '<=>' symbolisiert. Die Pfeile '-->' und '=' klassifizieren irreversible Reaktionen und Reaktionen, welche sich in der Zeitskala der Simulation stets im Gleichgewicht befinden. Zudem lassen sich die kinetischen Modelle mittels geeigneter Optionen nahezu beliebig verändern (siehe unten). Es sei darauf hingewiesen, dass zum Zeitpunkt dieser Arbeit der Compiler die Modelle für reversible und irreversible Reaktionstypen unterstützt. Gleichgewichtsreaktionen können bis jetzt nur indirekt durch reversible Reaktionen angenähert werden, indem für die Simulation die Werte der Geschwindigkeitskonstanten genügend groß gewählt werden.

Durch das Symbol 'e-' werden heterogene Elektronentransferreaktionen (VETs und OETs) erkannt (Abbildung 4.2(a)). Es können Oxidationen und Reduktionen formuliert werden. Durch die Wahl des Stöchiometrikoeffizienten des Elektrons sind Mehr-elektronentransfers zugänglich. Die Kennzeichnung einer Elektrodenoberfläche in der zu

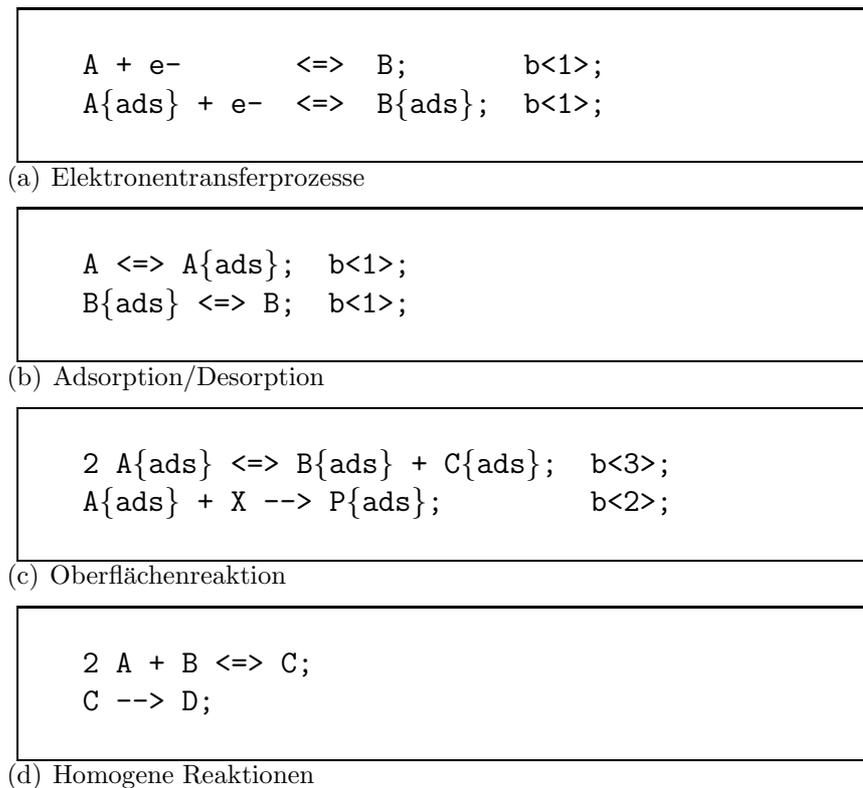


Abbildung 4.2: Ecco's Eingabesprache; Formulierungsbeispiele verschiedener Reaktionstypen.

simulierenden Modellgeometrie erfolgt als Option durch einen sogenannten Randoperator 'b<n>', wobei der Index n den zugehörigen Rand der Modellgeometrie beschreibt (Abbildung 3.2). Auch Adsorptionen, Desorptionen und Oberflächenreaktionen müssen mittels Randoperatoren den Rändern zugeordnet werden (Abbildungen 4.2(b) und (c)). Adsorptionen und Desorptionen werden als unimolekulare Reaktionen formuliert (Abbildung 4.2(b)), während für Oberflächenreaktionen und homogene Reaktionen prinzipiell beliebige Molekularitäten gewählt werden können (Abbildungen 4.2(c) und (d)).²

Ohne die zusätzliche Angabe von kinetischen Optionen erfolgt die Erzeugung der Reaktions- und Randterme der Modellgleichungen nach bestimmten Voreinstellungen. So verlaufen heterogene Elektronentransferreaktionen nach dem Butler-Volmer-Modell, für Adsorptionen und Desorptionen wird das Langmuir-Geschwindigkeitsgesetz angenommen. Die Kinetik homogener sowie von Oberflächenreaktionen folgt Potenzgesetzen, in

²Es sei darauf hingewiesen, dass zwischen einem Stöchiometrikoeffizienten und dem Namen einer Spezies ein Leerzeichen vorhanden sein muss. Andernfalls wird der Koeffizient während des Übersetzungsprozesses in den Namen der Spezies übernommen.

Tabelle 4.2: Ecco's Eingabesprache; kinetische Optionen für Elektronentransfer- (oberer Block), Adsorptions- (mittlerer Block) und kinetische Modelle für homogene und Oberflächenreaktionen (unterer Block); Modell wird von Ecco zurzeit angeboten: Ja (✓), Nein (✗). Standardmodelle sind mit * gekennzeichnet.

Option	Modell	Unterstützung
Butler-Volmer	Butler-Volmer Kinetik*	✓
Nernst	Nernst-Gleichung	✗
Marcus	Marcus-Theorie	✗
Langmuir	Langmuir-Isotherme*	✓
Henry	Henry-Isotherme	✗
Temkin	Temkin-Isotherme	✗
Frumkin	Frumkin-Isotherme	✗
	Potenzgesetz*	✓
p<...> = ...	Potenzoperatoren	✓
r = ...	Generische Gesetze	✓

denen die Reaktionsordnungen den Stöchiometrikoeffizienten der Spezies entsprechen. Um diese Standardeinstellungen des Compilers zu ändern, sind in der Eingabesprache kinetische Optionen vorgesehen.

Zum einen werden Schlüsselwörter zur Verfügung gestellt, die eine Auswahl unterschiedlicher Modelle für heterogene Prozesse ermöglichen sollen. Zum anderen können für Oberflächenreaktionen und homogene Reaktionen entweder die Exponenten der voreingestellten Potenzgesetze verändert oder durch die Formulierung algebraischer Ausdrücke neue Geschwindigkeitsgesetze eingeführt werden.

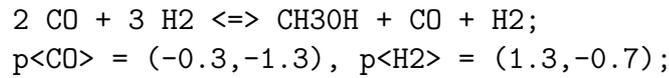
Durch Angabe eines Schlüsselwortes im Optionenabschnitt hinter einer heterogenen Reaktion erzeugt der Compiler die dazugehörige Modellgleichung. Tabelle 4.2 gibt eine Übersicht über die zurzeit verfügbaren kinetischen Optionen und die dazugehörigen Modelle.

Abbildung 4.3 zeigt einige (ausgewählte) Beispiele für die Formulierung von kinetischen Optionen hinter homogenen Reaktionen oder Oberflächenreaktionen.

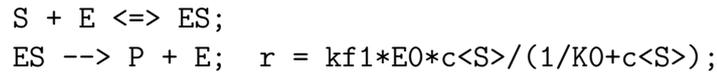
Die voreingestellten Werte der Exponenten α_k und β_k einer Substanzkonzentration in den Potenzgesetzen der Formen (3.81) bzw. (3.91)

$$r = \kappa_f \prod v_i^{\alpha_i} - \kappa_b \prod v_j^{\beta_j}$$

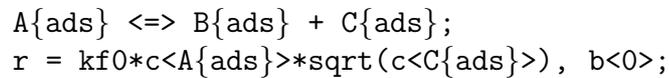
lassen sich mittels sogenannter Potenzoperatoren 'p<...>' überschreiben. Dazu muss der Name der gewünschten Substanz als Argument des Potenzoperators angegeben werden. So werden z.B. durch die Zuweisung



(a) Methanolsynthese[75]: Änderung des Potenzgesetzes



(b) Michaelis-Menten Kinetik: Potenzgesetz und generisches Geschwindigkeitsgesetz



(c) Oberflächenreaktion mit generischem Geschwindigkeitsgesetz

Abbildung 4.3: Ecco's Eingabesprache; Beispiele zur Einführung komplexer Kinetik durch den Ersatz der Exponenten in Potenzgesetzen (a) und durch die Definition generischer Geschwindigkeitsgesetze (b) und (c).

$$p\langle\text{CO}\rangle = (-0.3, -1.3)$$

in Abbildung 4.3(a) die Exponenten der Kohlenmonoxidkonzentration auf $\alpha = -0.3$ und $\beta = -1.3$ gesetzt. Erscheint eine Substanz nur als Edukt (bzw. Produkt) einer Reaktion, so genügt die Angabe eines einzelnen numerischen Werts, z.B.

$$p\langle\text{A}\rangle = -0.5$$

um den entsprechenden Wert für α (bzw. β) festzulegen. Für die Werte der Exponenten sind reelle Zahlen zulässig, solange die Gleichung

$$s = \frac{\beta_k - \alpha_k}{\nu_k} \tag{4.1}$$

für ein $s \in \mathbb{R}^+$ eingehalten wird (siehe Gleichung (3.19)). Diese Bedingung gewährleistet eine konsistente Beziehung zwischen den Exponenten des Potenzgesetzes und dem Stöchiometriekoeffizienten ν_k einer Spezies als Exponent des Massenwirkungsgesetz im Gleichgewichtszustand[72, 75, 77, 122]. Gleichung (4.1) wird von Ecco während des Übersetzungsprozesses geprüft.

Tabelle 4.3: Ecco's Eingabesprache; vordefinierte Komponenten eines generischen Geschwindigkeitsgesetzes.

Komponente	Eingabesymbol	Argument(e)
Konzentration	<code>c<...></code>	Substanzname
Geschwindigkeitskonstanten	<code>kfn</code> , <code>kbn</code>	Reaktionsindex
GG-konstanten	<code>Kn</code>	Reaktionsindex
math. Funktionen	<code>abs()</code> <code>acos()</code> , <code>acosh()</code> <code>asin()</code> , <code>asinh()</code> <code>atan()</code> , <code>atanh()</code> <code>cos()</code> , <code>cosh()</code> <code>exp()</code> , <code>log()</code> <code>sin()</code> , <code>sinh()</code> <code>tan()</code> , <code>tanh()</code> <code>sqrt()</code> , <code>pow(.,.)</code>	math. Ausdruck

Ein allgemeineres Konzept ist durch die Formulierung von generischen Geschwindigkeitsgesetzen der Formen (3.84) bzw. (3.95)

$$\frac{1}{\nu_k} \partial_t v_k = f(v) \quad (4.2)$$

gegeben. Die rechte Seite $f(v)$ in Gleichung (4.2) kann als ein algebraischer Ausdruck formuliert werden (Abbildungen 4.3(b) und (c)). Neben den üblichen mathematischen Rechensymbolen stellt die Eingabesprache für die Komponenten dieser Ausdrücke einige vordefinierte Symbole zur Verfügung (Tabelle 4.3). Substanzkonzentrationen werden dabei durch den Konzentrationsoperator '`c<...>`' symbolisiert, welcher den Namen der Spezies als Argument erhält. Geschwindigkeits- und Gleichgewichtskonstanten einzelner Reaktionen sind mit den Symbolen '`kfn`', '`kb n` ' und '`K n` ' gegeben, wobei n den Index der Reaktion beschreibt. Reaktionen werden von Null beginnend aufsteigend nummeriert. Im Geschwindigkeitsgesetz von Abbildung 4.3(c) bedeutet demnach '`kf1`' die Geschwindigkeitskonstante der Hinreaktion in der zweiten Zeile, während '`K0`' die Gleichgewichtskonstante der ersten Reaktion beschreibt. Der Anwender kann weitere Konstanten durch einen Buchstaben, gefolgt von einer oder mehreren Ziffern, einführen (siehe z.B. '`E0`' in Abbildung 4.3(c)). Es sei darauf hingewiesen, dass der Compiler den physikalisch-chemischen Sinn der formulierten generischen Geschwindigkeitsgesetze zurzeit nicht überprüft, sodass inkonsistente Simulationsmodelle formuliert werden können.

Bei bedachter Anwendung bietet das Konzept jedoch eine äußerst flexible und elegante Methode, komplexe kinetische Zusammenhänge zu modellieren.

4.2. Reaktionsterme und Randgleichungen

Die allgemeine Vorschrift zur Erzeugung der durch einen Reaktionsmechanismus definierten Reaktions- (f_k) und Randterme (g_k) des Modells ist durch³

$$\begin{aligned} f_k &= \sum_i \nu_{ik} \cdot r_i \\ g_k &= \sum_i \nu_{ik} \cdot r_i \end{aligned} \quad (4.3)$$

gegeben (vgl. Gleichung (3.110)). Für jede Spezies (Index k) des Reaktionsmechanismus werden die einzelnen kinetischen Ausdrücke der Reaktionen r_i , an denen die Substanz beteiligt ist, mit dem zugehörigen Stöchiometrikoeffizienten ν_{ik} multipliziert und aufsummiert. Weiterhin gilt an einer elektroaktiven Oberfläche Γ_e das Strom-Potential-Residuum (3.98)

$$h_e = \chi - \sum_{j=0}^{N_{\text{ET}}-1} \nu_{j,e} \cdot r_{\text{ET},j} \quad (4.4)$$

In Gleichung (4.3) steht der Ausdruck f_k für den Kopplungsterm der Volumen- oder Oberflächenkonzentrationen:

$$\partial_t v_k = f_k \quad (4.5)$$

Die rechte Seite einer Neumann-Randbedingung des Modells wird durch g_k beschrieben:

$$-\beta_k \partial_x u_k = g_k \quad (4.6)$$

Eine weitere Randbedingung folgt mit (4.4)

$$0 = h_e \quad (4.7)$$

Die spezielle Form der r -Terme in den Gleichungen (4.3) und (4.4) hängt von dem vom Anwender gewählten Modell für die entsprechende Reaktion ab. Die Funktionssummen f_k , g_k und h_e werden von Ecco automatisch aus dem Reaktionsmechanismus generiert.

³Die Ausdrücke g_k stehen für die Randterme der *gelösten* Spezies (vgl. Gleichung (4.6)) und setzen sich aus den Reaktionsraten der heterogenen Prozesse (VETs, Adsorptionen, heterogene Oberflächenreaktionen) zusammen.

So erzeugt der Compiler für den Mechanismus aus Abbildung 4.2(d) das System

$$\begin{aligned} f_0 &= -2 \cdot r_0 \\ f_1 &= -1 \cdot r_0 \\ f_2 &= 1 \cdot r_0 + (-1) \cdot r_1 \\ f_3 &= 1 \cdot r_1 \end{aligned} \quad (4.8)$$

mit

$$r_0 = \kappa_{f,0} \cdot (u_0)^2 \cdot (u_1)^1 - \kappa_{b,0} \cdot (u_2)^1 \quad (4.9)$$

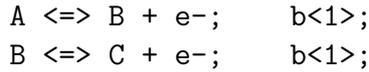
und

$$r_1 = \kappa_{f,1} \cdot (u_2)^1 \quad (4.10)$$

Durch die Angabe von Potenzoperatoren hinter den Reaktionen in Abbildung 4.2(d) können die Konzentrationsexponenten der Potenzgesetze (4.9) und (4.10) verändert werden. Die Formulierung von generischen Geschwindigkeitsgesetzen hingegen hat die Neudefinition der gesamten Ausdrücke r_0 und r_1 zur Folge.

Die Gleichungen (4.8) können als die rechte Seite eines ODE-Systems aufgefasst werden, dessen Lösung den zeitlichen Verlauf der einzelnen Substanzkonzentrationen $u_k(t)$ beschreibt (siehe z.B. Gleichung (3.96)). Zeit- und ortsabhängige Prozesse können modelliert werden, indem man die f_k -Ausdrücke an entsprechende Transportterme addiert (PDEs, siehe z.B. Gleichung (3.97)). Zur Lösung der PDEs müssen u.a. Randbedingungen angegeben werden. Diese werden aus den im Reaktionsmechanismus formulierten heterogenen Elektronentransfers, Adsorptionen und Oberflächenreaktionen erzeugt.

Im Falle zweier gekoppelter Oxidationen im Volumen



generiert Ecco für den Rand Γ_1 das System

$$\begin{aligned} g_0 &= -1 \cdot r_0 \\ g_1 &= 1 \cdot r_0 + (-1) \cdot r_1 \\ g_2 &= 1 \cdot r_1 \end{aligned} \quad (4.11)$$

und

$$h_1 = 1 \cdot r_0 + 1 \cdot r_1 \quad (4.12)$$

mit

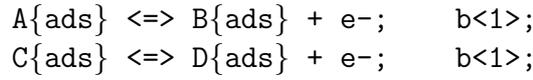
$$r_0 = \psi_{h,0} \cdot \left(e^{\alpha_0 \cdot (p-p_0^0)} \cdot u_0 - e^{-(1-\alpha_0) \cdot (p-p_0^0)} \cdot u_1 \right) \quad (4.13)$$

und

$$r_1 = \psi_{h,1} \cdot \left(e^{\alpha_1 \cdot (p-p_1^0)} \cdot u_1 - e^{-(1-\alpha_1) \cdot (p-p_1^0)} \cdot u_2 \right) \quad (4.14)$$

wobei $\psi_{h,i}$, α_i und p_i^0 die dimensionslosen Parameter (heterogene Geschwindigkeitskonstante, Transferkoeffizient und Formalpotential) der Reaktion mit Index i beschreiben. Die Kopplungsterme für Adsorptionen und Oberflächenelektronentransfers ergeben sich analog.

Für zwei parallele OETs am Rand Γ_1



wird das System

$$\begin{aligned} f_0 &= -1 \cdot r_0 \\ f_1 &= 1 \cdot r_0 \\ f_2 &= -1 \cdot r_1 \\ f_3 &= 1 \cdot r_1 \end{aligned} \quad (4.15)$$

und

$$h_1 = 1 \cdot r_0 + 1 \cdot r_1 \quad (4.16)$$

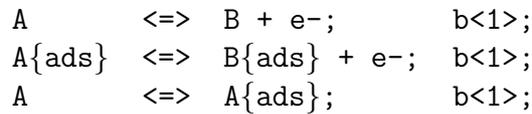
mit

$$r_0 = \psi_{s,0} \cdot \left(e^{\alpha_0 \cdot (p-p_0^0)} \cdot w_0 - e^{-(1-\alpha_0) \cdot (p-p_0^0)} \cdot w_1 \right) \quad (4.17)$$

und

$$r_1 = \psi_{s,1} \cdot \left(e^{\alpha_1 \cdot (p-p_1^0)} \cdot w_2 - e^{-(1-\alpha_1) \cdot (p-p_1^0)} \cdot w_3 \right) \quad (4.18)$$

gebildet. Treten gelöste und adsorbierte Spezies im Reaktionsmechanismus auf, z.B.



ergeben sich die Kopplungsterme

$$\begin{aligned} f_0 &= -1 \cdot r_1 + 1 \cdot r_2 \\ f_1 &= 1 \cdot r_1 \\ f_2 &= 0 \\ f_3 &= 0 \end{aligned} \quad (4.19)$$

Für die Randterme folgt

$$\begin{aligned}g_2 &= -1 \cdot r_0 + (-1) \cdot r_2 \\g_3 &= 1 \cdot r_0\end{aligned}\tag{4.20}$$

und

$$h_1 = 1 \cdot r_0 + 1 \cdot r_1\tag{4.21}$$

mit

$$\begin{aligned}r_0 &= \psi_{h,0} \cdot \left(e^{\alpha_0 \cdot (p-p_0^0)} \cdot v_2 - e^{-(1-\alpha_0) \cdot (p-p_0^0)} \cdot v_3 \right) \\r_1 &= \psi_{s,1} \cdot \left(e^{\alpha_1 \cdot (p-p_1^0)} \cdot v_0 - e^{-(1-\alpha_1) \cdot (p-p_1^0)} \cdot v_1 \right) \\r_2 &= \kappa_{\text{ads},0} \cdot (\varphi - v_0) \cdot v_2 - \kappa_{\text{des},0} \cdot v_0\end{aligned}\tag{4.22}$$

Das objekt-orientierte Design des Compilers bildet die mathematischen Ausdrücke r_i , f_k , g_k und h_e auf natürliche Weise in der Software durch Objekte ab. In den nächsten Abschnitten wird das Design und der Übersetzungsalgorithmus des Compilers näher erläutert.

4.3. Objekt-orientiertes Design und Algorithmus

Die Eingabesprache von Ecco kann als Programmiersprache angesehen werden, welche es ermöglicht, elektrochemische Reaktionsmechanismen und deren kinetische Modelle in für einen Chemiker intuitiver Formulierung darzustellen. Die Symbole und Regeln der Programmiersprache werden in Form einer Grammatik definiert, auf deren Grundlage der Compiler die eingegebenen Mechanismen analysiert und in Computercode übersetzt. Im Allgemeinen besteht ein Compiler aus verschiedenen *Phasen*, welche sich zwei *Teilen* zuordnen lassen[123]. So werden z.B. die *lexikalische Analyse*, die *Syntaxanalyse* und die *semantische Analyse* zum *Analyseteil* gezählt, während die *Zwischencode-Erzeugung* und die *Code-Erzeugung* den *Syntheseteil* bilden. Alle Phasen werden von Fehlererkennung und Datenverwaltung begleitet.

Im Analyseteil werden durch die lexikalische Analyse die Zeichenketten aus dem Eingabetext zu den Symbolen (*Token*) der Sprache zusammengefasst. Danach erzeugt die Syntaxanalyse aus dem generierten Token-Strom eine hierarchische Struktur in Form eines Syntaxbaums (Parserbaum). Syntaktische Fehler werden gemeldet.

Für die Realisierung der lexikalischen Analyse und Syntaxanalyse wurde Spirit eingesetzt[124]. Spirit beschreibt eine objekt-orientierte C++-Template-Bibliothek zur Erzeugung von rekursiv-absteigenden Parsern.⁴ Im Gegensatz zu anderen Parsergeneratoren[125–128] kann Spirit vollständig in fremden Quellcode eingebunden werden und

⁴Ein *Parser* bezeichnet allgemein ein Programm, welches einen Text in eine andere Struktur, z.B. einen Parserbaum übersetzt; *rekursiv-absteigend* beschreibt die spezielle Implementierungsart des Parsers[123].

benötigt keinen externen Übersetzungsschritt. Dies erhöht die Effizienz und erleichtert eine klare Strukturierung des Anwendercodes.

Neben dem Gesamtdesign des Compilers wurde in der vorliegenden Arbeit insbesondere das Design und die Implementierung der semantischen Analyse und des Syntheseteils des Compilers durchgeführt. Während der semantischen Analyse wird mit Hilfe des Parserbaums die Bedeutung des Eingabetextes geprüft. Auch im Syntheseteil des Compilers spielt der Parserbaum eine zentrale Rolle. Im nächsten Abschnitt wird daher zunächst die Grammatik der chemischen Eingabesprache und die Struktur des Parserbaums besprochen.

4.3.1. Grammatik

Eine Grammatik ist üblicherweise aus *nicht-terminalen* und *terminalen Symbolen* (Tokens) aufgebaut. Die hierarchische Struktur der Grammatik wird durch *Syntaxregeln* definiert,

$$\begin{aligned} lhs_1 &= rhs_1 \\ lhs_2 &= rhs_2 \\ &\vdots \end{aligned}$$

wobei jede linke Seite (lhs_i) ein nicht-terminales Symbol beschreibt. Die rechten Seiten (rhs_i) setzen sich im Allgemeinen aus terminalen und nicht-terminalen Symbolen zusammen. Eines der nicht-terminalen Symbole muss als Startsymbol festgelegt werden. Die hier verwendete Terminologie basiert auf der *Extended Backus-Naur Form* (EBNF)[129]. Abbildung 4.4 zeigt einen Auszug aus der Grammatik der Eingabesprache von Ecco.

Das nicht-terminale Symbol `reactionNetwork` bildet das Startsymbol und wird durch ein oder mehrere Reaktionsblöcke (`reactionBlock`) definiert. Ein Reaktionsblock ist durch eine `reaction`, ein Semikolon und bedarfsweise durch `options` gegeben. Die Sequenz aus Syntaxregeln für eine chemische Reaktion ist offensichtlich. Als Option kann ein `boundaryOperator` und/oder ein `law` formuliert werden. Zwischen Randoperator und Gesetz muss ein Komma stehen. Wie Reaktionen werden auch die Optionen durch ein Semikolon abgeschlossen.

Neben den nicht-terminalen Symbolen, die den Typ der Adsorptionen, der heterogenen Elektronentransfers oder der Potenzgesetze festlegen, werden generische Geschwindigkeitsgesetze durch eine etwas kompliziertere Sequenz definiert. Wie in den Abbildungen 4.3(b) und (c) dargestellt, erkennt der Compiler ein generisches Geschwindigkeitsgesetz anhand der terminalen Symbole 'r' gefolgt von '=' und einem algebraischen Ausdruck. Dies ist durch die letzte Syntaxregel in Abbildung 4.4 gegeben. Die vollständige Grammatik der Eingabesprache ist in Form von C++ - Code in Anhang A.3 angegeben.

Im Analyseteil des Compilers wird, basierend auf der Grammatikdefinition, ein dem Eingabemechanismus entsprechender Parserbaum generiert. Abbildung 4.5 zeigt den Ast eines Baums, welcher die erste Reaktion des Mechanismus aus Abbildung 4.2(a) darstellt.

```

reactionNetwork
  = reactionBlock >> *( reactionBlock );
reactionBlock
  = reaction >> ';' >> !(options);
reaction
  = educts >> reactionArrows >> products;
educts
  = reactionAddend >> *( '+' >> reactionAddend );
products
  = reactionAddend >> *( '+' >> reactionAddend );
reactionAddend
  = !(coefficient) >> species;
species
  = electron | adsIon | adsNeutrum | ion | neutrum;
neutrum
  = +( alnum | '(' | '[' ) >> *( alnum | ']' | ')' | '_' | '-' );
options
  = ( boundaryOperator >> !( ',' >> laws ) >> ';' )
    | ( laws >> !( ',' >> boundaryOperator ) >> ';' );
laws
  = adsorptionLaw | electrodeLaw | homogeneousLaw;
homogeneousLaw
  = powerLaw | genericRate;
genericRate
  = 'r' >> '=' >> expression;

```

Abbildung 4.4: Auszug aus der Grammatik der Eingabesprache von Ecco. Das Symbol '>>' bedeutet *gefolgt von*, '*' *kein oder mehrere*, '+' *ein oder mehrere*, '|' *kein oder ein*, und '|' bedeutet *oder*. Terminale Symbole sind durch Anführungszeichen gekennzeichnet. Das `alnum` Symbol steht für einen Buchstaben oder eine Zahl. Die Notation folgt dem EBNF-ähnlichen Formalismus der Spirit-Bibliothek[124].

Jedes nicht-terminale Symbol des Parserbaums ist durch eine eindeutige Identifikationsnummer (ID) gekennzeichnet. Mit Hilfe der IDs wird die semantische Analyse des Eingabemechanismus und die Zwischencode-Erzeugung des Syntheseteils gesteuert.

4.3.2. Der Kinetik-Compiler

Die Diskussion des Kinetik-Compilers wird im Folgenden in drei Blöcke aufgeteilt. Zunächst wird das Gesamtdesign des Compilers und das Design des Übersetzungsalgorithmus besprochen. Danach folgt die Diskussion des Reaktionsnetzwerkes als Ausgabe des Compilers. Die Konsistenzprüfungen werden zum Schluss behandelt.

Für Design und Implementierung wurden insbesondere die Entwurfsmuster *Fassade*,

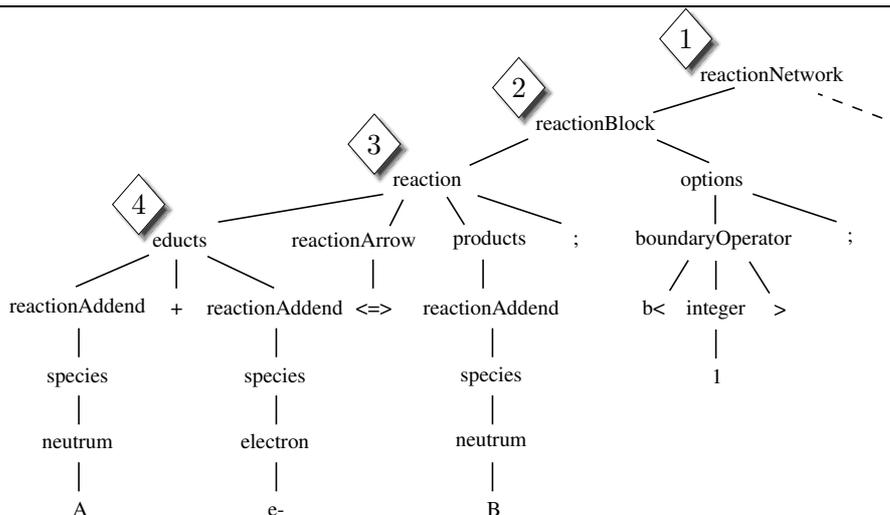


Abbildung 4.5: Auszug aus einem von Ecco erzeugten Parserbaum. Der gezeigte Ast stellt die erste Reaktion des Mechanismus aus Abbildung 4.2(a) dar. Die Identifikationsnummern der ersten vier nicht-terminalen Symbolen sind dargestellt.

Befehl, *Proxy*, *Kompositum*, *Vermittler* und eine generische Abwandlung (siehe Anhang A.2) des *Beobachter*-Musters⁵ eingesetzt[27]. Die für die Klassen-, Objekt- und Interaktionsdiagramme verwendeten Symbole folgen der im Anhang des GoF-Buches[27] beschriebenen OMT-ähnlichen Notation.⁶

Der Kinetik-Compiler ist eine Implementierung des Fassade-Musters (Abbildung 4.6), wobei die *KineticCompiler*-Klasse die Fassade verkörpert. Hinter dieser Fassade verbergen sich eine Vielzahl an interagierenden Klassen (Subsystem). Das Muster schirmt den Anwendercode (Client) vom internen Übersetzungsalgorithmus ab und ermöglicht somit die Weiterentwicklung des Algorithmus, ohne dabei bereits bestehenden Anwendercode verändern zu müssen.

Zur Laufzeit delegiert die *compile()*-Funktion des *KineticCompilers* die gesamte Übersetzungsarbeit an das Subsystem und gibt schließlich ein *ReactionNetwork*-Objekt zurück. Die Deklaration der *compile()*-Methode in der *KineticCompiler*-Klasse lautet

```
ReactionNetwork* compile( const std::string& filename,
                          const std::set<int>& boundaries );
```

Der Eingabeparameter *filename* steht für den Namen der ASCII-Datei, die den Reaktionsmechanismus enthält. Die Randindizes der Modellgeometrie müssen in dem *set*

⁵Im Englischen lauten die Namen der Muster: *Facade*, *Command*, *Proxy*, *Composite*, *Mediator* und *Observer*.

⁶*GoF* ist die Abkürzung für *Gang of Four* und wird in der Literatur oft für die vier Autoren des zitierten Werkes verwendet. *OMT* ist die Abkürzung für *Object Modeling Technique*[130]

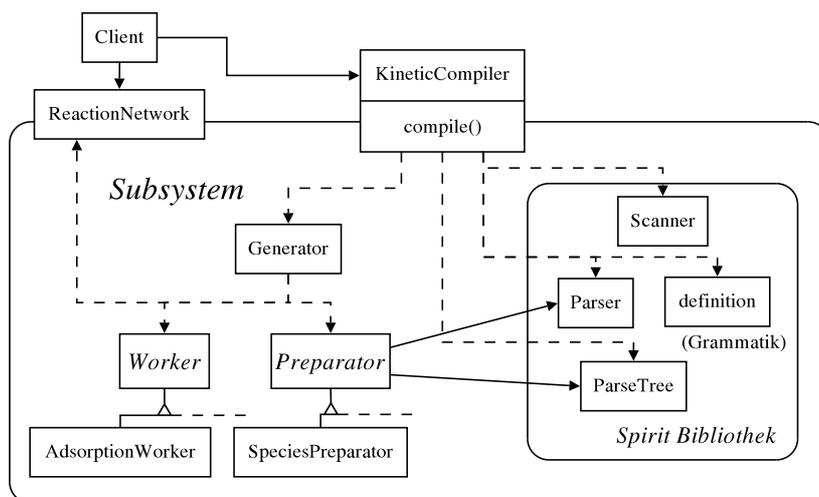


Abbildung 4.6: OO-Gesamtdesign des Kinetik-Compilers (Fassade-Muster). Der äußere Rahmen umfasst alle Klassen, die während der Übersetzung benötigt werden (Subsystem). Der innere Rahmen symbolisiert den Aufgabenbereich der Spirit-Bibliothek (siehe Text).

`boundaries` angegeben werden. Das `ReactionNetwork`-Objekt wird in Abschnitt 4.3.3 genauer besprochen. Für das 1D-Modell aus Abbildung 3.2(b) könnte ein Auszug eines Anwendercodes wie folgt aussehen (Erklärungen der einzelnen Zeilen sind durch Kommentare (`//`-Symbol) gegeben):

```
// Erzeuge ein set für die Randindizes
std::set<int> boundaries;

// Füge die Randindizes in das set ein
boundaries.insert(0);
boundaries.insert(1);

// Erzeuge den Compiler
ecco::KineticCompiler* kineticCompiler =
    new ecco::KineticCompiler();

// Starte die Übersetzung und erhalte Reaktionsnetzwerk
ecco::ReactionNetwork* net =
    kineticCompiler->compile( filename, boundaries );
```

Als erster Schritt der Übersetzung führt der `KineticCompiler` mit Hilfe der `Spirit-Bibliothek` die lexikalische Analyse und die Syntaxanalyse des Eingabemechanismus durch (innerer Rahmen in Abbildung 4.6). Dazu dienen eine `Scanner`- und eine `Parser`-Klasse,

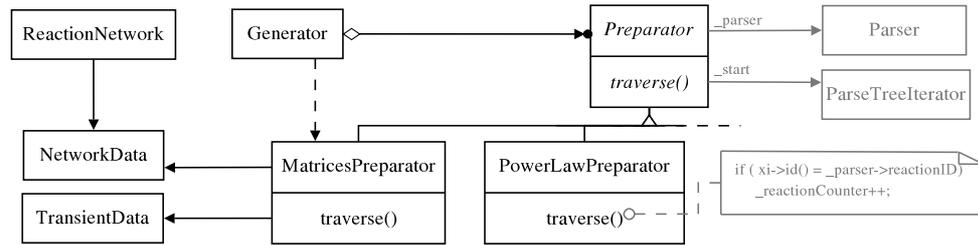


Abbildung 4.7: Design der Zwischencode-Erzeugung (Befehl-Muster). Die Hierarchie zeigt zwei ausgewählte Präparator-Klassen. Ein ähnliches Design wird für die Arbeiterklassen verwendet.

sowie die Grammatik, welche in einer geschachtelten definition-Klasse⁷ innerhalb des Parsers definiert ist (siehe Anhang A.3). In diesem Teil wird der Parserbaum als ein Objekt der ParseTree-Klasse erzeugt und an den Syntheseteil des Compilers übergeben. Im Syntheseteil steuert eine Generator-Klasse die Übersetzung mit Hilfe des Befehl-Musters. Die Zwischencode- und Codeerzeugung werden durch Präparator- (Preparator) und Arbeiterklassen (Worker) als Befehlsklassen realisiert. Abbildung 4.7 zeigt die Implementierung des Musters anhand der Präparator-Klassen.

Die abstrakte Präparator-Oberklasse definiert eine rein virtuelle `traverse()`-Funktion, die von den konkreten Präparator-Unterklassen implementiert wird. In den Unterklassen wird mit Hilfe eines Iterators (`ParseTreeIterator`) der Parserbaum rekursiv durchlaufen und auf das Erscheinen, der durch die Grammatik definierten, nicht-terminalen Symbole überprüft. Diese werden durch ihre Identifikationsnummern erkannt (Abbildung 4.5). So wird z.B. durch den in Abbildung 4.7 gezeigten Pseudocode ein Reaktionszähler inkrementiert, sobald der Iterator (`xi`) auf einen Reaktionsknoten des Parserbaums trifft.

Im Allgemeinen dienen die Präparator-Klassen dazu, Datenstrukturen aufzubauen, die später von den Arbeiterklassen und vom Reaktionsnetzwerk benötigt werden. Die Informationen werden von den Datenklassen `TransientData` und `NetworkData` empfangen (Empfängerklassen des Befehl-Musters). Der Generator führt die `traverse()`-Methoden auf den Befehlsobjekten aus. Er verkörpert somit den Aufrufer des Musters. Eine leichte Abweichung vom Befehl-Muster stellt die Erzeugung der Befehlsobjekte dar. Sie wird nicht von außen durch einen Klienten durchgeführt, sondern durch den Aufrufer selbst. Das Befehl-Muster vereinfacht es, neue Befehle durch weitere Unterklassen hinzuzufügen und somit die Funktionalität des Compilers zu erweitern. Die Implementierung der einzelnen Befehlsklassen kann verändert werden, ohne andere Teile des Algorithmus neu übersetzen zu müssen. Dies führte insbesondere während der Entwicklung des Compilers zu erheblichen Zeitersparnissen.

Eine wichtige Aufgabe fällt während der Zwischencode-Erzeugung einem Objekt der

⁷Eine *geschachtelte* Klasse ist eine Klasse innerhalb einer anderen Klasse (oder Methode)[131].

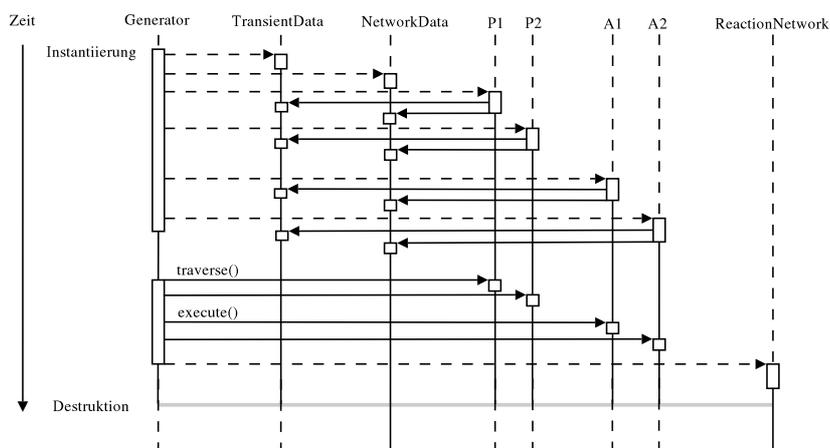


Abbildung 4.8: Interaktionsdiagramm des Syntheseteils, beginnend ab der Instantiierung des Generators. Eine vertikale, durchgezogene Linie beschreibt die Lebensdauer eines instantiierten Objekts. Eine Box symbolisiert eine aktive Operation auf einem Objekt. Ein gestrichelter Pfeil bedeutet die Erzeugung eines neuen Objekts, und durchgezogene Pfeile beschreiben Anfragen an die Objekte. P1, P2, A1 und A2 stellen Beispiele für **P**räparatoren und **A**rbeiter dar.

MatricesPreparator-Klasse zu: Es generiert die in Abschnitt 3.4 beschriebene Stöchiometriematrix S .

Ein zu Abbildung 4.7 analoges Design wird für die Implementierung der Arbeiterklassen verwendet. Die Arbeiter hängen nicht mehr vom Parserbaum ab, sondern generieren anhand der durch die Präparatoren erzeugten Daten eine objekt-orientierte Repräsentation des Reaktionsmechanismus. So werden für chemische Reaktionen Reaktionsobjekte gebildet und mit Reaktionsraten ausgestattet. Letztere werden durch Law-Objekte verkörpert (siehe Abschnitte 4.3.3 und 4.3.4). Darüber hinaus werden für jede Spezies nach der Vorschrift (4.3) die Kopplungs- und Randterme erzeugt. Die Arbeiter bieten dafür eine `execute()`-Funktion an, die der `traverse()`-Methode der Präparatoren entspricht. Abbildung 4.8 zeigt ein Interaktionsdiagramm des Syntheseteils des Compilers.

Zum Zeitpunkt der Instantiierung (Erzeugung) des Generators, wird der genaue Ablauf des Übersetzungsalgorithmus definiert. Dazu werden die Befehlsobjekte erzeugt (gestrichelte Pfeile von links nach rechts) und in einer durch logische Abhängigkeiten zwischen Präparatoren und Arbeitern bestimmten Reihenfolge gespeichert. Gleichzeitig erhalten die Befehlsobjekte Zeiger auf die Datenstrukturen der Datenobjekte (durchgezogene Pfeile von rechts nach links). Anschließend wird der Synthesalgorithmus gestartet und die `traverse()`- und `execute()`-Funktionen der Befehlsobjekte aufgerufen. Am Ende wird ein ReactionNetwork-Objekt erzeugt, welches den Zugriff auf die generierten Netzwerk-

Tabelle 4.4: Die bisherigen Befehlsklassen der Ecco-Bibliothek

AdsorptionLawPreparator, BoundaryPreparator, ElectrodeLawPreparator, FirstPreparator, GenericRatePreparator, LawPreparator, MatricesPreparator, PowerLawPreparator, ReactionPreparator, SpeciesPreparator
AdsorptionWorker, BalanceWorker, BoundaryTermWorker, BoundaryWorker, BulkWorker, DirectorsWorker, EtReactionWorker, KineticTermWorker, LastWorker, ResponseWorker, SurfaceWorker

daten kontrolliert. Beide Objekte bleiben nach der Destruktion des Generators instantiiert. Zum Zeitpunkt dieser Arbeit umfasst der Syntheseteil insgesamt 21 Befehlsobjekte (10 Präparatoren, 11 Arbeiter, siehe Tabelle 4.4).

Obwohl der Algorithmus auf einem umfangreichen Gebrauch der speicherintensiven, assoziativen Container-Klassen *map* und *set* der *Standard Template Library* (STL)[131] beruht, konnte eine effiziente Implementierung auf zwei Wegen erreicht werden: Zum einen werden die (leeren) Datenstrukturen nur *einmal* entweder im TransientData- oder im NetworkData-Objekt erzeugt und während der gesamten Übersetzung nie kopiert. Zum anderen bieten beide Datenobjekte vollen Lese- und Schreibzugriff auf ihre Daten mittels Zeigern. Dem dadurch verbundenen Sicherheitsverlust muss jedoch bei einer zukünftigen Erweiterung des Compilers durch eine besonnene Programmierung Rechnung getragen werden. Die Verwendung der Netzwerkdaten durch Klienten außerhalb des Subsystems wird hingegen durch das ReactionNetwork-Objekt als Stellvertreter (Proxy) abgesichert (siehe Abschnitt 4.3.3).

4.3.3. Das *ReactionNetwork*-Objekt

Das Subsystem des Compilers gibt ein ReactionNetwork-Objekt als Ausgabe zurück (Abbildung 4.6). Das Objekt dient dabei als eine Art Schutzproxy, welches den Schreibzugriff von Klientencode auf die generierten Netzwerkdaten unterbindet. Das Muster ist in Abbildung 4.9 dargestellt.

Das Reaktionsnetzwerk bietet dem Klienten eine zu den Netzwerkdaten äquivalente Schnittstelle an, wobei die Funktionen lediglich eine Kopie der jeweiligen Datenstruktur zurückliefern. Dadurch kann der Klient die Originaldaten des NetworkData-Objekts nicht verändern. Die Schnittstelle der ReactionNetwork-Klasse erfüllt drei grundlegende Anforderungen. Sie ermöglicht den Zugriff

- ⇒ auf die Spezies- und Reaktionsobjekte des Reaktionsmechanismus (siehe z.B. `diss_species()` oder `dyn_hom_reactions()`)

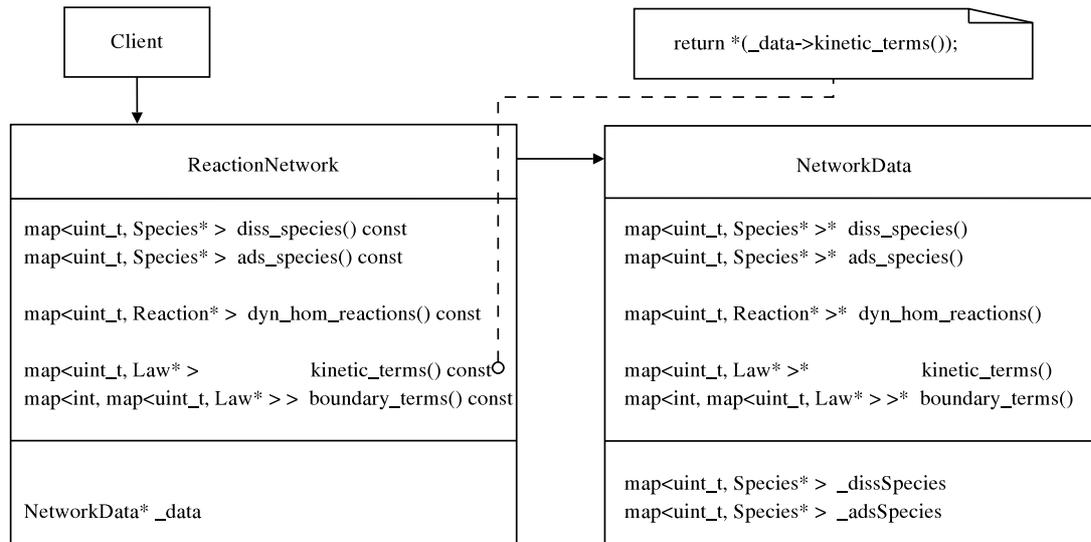


Abbildung 4.9: Das Proxy-Muster in der Implementierung des Reaktionsnetzwerkes.

- ⇒ auf die Datenstrukturen, welche die nach der Vorschrift (4.3) gebildeten Rand- und Kopplungsterme enthalten
(siehe `kinetic_terms()` und `boundary_terms()`)
- ⇒ auf die `CurrentPotentialResidual`- und `CurrentResponse`-Objekte
(siehe Abschnitte 4.3.5 und 4.3.6)

Die meisten Datenstrukturen des Reaktionsnetzwerkes werden durch *maps* verkörpert, welche Schlüssel-Werte-Paare verwalten. Als Schlüssel werden zumeist Spezies-, Reaktions- oder Randindizes verwendet. Die dazugehörigen Werte beschreiben beispielsweise Zeiger auf die verschiedenen Objekte des Reaktionsmechanismus (Reaktionsnetzwerk-Objekte).

Die Klassenstruktur des Reaktionsnetzwerkes ist in Abbildung 4.10 dargestellt. Jede Reaktion besitzt ein Geschwindigkeitsgesetz (*Law*), welches wiederum Reaktionsparameter beinhaltet (*ReactionParameters*). So generiert der Compiler z.B. für eine im Eingabemechanismus vorhandene Adsorption ein Reaktionsobjekt und stattet dieses mit einem Objekt aus, welches eine Langmuir-Reaktionsrate verkörpert (*LangmuirKinetics*). Jedes Gesetzobjekt erzeugt dabei ein eigenes *ReactionParameters*-Objekt, welches von außen durch die Funktion `parameters()` zugänglich ist. Die Werte der jeweiligen Reaktionsparameter, wie z.B. Geschwindigkeitskonstanten oder Formalpotentiale, können durch die `value()` Methoden der Reaktionsparameter-Klassen gelesen oder verändert werden. Um ineffiziente Zugriffe auf die *ReactionParameters*-Objekte während einer Simulation zu vermeiden, wird die Kommunikation zwischen Parameterobjekten auf der einen Seite und *Law*-Objekten auf der anderen Seite durch das Beobachter-Muster gesteuert.

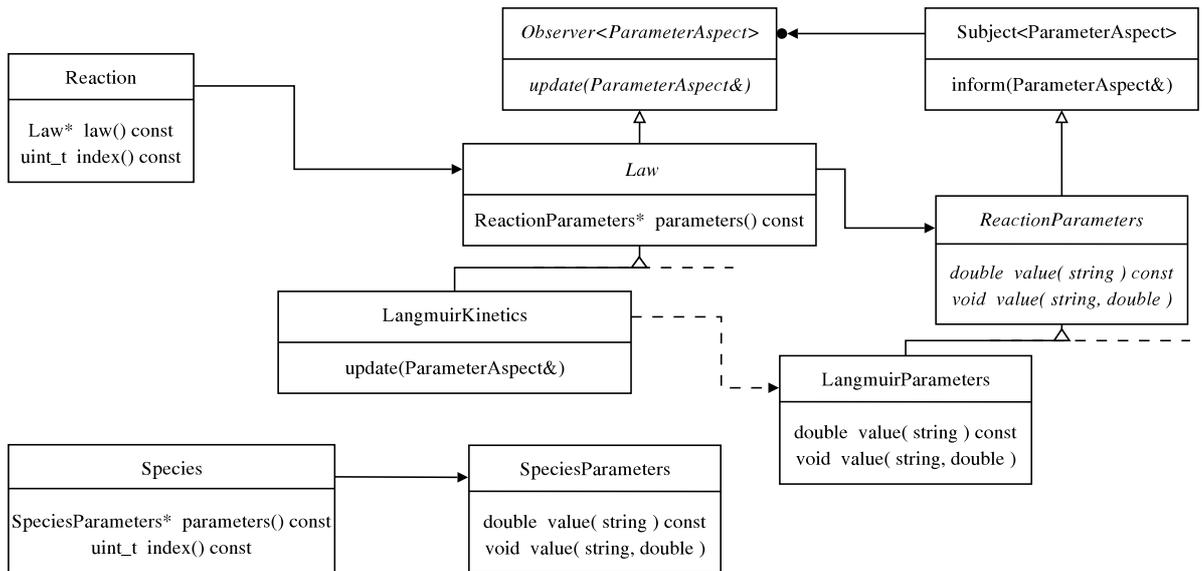


Abbildung 4.10: Komposition der Reaktionsnetzwerk-Klassen. Reaktionen besitzen Gesetze. Gesetze besitzen Reaktionsparameter. Die Kommunikation zwischen Parameter- und Law-Objekten wird über ein Beobachter-Muster gesteuert. Chemische Spezies besitzen Speziesparameter.

Die Subject-Klasse verwaltet dabei Zeiger auf die von den Parametern abhängigen Law-Objekten (Observer). Sobald ein Parameterwert, z.B. in der LangmuirParameter-Klasse, verändert wird, gibt die inform()-Methode der Subjekt-Basisklasse den abhängigen Observern das Signal, ihre internen Daten zu aktualisieren. So verlangt ein LangmuirKinetics-Objekt z.B. nach dem Wert einer Geschwindigkeitskonstanten nur dann, wenn dieser auch verändert wurde. Unnötige Anfragen werden vermieden.

Zur Effizienzsteigerung des Aktualisierungsmechanismus können Parameter-Aspekte verwendet werden, unter denen ein Signal über die Subjektklasse gesendet wird. Dies wird später in Abschnitt 4.3.7 erläutert.

Eine zu den Reaktionen ähnliche, jedoch ungleich einfachere Struktur ist für chemische Spezies gegeben. Diese besitzen SpeciesParameter-Objekte, welche Anfangskonzentrationen und Diffusionskoeffizienten speichern.

4.3.4. Die *Law*-Objekte

Eine für die numerische Simulation wichtige Aufgabe fällt den Law-Klassen zu. Sie werden durch den Compiler nach der Vorschrift (4.3) zu Reaktions- und Randtermen zusammengesetzt. Dies wird durch die Klassenhierarchie aus Abbildung 4.11 ermöglicht, die eine Implementierung des Kompositum-Musters darstellt. Die CompositeLaw-Klasse

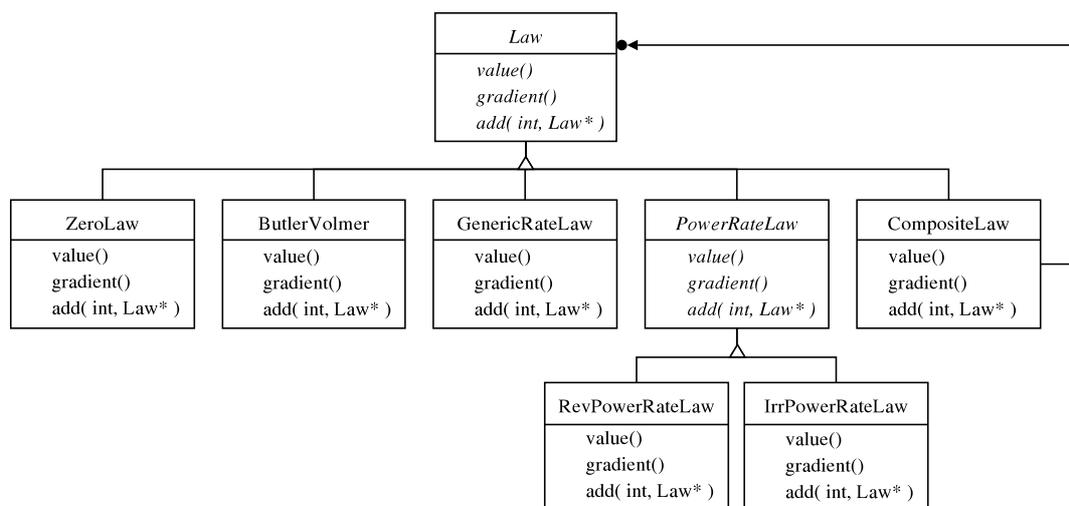


Abbildung 4.11: Kompositum-Muster der Geschwindigkeitsgesetze.

bildet das Kompositum, während alle anderen Law-Klassen die Blattklassen des Musters beschreiben.

Im einfachsten Fall wird der Reaktions- und Randterm einer chemischen Substanz durch ein `ZeroLaw` repräsentiert, welches den Wert Null für alle Zeit- und Raumpunkte zurückgibt. Falls der Konzentrationsverlauf einer Substanz von einer oder mehreren Reaktionsraten abhängt, wird ein `CompositeLaw`-Objekt gebildet. Mit einem solchen Objekt lassen sich gemäß der Vorschrift (4.3) die jeweiligen Reaktionsraten mit einem beliebigen Faktor multiplizieren und aufaddieren. Dies wird von der `add()`-Funktion des Kompositums geleistet. Die vollständige Deklaration der Funktion lautet

```
virtual void add( const int& scalar, Law* law ) = 0;
```

Auch zwei bereits zusammengesetzte Ausdrücke können auf diese Weise miteinander verknüpft werden. Es sei darauf hingewiesen, dass die `add()`-Funktion als rein virtuell deklariert ist. Sie muss somit von jeder Unterklasse überschrieben werden, obwohl eine Implementierung nur in der Kompositum-Klasse sinnvoll ist. Wir folgen hier jedoch der Diskussion in [27] und lassen die `add()`-Funktion einen Fehler melden, sobald sie auf einer der Blattklassen aufgerufen wird.

Unabhängig von ihrer Zusammensetzung erfüllen alle gebildeten Terme die Schnittstelle der Law-Oberklasse und bieten somit den Zugriff auf den Funktionswert, auf die Werte der partiellen Ableitungen und somit auf den Konzentrationsgradienten (siehe dazu Abschnitt 3.12). Im Detail lauten die Deklarationen der entsprechenden Funktionen der Law-Klassenschnittstelle

```
virtual double value( const std::vector<double>& v,
                    const double& p = 0.0 ) = 0;
```

```
virtual void grad( std::vector<double>& grad,  
                  const std::vector<double>& v,  
                  const double& p = 0.0 ) = 0;  
  
virtual double partial( const uint_t& index,  
                        const std::vector<double>& v,  
                        const double& p = 0.0 ) = 0;  
  
virtual double value_dp( const std::vector<double>& v,  
                          const double& p = 0.0 );
```

wobei der Typ `uint_t` eine natürliche Zahl beschreibt. Der Eingabeparameter `index` der `partial()`-Methode steht für den Speziesindex derjenigen Konzentration, für die die partielle Ableitung berechnet werden soll. Der Vektor `v` muss die dimensionslosen Konzentrationswerte aller Spezies unter Berücksichtigung der genauen Indizierung beinhalten. Im Falle potentialabhängiger Reaktionsraten wird die partielle Ableitung nach dem Potential durch die `value_dp()`-Methode berechnet. Potentialunabhängige Terme geben den Wert Null zurück.

Generische Geschwindigkeitsgesetze (Abschnitt 4.1) werden von der `GenericRateLaw`-Klasse dargestellt. Die Implementierung der Klasse basiert auf der symbolischen Algebra-Bibliothek `GiNaC`[132, 133]. Die zur Simulation benötigten Funktions- und Ableitungswerte der generischen Gesetze können somit mit Hilfe des Auswertungsmechanismus der `GiNaC`-Bibliothek berechnet werden. Die symbolische Differentiation der Terme wird dabei bereits während des Übersetzungsschrittes geleistet. Die Effizienz der Simulation bleibt davon folglich unbeeinflusst.

Die Bildung und insbesondere eine effiziente Auswertung der partiellen Ableitungen, welche die Einträge der Jacobimatrix darstellen (vgl. Abschnitt 6.2.1), sind entscheidende Faktoren in der Implementierung eines allgemeinen und schnellen Simulationsalgorithmus. Die numerische Berechnung eines Ableitungswerts mittels einer Finite-Differenzen-Approximation führt im einfachsten Fall zu einer zusätzlichen Funktionsauswertung[134]. Zudem treten dabei stets Diskretisierungsfehler auf. Folglich ist eine symbolische Auswertung, welche die exakten Ableitungswerte liefert, im Zusammenhang *a priori* unbekannter generischer Reaktionsraten vorzuziehen.

Die partiellen Ableitungen der anderen Ausdrücke, wie z.B. irreversibler oder reversibler Potenzgesetze, sind zur Übersetzungszeit von Ecco bereits bekannt und in den jeweiligen Unterklassen implementiert. Folglich kann mit Ecco stets eine exakte Jacobimatrix berechnet werden. Im Falle nicht generischer Geschwindigkeitsgesetze erfolgt die Berechnung von Funktions- und Ableitungswerten mit relativ geringem Aufwand.

4.3.5. Das *CurrentPotentialResidual*-Objekt

Ecco liefert zur Berechnung der Observablen eines elektrochemischen Experiments (siehe Abschnitt 3.11) für jeden Rand der Modellgeometrie ein *CurrentPotentialResidual*-Objekt. Von außen bietet erneut das *ReactionNetwork*-Objekt (Abschnitt 4.3.3) durch die Methode

```
std::map<int,CurrentPotentialResidual* >
    current_potential_residuals() const;
```

den Zugriff auf die entsprechende Datenstruktur. Wie schon bei den Objekten des Reaktionsnetzwerkes aus Abschnitt 4.3.3, werden die *CurrentPotentialResidual*-Objekte mittels einer STL-map[131] verwaltet, wobei der Schlüssel einen Randindex beschreibt. Die Schnittstelle des *CurrentPotentialResidual*-Objekts bietet die Methoden

```
double value( const std::vector<double>& concs,
              const double& p,
              const double& xi ) const;

double partial( const uint_t& s,
               const std::vector<double>& concs,
               const double& p );

double value_dp( const std::vector<double>& concs,
                const double& p );

double value_dxi();
```

an. Wie schon bei der *Law*-Klassenschnittstelle dienen die ersten drei Methoden der Auswertung des Funktionswerts (*value*) und der partiellen Ableitungen bezüglich der einzelnen Konzentrationen (*partial*) bzw. des Potentials (*value_dp*). Zusätzlich existiert die *value_dxi*()-Funktion zur Berechnung der partiellen Ableitung bezüglich des Stroms.

4.3.6. Das *CurrentResponse*-Objekt

Als Alternative zur Verwendung des Strom-Potential-Residuums lässt sich der Strom im Falle potentialkontrollierter Experimente aus dem Massenfluss der Teilchen an der Elektrodenoberfläche berechnen (Abschnitt 3.11). Diese Aufgabe übernimmt ein *CurrentResponse*-Objekt.

Das Objekt repräsentiert eine Funktion der Form

$$\chi = \sum \nu_e \cdot S(\partial_x u, \partial_t w) \quad (4.23)$$

wobei ν_e den Stöchiometrikoeffizienten einer ET-Reaktion beschreibt (siehe Gleichung (3.101)). Der Ausdruck *S* steht für eine Summe, die sich aus den Massenflüssen gelöster

Teilchen $\beta \cdot \partial_x u$ und der zeitlichen Ableitungen adsorbierter Formen $\partial_t w$ bildet.

$$S(\partial_x u, \partial_t w) = \sum_i c_i \cdot \beta_i \cdot \partial_x u_i + \sum_j c_j \cdot \partial_t w_j \quad (4.24)$$

Die Koeffizienten c_i bzw. c_j setzen sich gemäß (3.104) und (3.105) rekursiv aus den Quotienten der Stöchiometrikoeffizienten der beteiligten Spezies zusammen.

Welche Substanzen an der Summenbildung beteiligt sind, bestimmt Ecco aus dem elektrochemischen Reaktionsmechanismus mit Hilfe eines rekursiven Algorithmus.

Um die Summe möglichst einfach zu halten, wird jeder Spezies des Mechanismus für jeden Rand zunächst ein sogenannter *surface index* (SI) zugeordnet. Der SI einer Spezies steht für die Anzahl der Elektronentransfers, Adsorptionen und Oberflächenreaktionen eines Randes, an denen die Spezies beteiligt ist. Je kleiner der SI der Spezies, desto kürzer fallen die Summen der Formeln (3.104) und (3.105) aus. Dies kann insgesamt zu einer Vereinfachung der Beziehung (4.24) führen.

Abbildung 4.12 zeigt ein Flussdiagramm des Algorithmus. Die Grundidee besteht darin, ausgehend von Gleichung (3.101) und mithilfe der Beziehungen (3.104) und (3.105), Datenstrukturen aufzubauen, die später eine einfache Berechnung des Stroms nach (4.23) ermöglichen. Zu den relevanten Daten gehören neben den Stöchiometrikoeffizienten des Elektrons (ν_e) insbesondere die Indizes der Spezies (Menge \mathcal{K}), deren Massenfluss oder Zeitableitungen in den Summen $S(\partial_x u, \partial_t w)$ erscheinen sowie die dazugehörigen Koeffizienten c_i bzw. c_j (Menge $\mathcal{C}_{\mathcal{K}}$).

Im Allgemeinen steht für jeden ET (VET und OET) eines Mechanismus eine Summe $S(\partial_x u, \partial_t w)$. Um die Index-Menge \mathcal{K} einer solchen Summe zu ermitteln, wird rekursiv nach allen Spezies gesucht, die mit einer der elektroaktiven Substanzen des ETs über weitere Reaktionen eines Randes direkt oder indirekt in Verbindung stehen.

Zunächst müssen dafür Adsorptionssysteme der Form



gesondert betrachtet werden. Für den Strom dieser Systeme gilt nach Gleichung (3.101)

$$\chi = -r_{\text{VET},0} - r_{\text{OET},1} \quad (4.26)$$

Mit den Beziehungen

$$r_{\text{VET},0} = f_2 - r_{\text{ADS},2} \quad (4.27)$$

und

$$r_{\text{ADS},2} = \partial_t w_0 + r_{\text{OET},1} \quad (4.28)$$

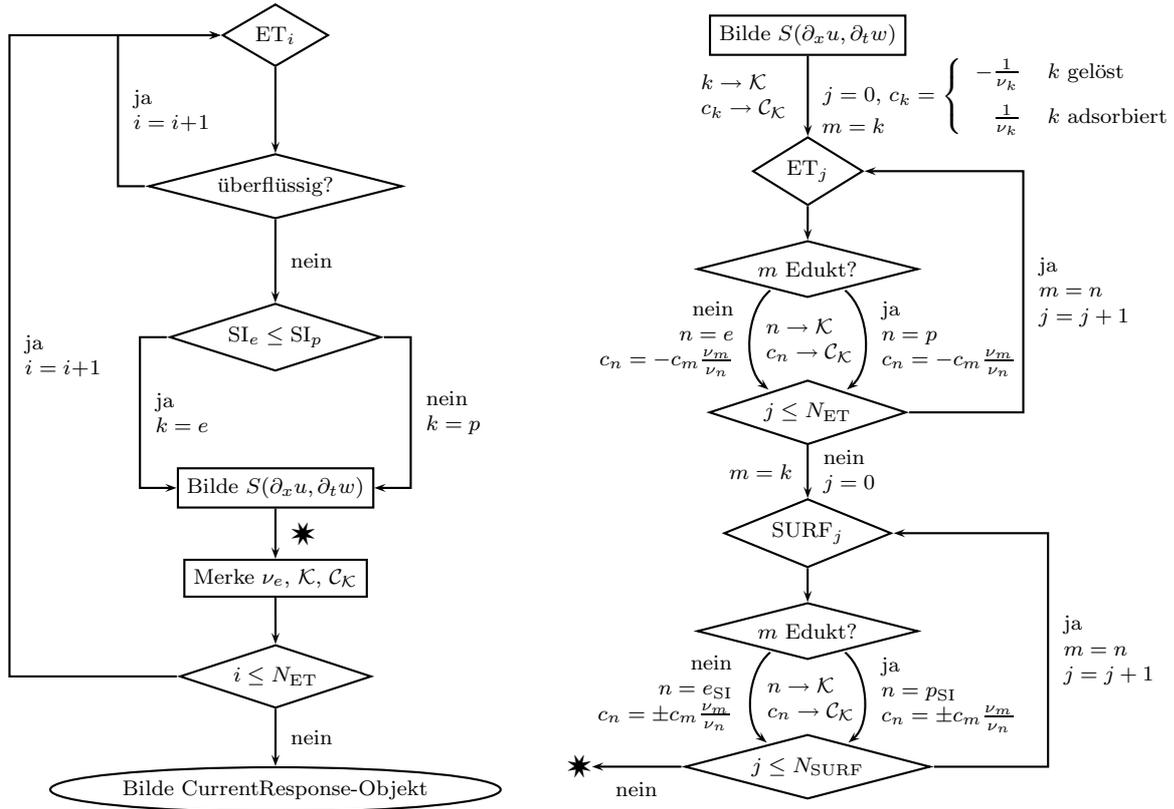


Abbildung 4.12: Flussdiagramm des Algorithmus zur Erzeugung der rechten Seite von Gleichung (4.23). Die Notation e_{SI} (p_{SI}) deutet auf eine Entscheidung zwischen den Indizes von Edukten (Produkten) aufgrund eines kleineren *surface index* hin. Erklärung siehe Text.

(vgl. Gleichungen (3.104) und (3.105)) folgt aus Gleichung (4.26)

$$\chi = -f_2 + \partial_t w_0 + r_{OET,1} - r_{OET,1} = -f_2 + \partial_t w_0 \quad (4.29)$$

Man erkennt in Gleichung (4.29), dass sich der Term $r_{OET,1}$ wegekürzt. Der OET darf somit bei der Bildung der Summen nicht berücksichtigt werden. Er wird als “überflüssig” markiert (Abbildung 4.12, linke Seite).⁸ Der Algorithmus vergleicht im nächsten Schritt die SIs der Redoxformen eines ETs und wählt entsprechend entweder den Index des Edukts ($k = e$) oder des Produkts ($k = p$) zur weiteren Analyse aus.

Für die Summe des ETs werden auf der Stufe “Bilde $S(\partial_x u, \partial_t w)$ ” die Datenstrukturen \mathcal{K} und $\mathcal{C}_{\mathcal{K}}$ erzeugt. Zusammen mit den Stöchiometriekoeffizienten der beteiligten Elektroden werden beide Datenstrukturen dem CurrentResponse-Objekt zum Zeitpunkt seiner

⁸Das Formalpotential des OETs in Systemen der Form (4.25) wird durch das Formalpotential des VETs und die Lage der Adsorptionsgleichgewichte bestimmt. Der OET stellt somit auch im Sinne von [74] eine “thermodynamisch überflüssige” Reaktion dar (siehe Abschnitt 4.3.7).

Instantiierung übergeben.

Auf der rechten Seite von Abbildung 4.12 ist die Erzeugung von \mathcal{K} und $\mathcal{C}_{\mathcal{K}}$ im Detail dargestellt. Zunächst wird der Koeffizient der elektroaktiven Substanz c_k berechnet und zusammen mit dem Index k in die Datenstrukturen eingetragen: $k \rightarrow \mathcal{K}$ bzw. $c_k \rightarrow \mathcal{C}_{\mathcal{K}}$. Danach werden die weiteren ETs und anschließend Adsorptionen und Oberflächenreaktionen (SURF) betrachtet, an denen die Spezies mit Index m beteiligt sind.

Die Durchführung des Algorithmus sowie die Bildung der CurrentPotentialResidual- und CurrentResponse-Objekte werden von einem sogenannten ResponseWorker-Objekt übernommen.

Wie schon in Abschnitt 4.3.5, erfolgt der Zugriff auf das CurrentResponse-Objekt über das ReactionNetwork-Objekt durch die Methode

```
std::map<int,CurrentResponse* > current_responses() const;
```

Im Vergleich zur Erzeugung ist die Verwendung des Objekts nahezu trivial. Es bietet die Funktion

```
double value( const std::vector<double>& du ) const;
```

an. Diese erwartet einen Vektor 'du' als Variable und gibt den mittels Gleichung (4.23) berechneten Strom zurück. Die Einträge des Vektors sind nach den Speziesindizes geordnet und müssen die Werte der Zeitableitungen (für adsorbierte Spezies) bzw. der Oberflächengradienten (für gelöste Spezies) enthalten.

4.3.7. Konsistenzprüfung

Neben der Erzeugung der in den vorherigen Abschnitten beschriebenen Reaktionsnetzwerkobjekte analysiert der Compiler die Semantik des eingegebenen Reaktionsmechanismus und erkennt physikalische Abhängigkeiten zwischen Parameterwerten.

Die Analyse beinhaltet die Überprüfung der Reaktionsstöchiometrie, der Ladungsbilanz und der Abhängigkeiten zwischen Parameterwerten im Zusammenhang von Zellgeometrien mit mehreren Arbeitselektroden, generischen Geschwindigkeitsgesetzen sowie TSRs (siehe Seite 41).

Sobald die Stöchiometriematrix S gemäß Gleichung (3.9) gebildet wurde, führt der Compiler eine Überprüfung der Reaktionsstöchiometrie durch (Massenbilanz)[70, 72]. Mathematisch muss dafür das System

$$S \cdot m = 0 \tag{4.30}$$

erfüllt sein, wobei alle Einträge des Vektors $m \in \mathbb{R}^{N_s \times 1}$ notwendigerweise positive Massen der Reaktanden beschreiben ($m_k > 0, \forall k$). Um das Problem numerisch zu behandeln, wird es auf ein lineares Optimierungsproblem zurückgeführt.

Mit Gleichung (4.30) existiert ein $\lambda > 0$, sodass für ein $y = \lambda \cdot m, y \geq 1$ die Gleichung

$$S \cdot y = 0 \tag{4.31}$$

gilt. Sei $x = y - \mathbf{1}$ und $z = -\sum_k x_k + 1$, dann ist die Stöchiometrie eines Reaktionsmechanismus erfüllt, falls für die lineare Optimierungsaufgabe

$$\left\{ \begin{array}{l} \text{maximiere } z = -\sum_k x_k + 1 \\ \text{mit } S \cdot x = b, \quad x \geq 0 \\ \text{wobei } b = -S \cdot \mathbf{1}, \quad \mathbf{1} = (1, 1, \dots, 1)^T \end{array} \right. \quad (4.32)$$

eine gültige Lösung existiert[70, 72]. Ecco löst Aufgabe (4.32) durch ein BalanceWorker-Objekt, welches eine Implementierung des Simplex-Algorithmus kontrolliert[134]. Optional kann die Ladungsbilanz

$$\sum_k \nu_{ik} z_{ik} = 0, \quad i = 0, \dots, N_r - 1 \quad (4.33)$$

der formulierten Reaktionen überprüft werden, wobei z die Ionenladung einer Spezies beschreibt.

Bei der Modellierung elektrochemischer Reaktionsmechanismen können physikalische Abhängigkeiten zwischen einzelnen Reaktionsparametern entstehen. Um konsistente Parameterwerte bei der Eingabe durch den Anwender zu gewährleisten, muss der Compiler die Abhängigkeiten der Parameter anhand des Reaktionsmechanismus erkennen und gegebenenfalls konsistente Werte automatisch berechnen.

So werden die Formalpotentiale eines VETs im Allgemeinen unabhängig vom Material der Elektrode angenommen. Die Kinetik des Elektronentransfers hingegen kann durch die Beschaffenheit der Elektrodenoberfläche beeinflusst werden[51]. Findet also ein Elektronentransfer einer gelösten Substanz an unterschiedlichen Elektroden in einer Modellgeometrie statt, muss der Compiler für jede Elektrodenreaktion Geschwindigkeitsgesetze mit eigenen Reaktionsparametern erzeugen und gleichzeitig garantieren, dass die Werte der Formalpotentiale in den Ausdrücken übereinstimmen. Typische Beispiele für Systeme, in denen Elektronentransfers an unterschiedlichen Elektroden stattfinden, sind Ring-Scheiben-Elektroden[135] sowie Dünnschichtzellen[51].

Falls generische Reaktionsraten Reaktionsparameter enthalten, die zu anderen Geschwindigkeitsgesetzen gehören, entsteht eine weitere Parameterabhängigkeit. Eine solche Situation ist in dem Mechanismus aus Abbildung 4.3(b) dargestellt. Hier muss der Wert der Gleichgewichtskonstante des Potenzgesetzes der ersten Reaktion mit dem Wert K_0 der generischen Reaktionsrate übereinstimmen.

Im Falle von TSRs wird der Zusammenhang zwischen Gleichgewichtskonstanten von homogenen Reaktionen und Elektronentransfers durch

$$K_i^{\omega_i} = \prod_{j=0, j \neq r}^{N_{\text{tsr}}-1} K_j^{\omega_j} \quad (4.34)$$

gegeben, wobei N_{tsr} die Anzahl der *abhängigen Reaktionen* beschreibt[72, 74, 136]. Die K -Werte der Elektronentransfers sind über

$$K = \exp\left[-\frac{\nu_e F}{RT} E^0\right] \quad (4.35)$$

mit den Formalpotentialen verbunden (siehe Diskussion zu Gleichung (24) in[72]). Für eine abhängige Reaktion kann der Stöchiometrievektor

$$s_i = (\nu_{ik})_k \quad (4.36)$$

(s_i entspricht hier, bis auf einen möglichen Indizierungswechsel, einer Zeile der Stöchiometriematrix S in Gleichung (3.9)) als eine Linearkombination aus anderen Stöchiometrievektoren s_j

$$s_i = \sum_{j=0, j \neq r}^{N_{\text{tsr}}-1} \omega_j \cdot s_j \quad (4.37)$$

mit den Koeffizienten ω_j dargestellt werden. Um linear abhängige Zeilen von S zu erkennen, wurde im Gegensatz zu früheren Ansätzen (Minimierung der Fehlerquadrate[74], Gauß-Jordan-Eliminierung[72]) in dieser Arbeit eine numerische Rangentscheidung auf der Grundlage der QR -Zerlegung[134] entworfen. Darüber hinaus wurde das TSR-Konzept auf Oberflächenreaktionen erweitert.

In einem ersten Schritt wird dazu aus der Stöchiometriematrix eine reduzierte Matrix \tilde{S} gebildet, indem zunächst die Spalten aller Elektronen entfernt werden. Dadurch können VETs einer Spezies, die an unterschiedlichen Elektrodenoberflächen eintreten, identische Stöchiometrievektoren aufweisen.⁹ In einem solchen Fall wird in \tilde{S} nur *einer* dieser Vektoren beibehalten.

Danach werden die Zeilen von \tilde{S} nacheinander in eine Testmatrix A gefüllt. Falls A^T mindestens zwei Spalten enthält, wird der Rang von A^T (rank_A) durch die QR -Zerlegung von A^T bestimmt. Abhängige Reaktionen besitzen linear abhängige Stöchiometrievektoren. Sie werden somit genau dann erkannt, falls ein rank_A -Wert auftritt, der kleiner ist als die Anzahl der Spalten von A^T .

Für die Berechnung der ω_i -Koeffizienten in Gleichung (4.37) wird der letzte Spaltenvektor aus A^T entfernt und in einen Vektor b gespeichert. Die so erhaltene Matrix \tilde{A}^T besitzt vollen Rang, und das überbestimmte lineare Gleichungssystem

$$\tilde{A}^T \cdot \omega = b \quad (4.38)$$

kann mit Hilfe des linearen Ausgleichsproblems

$$\|\tilde{A}^T \cdot \omega - b\|_2 \rightarrow \min! \quad (4.39)$$

⁹Ecco unterscheidet zwischen Elektronen, die an verschiedenen Rändern übertragen werden (siehe Abschnitt 3.4).

eindeutig gelöst werden. Dazu wird eine QR -Zerlegung der Matrix $\tilde{A}^T = Q \cdot R$ durchgeführt. Aus Gleichung (4.38) folgt

$$\begin{aligned} Q \cdot R \cdot \omega &= b \\ \Leftrightarrow Q^T Q \cdot R \cdot \omega &= Q^T \cdot b \\ \Leftrightarrow R \cdot \omega &= z \end{aligned} \tag{4.40}$$

wobei $z = Q^T \cdot b$ und, aufgrund der Orthogonalität der Matrix Q , die Identität $Q^T Q = I$ gilt. Matrix R besitzt obere Dreiecksform, wodurch sich mit $\|x\|_2 = (x^T x)^{1/2}$ und der Dreiecksungleichung $\|x+y\|_2 \leq \|x\|_2 + \|y\|_2$ das zu Ausgleichsproblem (4.39) äquivalente Problem

$$\begin{aligned} \left\| \begin{pmatrix} \tilde{R} \\ 0 \end{pmatrix} \cdot \omega + \begin{pmatrix} -\tilde{z} \\ -\hat{z} \end{pmatrix} \right\|_2 &= \|\tilde{R} \cdot \omega - \tilde{z} + (-\hat{z})\|_2 \\ &\leq \|\tilde{R} \cdot \omega - \tilde{z}\|_2 + \|-\hat{z}\|_2 \rightarrow \min! \end{aligned} \tag{4.41}$$

ergibt. In (4.41) beschreiben \tilde{R} den oberen Dreiecksblock von R und \tilde{z} die oberen sowie \hat{z} die unteren Einträge des Vektors z . Weiterhin gilt $\|x\|_2 \geq 0$, sodass die rechte Seite von (4.41) genau dann minimal wird, wenn

$$\tilde{R} \cdot \omega = \tilde{z} \tag{4.42}$$

Die Lösung des Ausgleichsproblems (4.39) ist somit auf die Lösung des linearen Gleichungssystems (4.42) zurückgeführt, welche sich, aufgrund der Dreiecksgestalt von \tilde{R} , ohne großen zusätzlichen Aufwand berechnen lässt.

Zusätzliche TSR-ähnliche Parameterabhängigkeiten entstehen in Adsorptionssystemen der Form (4.25). Für solche Systeme und im Falle der in dieser Arbeit verwendeten Langmuir-Gesetze, ergibt sich das Formalpotential des OETs gemäß[78]

$$E_1^0 = E_0^0 + \frac{RT}{\nu_e F} \ln \frac{K_2}{K_3} \tag{4.43}$$

wobei für die Stöchiometrikoeffizienten des Elektrons $\nu_e = \nu_{e,1} = \nu_{e,2}$ gilt. Ecco erkennt diese Zusammenhänge und unterstützt die automatische Berechnung des E_1^0 -Wertes.

Im Allgemeinen entstehen durch die oben beschriebenen Parameterbeziehungen Abhängigkeiten zwischen mehreren kinetischen sowie thermodynamischen Parameterwerten ($n:m$ -Objektabhängigkeit). Darüber hinaus können indirekte Wechselbeziehungen zwischen unterschiedlichen abhängigen Parametergruppen entstehen, falls die beschriebenen Fälle gleichzeitig in einem Modell auftreten. Sobald ein Parameterwert verändert wurde, müssen alle direkt sowie indirekt abhängigen Werte synchronisiert werden.

Ein objekt-orientiertes Design, welches sich für einen solch komplexen Aktualisierungsmechanismus eignet, ist ein Zusammenspiel zwischen Vermittler- und Beobachter-Muster[27]. Die Einführung eines vermittelnden Objekts reduziert die $n:m$ - zu einer $1:m$ -Abhängigkeit, die durch das Beobachter-Muster bewältigt werden kann. Abbildung 4.13

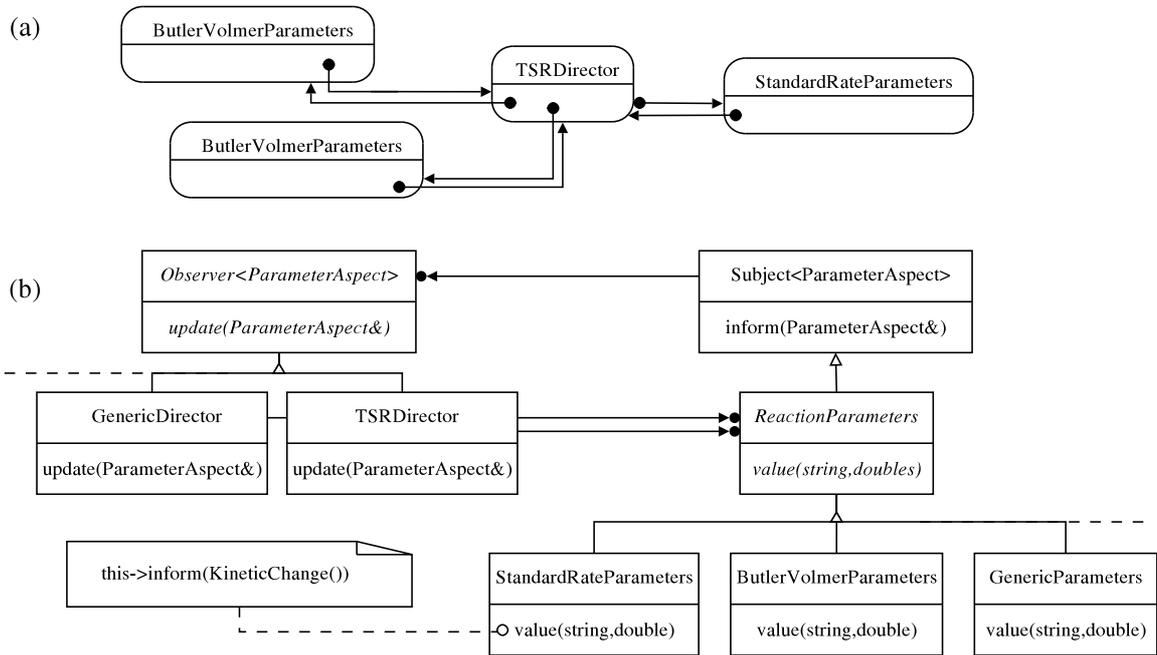


Abbildung 4.13: (a) Beispiel einer Objektstruktur im Falle einer TSR (siehe Mechanismus (4.44)) und (b) die dazugehörige Klassenhierarchie, welche das Vermittler- und Beobachter-Muster verbindet. Template-Klassen sind mit $\langle \dots \rangle$ gekennzeichnet.

(a) zeigt eine typische Objektstruktur, die z.B. durch die Formulierung eines EE_{disp} -Mechanismus



entsteht. In Mechanismus (4.44) sind die (reduzierten) Stöchiometrievektoren der Reaktionen voneinander abhängig, wodurch die Parameter über die Gleichungen (4.34) und (4.35) gemäß

$$K_2^1 = K_0^{-1} \cdot K_1^1 = \exp\left[\frac{F}{RT}(E_0^0 - E_1^0)\right] \tag{4.45}$$

in Verbindung stehen. Beide Elektronentransfers besitzen z.B. Butler-Volmer-Reaktionsraten mit `ButlerVolmerParameters`-Objekten. Die Kinetik der Disproportionierung folgt einem Potenzgesetz mit Reaktionsparametern, welche sowohl die Geschwindigkeitskonstanten als auch die Gleichgewichtskonstante K_2 beschreiben (`StandardRateParameters`). In Abbildung 4.13(a) agiert ein `TSRDirector`-Objekt als Vermittler. Die dazugehörige Klassenhierarchie ist in Abbildung 4.13(b) gegeben.

Die Kommunikation zwischen den Direktoren (Vermittler, Beobachter) auf der einen und Parameterklassen (Subjekte) auf der anderen Seite wird über zwei Basisklassen gesteuert, die eine generische Implementierung des Beobachter-Musters darstellen. Als Template-Parameter können sogenannte *Aspekte*[27] angegeben werden, die in dem vorliegenden Fall detaillierte Informationen über den veränderten Parameter transportieren. Anhand dieser Informationen wird in der Subjektklasse entschieden, welche Beobachter benachrichtigt werden. Dadurch wird die Aktualisierung im Falle mehrerer Beobachterobjekte beschleunigt.

In den `update()`-Methoden der Beobachterklassen können die Informationen weiter verwendet werden. So berechnet im Falle des EE_{disp} -Mechanismus (4.44) der `TSRDirector` nach Gleichung (4.45) konsistente Parameterwerte und gibt diese an die jeweiligen Parameterobjekte weiter.

Eine Situation, in der eine Geschwindigkeitskonstante in der `StandardRateParameters`-Klasse verändert wird, ist durch den Pseudocode in Abbildung 4.13(b) dargestellt. Nachdem der neue Wert erhalten wurde, wird die `inform()`-Methode der Subjekt-Basisklasse aufgerufen, die einen Parameteraspekt vom Typ `KineticChange` erhält. Während der an thermodynamischen Parametern interessierte `TSRDirector` davon unbeeinflusst bleibt, kann der `GenericDirector`, falls die veränderte Geschwindigkeitskonstante in einer generischen Reaktionsrate vorkommt, den Wert an das entsprechende `GenericParameters`-Objekt weiterleiten. Durch den Update-Mechanismus aus Abbildung 4.10 wird anschließend das entsprechende `GenericRateLaw`-Objekt benachrichtigt.

4.3.8. Wiederverwendung und Erweiterbarkeit

Alle Klassen des Compilers sind in einer frei verfügbaren C++-Klassenbibliothek (Ecco-Bibliothek) zusammengefasst und dokumentiert. Eine Grundvoraussetzung zur *Wiederverwendung* des Compilers außerhalb der `EChem++`-Software ist somit erfüllt. Darüber hinaus wurde während der Entwicklung der Software weitestgehend auf spezifische Annahmen verzichtet, sodass eine Verwendung der Bibliothek in einem anderen als dem hier beschriebenen Zusammenhang möglich sein sollte.

Der genaue Grad der Kontextunabhängigkeit von Ecco lässt sich jedoch schwer bestimmen. So wird die Kinetik des Eingabemechanismus von Ecco stets durch *dimensionslose* Modellgleichungen abgebildet. Die Wiederverwendung der generierten Objekte im Zusammenhang unterschiedlicher Transformationsansätze ist somit möglich. Die Modellierung elektrochemischer und chemischer Reaktionskinetik ist allerdings durch das Angebot der implementierten Law-Klassen beschränkt. Die bisherigen Gesetze beinhalten temperatur- und druckunabhängige Konstanten, sodass zum Zeitpunkt dieser Arbeit isotherme ($\partial_t T = 0$) und isochore ($\partial_t V = 0$) Reaktionsbedingungen abgebildet werden können. Während letztere Annahmen im Kontext elektroanalytischer Methoden normalerweise ausreichen, können sie z.B. bei der Modellierung und Simulation von chemischen Reaktorprozessen die Wiederverwendung der Software erschweren. Ein wichtiger Aspekt ist daher die *Erweiterbarkeit* des Compilers.

Wie schon bei der Diskussion über die Wiederverwendung von Ecco, ist auch hier die freie Verfügbarkeit des Quellcodes eine grundlegende Voraussetzung zur Erweiterung der Software. Neben diesem durchaus pragmatischen Aspekt bildet das objekt-orientierte Design von Ecco das wichtigste Fundament für zukünftige Erweiterungen. In diesem Zusammenhang sei insbesondere die Verwendung der Fassade-, Befehl- und Kompositum-Muster erwähnt.

Die Fassade schirmt das gesamte Subsystem des Compilers vom Klientencode ab (siehe Abbildung 4.6). Dadurch wird eine Veränderung des Compiler-Inneren ermöglicht, ohne abhängige Softwarekomponenten neu übersetzen zu müssen. Dies erhöht die Wahrscheinlichkeit, dass von Ecco abhängige Softwarebausteine mit zukünftigen Versionen des Compilers kompatibel bleiben.

Eine schnelle Weiterentwicklung von Ecco wird durch die Aufteilung des Übersetzungsalgorithmus in einzelne Befehlsobjekte erreicht (Befehl, siehe Abbildung 4.7). Verändert man die Implementierung einer einzelnen konkreten Befehlsklasse, bleiben andere Befehlsobjekte davon unberührt und müssen nicht neu übersetzt werden. Darüber hinaus lässt sich der Algorithmus durch das Einbringen neuer Befehlsunterklassen auf einfache Weise erweitern. So wurden während der bisherigen Entwicklung immer wieder neue Befehlsklassen hinzugefügt, um die Funktionalität des Compilers sukzessive zu vergrößern. Dabei zeigte sich, dass durch den Einsatz der Entwurfsmuster der Compiler auf elegante und effiziente Weise durch Algorithmen zur semantischen Analyse des Eingabemechanismus sowie durch weitere kinetische Modelle erweitert werden konnte. Letzteres wurde insbesondere auch durch das Softwaredesign der Law-Klassen (Kompositum, Abbildung 4.11) erleichtert.

Neue kinetische Zusammenhänge lassen sich, wie schon beim Befehlsmuster, durch zusätzliche Law-Unterklassen einbringen. So könnte z.B. eine zukünftige "Marcus"-Klasse optional die Kinetik eines Elektronentransfers repräsentieren. In den zusammengesetzten Objekten der Randgleichungen würden dann, statt der Butler-Volmer-Gesetze, Marcus-Gesetze ausgewertet werden. Ein von den abstrakten Law-Klassen abhängiger, numerischer Algorithmus bliebe davon gänzlich unbeeinflusst.

Das Modellierungskonzept von Ecco beinhaltet sogenannte Randindizes, die als einzige geometrische Informationen dem Kinetik-Compiler übermittelt werden müssen. Der Kinetik-Compiler ist somit von einer Erweiterung von Ecco durch komplexere Modellgeometrien weitestgehend unabhängig.

Die Erweiterung der Eingabesprache sollte, sofern sie die Struktur der bestehenden Grammatik nicht verändert, ohne größere Schwierigkeiten erfolgen. Zusätzliche Syntaxregeln und die dazugehörigen Identifikationsnummern können in der definition-Klasse definiert werden (siehe dazu Abbildungen 4.4 – 4.6). So könnte eine erweiterte Eingabesprache z.B. die Formulierung von Mehrphasensystemen unterstützen. Der Übergang einer Substanz zwischen zwei Phasen ließe sich eventuell wie folgt

$$A\{\text{phase1}\} \rightleftharpoons A\{\text{phase2}\}; \quad b\langle 4 \rangle;$$

darstellen. Die Grenze zwischen den Phasen wird von einem Rand der Modellgeometrie beschrieben (siehe Γ_4 in Abbildung 3.2(a)).

5. Die Adapter-Bibliothek

Ecco generiert aus einem elektrochemischen Reaktionsmechanismus Law-Objekte, die die dimensionslosen Gleichungen (3.81) – (3.95) verkörpern. Die Law-Objekte erzeugen die entsprechenden Parameterobjekte, welche z.B. die für das Geschwindigkeitsgesetz relevanten dimensionslosen Reaktionsparameter beinhalten (siehe Abbildungen 4.10 und 4.13). Die Transformationen zwischen dimensionslosen und dimensionierten Parameterwerten werden durch sogenannte Adapterklassen geleistet, die in einer C++-Klassenbibliothek (Adapter-Bibliothek) zusammengefasst sind. Das Softwaredesign beruht auf den Adapter- und Beobachter-Mustern, sowie dem Singleton-Muster[27]. Die Implementierung der Parametertransformationen (3.77) – (3.99) basiert auf dem Open-Source-Projekt Quantity[137].

Das Ziel der Adapter-Bibliothek besteht darin, für jedes Parameterobjekt von Ecco eine Schnittstelle anzubieten, welche die in Abschnitt 3.10 definierten Hin- und Rücktransformationen zwischen dimensionierten und dimensionslosen Parametern unterstützt. Die Bibliothek ermöglicht im Zusammenhang mit der graphischen Benutzeroberfläche des EChem++-Projekts eine anwenderfreundliche, dimensionierte Eingabe der Modellparameter.

Die Speicherung der Parameter erfolgt in dimensionsloser Form weiterhin in den Ecco-Objekten. Dimensionierte Referenzwerte werden in der Adapter-Bibliothek durch ein ReferenceParameters-Objekt verwaltet (Abbildung 5.1). Um die Konsistenz zwischen Referenzwerten und davon abhängigen dimensionslosen Parameterwerten zu gewährleisten, wird wie schon bei der Implementierung von Ecco, das Beobachter-Muster verwendet (vgl. Abbildungen 4.10 und 4.13).

Verändert sich z.B. die Temperatur in der ReferenceParameters-Klasse, so werden sämtliche Beobachterobjekte benachrichtigt, deren Transformationsgleichungen von der Temperatur abhängen. In Abbildung 5.1 ist die Klasse DissButlerVolmerParameters¹ dargestellt, die mit Hilfe der Temperatur das dimensionslose Formalpotential berechnet und den Wert an das ButlerVolmerParameters-Objekt der Ecco-Bibliothek weitergibt. Die Pseudo-Implementierung entspricht der Gleichung (3.87).

Alle dimensionierten Größen werden durch Objekte der Quantity-Bibliothek abgebildet, z.B. durch Objekte der Klasse ElectricPotential. Die Quantity-Bibliothek kontrolliert bei der Rechnung mit dimensionierten Größen die Konsistenz der dazugehörigen physikalischen Einheiten. Die Kontrolle der Einheiten geschieht bereits zur Übersetzungszeit

¹Der Zusatz “Diss” deutet auf die Parameter hin, die zu einem heterogenen Elektronentransfer einer gelösten (*dissolved*) Spezies gehören.

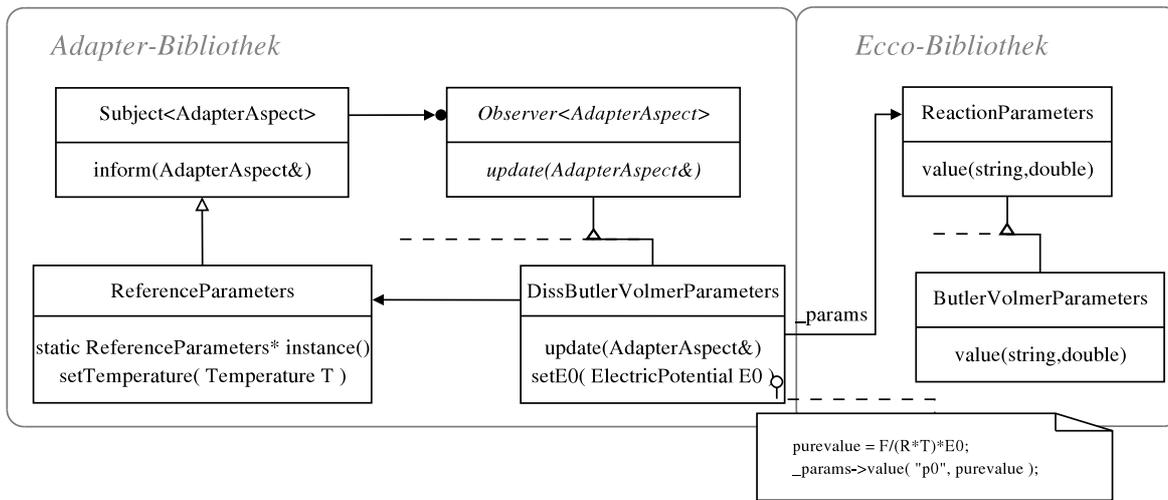


Abbildung 5.1: Grundstruktur der Adapter-Bibliothek am Beispiel dimensionierter Butler-Volmer-Parameter für Volumenelektronentransfers (DissButler-VolmerParameters).

des Quellcodes. Sie erleichterte somit während der Entwicklung der Adapterklassen eine frühe Fehlererkennung.

Die ReferenceParameters-Klasse implementiert das Singleton-Entwurfsmuster. Die Erzeugung eines Objekts der ReferenceParameters-Klasse erfolgt über die statische Funktion instance(). Die Funktion stellt während der Laufzeit des Anwenderprogramms die Eindeutigkeit der Referenzwerte sicher, indem nur ein einziges Objekt der Klasse instanziiert werden kann.

Teil II.

Simulation

6. Ein voll-adaptiver Algorithmus für Elektrodenprozesse

Die in Kapitel 3 behandelten Modellgleichungen (3.97) müssen mit Hilfe eines numerischen Verfahrens gelöst werden. Die Gleichungen bestehen im Allgemeinen aus parabolischen PDEs (Reaktions-Diffusions-Gleichungen) im Inneren der Zellgeometrie, kombiniert mit Systemen aus ODEs und algebraischen Gleichungen an den Rändern.

Als Lösung der numerischen Simulation erhält man zu jedem Zeitpunkt die Konzentrationsprofile der gelösten Spezies (u), sowie die Variation der Oberflächenkonzentrationen der adsorbierten Formen eines Randes (w). Die Berechnung der Observablen des elektrochemischen Experiments kann auf verschiedene Weisen erfolgen (siehe Abschnitt 3.11). Zum einen lässt sich der Strom (χ) oder das Elektrodenpotential (p) unmittelbar über das Strom-Potential-Residuum (3.98) berechnen (das Elektrodenpotential ist dabei durch die Geschwindigkeitsgesetze der Elektronentransfers gegeben). Wir wollen dieses Verfahren im Folgenden als *Variante 1 (V1)* bezeichnen.

Zum anderen existiert im potentialkontrollierten Fall die klassische Alternative, den Strom in einem nachverarbeitenden Schritt aus den berechneten Konzentrationsprofilen zu erhalten (*Variante 2, V2*). Dazu müssen die Ortsableitungen (Gradienten) der Volumenkonzentrationen an der Elektrodenoberfläche ($\partial_x u$) und die Zeitableitungen der Oberflächenkonzentrationen ($\partial_t w$) approximiert werden (siehe Abschnitte 3.11 und 4.3.6).

Die numerische Berechnung der Konzentrationsprofile kann in beiden Fällen durch verschiedene Phänomene erschwert werden. In Abhängigkeit von den Reaktions- und Randtermen stellen die Gleichungen im Allgemeinen nicht-lineare, steife und gekoppelte PDEs dar. Als Lösungen können steile, sich bewegende Konzentrationsfronten auftreten, die nur mit speziellen Algorithmen zufriedenstellend aufgelöst werden können.

Einen vielversprechenden Ansatz bieten hier sogenannte *adaptive* Verfahren, in denen sich die zur Ortsdiskretisierung eingesetzten Gitter automatisch an die Eigenschaft der Lösung anpassen. In Regionen, in denen sich Konzentrationsfronten ausbilden, werden die Gitter z.B. lokal verfeinert. Im Falle zeitabhängiger Probleme muss diese Gitteradaptation in jedem Zeitschritt der Simulation erfolgen.

Zusammen mit den Reaktionstermen sind lokale Gitterverfeinerungen bei der numerischen Behandlung von Reaktions-Diffusions-Gleichungen maßgeblich für die sogenannte *Steifheit* der Differentialgleichungen verantwortlich [138]. Aus Stabilitätsgründen müssen daher *implizite* Zeitintegratoren eingesetzt werden (siehe z.B. [139]).

Die Simulation elektrochemischer Experimente wird darüber hinaus durch zeitabhängige, zum Teil sogar unstetige Randbedingungen erschwert. Häufig bewirken die Randterme in relativ kurzen Zeitabschnitten der Simulation große Veränderungen in den Lösungsvariablen.¹ Um unter solchen Umständen eine verlässliche Approximation zu erreichen, muss die Zeitintegration mit sehr kleinen Schrittweiten durchgeführt werden. In anderen Zeitabschnitten hingegen sind kleine Integrationsschritte unnötig und führen zu einem insgesamt ineffizienten Verhalten der Algorithmen. Wie schon bei der Ortsdiskretisierung ist daher auch für die Zeitintegration eine automatische Adaption der Schrittweite von Vorteil.

Aus obigen Überlegungen folgt, dass sich zur Lösung von elektrochemischen Modellen im Allgemeinen *voll-adaptive* numerische Verfahren anbieten, in denen sowohl die Diskretisierungsgitter im Ort als auch die Schrittweiten in der Zeit kontrolliert werden.

Ein wichtiger Aspekt adaptiver Algorithmen ist die verlässliche Schätzung des Fehlers in den simulierten Daten, der durch die Zeit- und Ortsdiskretisierung entsteht. So wird auf der Grundlage einer lokalen Fehlerschätzung der adaptive Prozess in Zeit und Ort gesteuert. In elektrochemischen Simulationen sind dabei nicht nur kleine Fehler in den berechneten Konzentrationsprofilen von Interesse, sondern insbesondere auch in der primären Observablen des elektroanalytischen Experiments (Strom oder Elektrodenpotential).

Im Kontext elektrochemischer Simulationen entwickelte Bieniasz einen voll-adaptiven Algorithmus für 1D-Modelle auf der Basis finiter Differenzen[116, 140–152]. Im Hinblick auf die Simulation von höher-dimensionierten, komplizierteren Modellgeometrien kann die Verwendung von Finite-Differenzen-Verfahren zur Ortsdiskretisierung jedoch zu Schwierigkeiten führen[63]. Hier gilt die Finite-Elemente-Methode (FE-Methode) als flexible Alternative[63, 153–156]. Darüber hinaus wurde mit der Entwicklung zahlreicher *a posteriori* FE-Fehlerschätzer in den letzten Jahrzehnten eine fundierte mathematische Grundlage für die Kontrolle adaptiver FE-Algorithmen geschaffen[60, 157–176].

Nann und Heinze beschrieben 1999 einen ersten solchen Algorithmus in der Elektrochemie[118], der später zur Simulation von 2D-Modellen eingesetzt wurde[41, 48]. Die Gitteradaption wird dabei durch einen Fehlerschätzer gesteuert, der die Konzentrationsgradienten der numerischen Lösung benachbarter Elemente vergleicht. In Regionen mit großem Unterschied in den Gradienten wird das FE-Gitter durch die sogenannte *h*-Verfeinerung (Unterteilung von Elementen) lokal verbessert. Diese Methode stellt zwar einen sehr allgemein einsetzbaren Ansatz dar, kann jedoch keine zuverlässige Aussage über den Fehler in den simulierten Daten liefern. Eine Aussage über den Fehler in der Observablen ist nicht möglich.

Um diesen Nachteil zu umgehen, entwickelten Harriman et al. einen Fehlerschätzer für stationäre 2D-Modelle[36, 177] und erweiterten ihren Ansatz später auf dynamische Systeme[37]. Die Idee besteht darin, den Fehler alleine in der Observablen des Experiments zu kontrollieren. Der darin enthaltene Schätzer basiert auf Fehlerschranken für den Strom (oder für ein anderes *lineares* Funktional der PDE-Lösung) und lässt sich somit im

¹Als Beispiel seien die Randterme chronoamperometrischer Experimente erwähnt (Kapitel 7).

Prinzip auf potentialkontrollierte Experimente anwenden. Im Falle stromkontrollierter Experimente hängt das Elektrodenpotential jedoch auf *nicht-lineare* Weise von den Konzentrationsprofilen ab. Die Anwendung des Fehlerschätzers ist daher für solche Probleme zu bezweifeln. Die Fehlerschranken des Schätzers verlangen die numerische Approximation eines sogenannten dualen (oder adjungierten) Problems (siehe Gleichungen (46) und (47) in[36]). Folglich kann im Allgemeinen die Verlässlichkeit des Fehlerschätzers nicht garantiert werden[36, 38]. Darüber hinaus wurde für zeitabhängige Modelle, wie sie in der Elektroanalytik überwiegend auftreten, eine zusätzliche numerische Berechnung des dualen Problems für zu aufwendig erachtet, und es wurden weitere Abschätzungen eingeführt. Die Herleitung rigoroser Fehlerschranken war daher noch nicht möglich, und der Fehlerschätzer des Algorithmus kann wie schon zuvor bei Nann und Heinze bisher nur als Verfeinerungskriterium betrachtet werden[37]. Das Verhalten des von Harriman et al. vorgeschlagenen Algorithmus im Falle elektrochemischer Reaktionsmechanismen, die zu steilen Konzentrationfronten in der Diffusionsschicht führen, ist noch nicht untersucht worden. Es bleibt abzuwarten, ob eine Gitteradaptation, die alleine auf einer Fehlerkontrolle im Strom basiert, in diesen Fällen zu guten Ergebnissen führt. Die mathematischen Ausdrücke des Schätzers hängen zudem anscheinend von modell-spezifischen Informationen ab[178, 179]. Ein erfolgreicher Einsatz des Ansatzes in allgemeiner Modellierung und Simulationssoftware erscheint daher mindestens zweifelhaft. Der Algorithmus beruhte ursprünglich auf der h -Verfeinerung des FE-Gitters und wurde kürzlich auf die hp -Methode² erweitert[180, 181].

Abercrombie und Denuault beschreiben einen ähnlichen Ansatz[38], welcher jedoch ebenfalls auf die zusätzliche Lösung eines dualen Problems führt.

Eine andere Klasse voll-adaptiver Algorithmen wurde kürzlich von Lang[60] zur Lösung parabolischer PDAEs vorgestellt und für die Simulation mehrerer Modelle aus nicht-elektrochemischen Feldern der Naturwissenschaften eingesetzt. Die Verfahren beschreiben Abkömmlinge der sogenannten *Rothe*-Methode, in denen die Differentialgleichungen zunächst in der Zeit durch *Rosenbrock*-Verfahren diskretisiert werden. Die daraus resultierenden stationären Probleme werden anschließend mit Hilfe adaptiver finiter Elemente behandelt.

Die Steuerung der Zeitschrittweite erfolgt über den klassischen Ansatz eines eingebetteten Rosenbrock-Verfahrens. Der Fehler der FE-Diskretisierung im Ort wird mit Hilfe einer hierarchischen Erweiterung des linearen FE-Approximationsraums abgeschätzt[170, 182]. Er lässt sich somit durch kleine Residuumprobleme auch im Falle zeitabhängiger Modelle mit relativ geringem Aufwand berechnen.

Die Stabilität des Algorithmus wurde für einstufige Rosenbrock-Verfahren exakt bewiesen[60]. In mehrstufigen Verfahren ist eine nicht-lineare Fehlerfortpflanzung möglich. Unter praktischen Gesichtspunkten besitzen die Störungsterme jedoch nur geringen Einfluss[60].

²Die hp -Verfeinerung beschreibt ein Hybrid aus h - und p -Verfeinerung, wobei zur Verbesserung der Approximation zusätzlich der Grad der Interpolationspolynome variiert wird (p -Verfeinerung)

Ein entscheidender Vorteil gegenüber den bisherigen adaptiven Ansätzen ist, dass sich die allgemeinen Randgleichungen elektrochemischer Modelle (siehe (3.97) und (3.98)) auf elegante Weise in den Algorithmus integrieren lassen. Somit kann, neben der Fehlerkontrolle in den Konzentrationsprofilen gelöster Teilchen, auch eine Kontrolle des Fehlers für die Konzentrationen adsorbierter Moleküle und in der Observablen durchgeführt werden. Dies gilt gleichsam für die Simulation von potentialkontrollierten sowie stromkontrollierten Experimenten (im stromkontrollierten Fall wurde das Einbringen des Elektrodenpotentials in ein Rosenbrock-Schema bereits beschrieben[183]).

Darüber hinaus bieten die Rosenbrock-Methoden die Möglichkeit, ohne nennenswerten zusätzlichen Aufwand die Zeitableitungen ($\partial_t w$) zu berechnen. Dies ist insbesondere für $V2$ zur Berechnung des Stroms mittels der in Abschnitt 3.11 hergeleiteten Strom-Massenfluss-Beziehung von Bedeutung.

Die in[60] entwickelten Algorithmen stehen in Form des Software-Paketes KARDOS zur Verfügung. KARDOS ist in der Programmiersprache C geschrieben und lässt sich daher nicht ohne Aufwand in ein objekt-orientiertes Rahmenwerk integrieren. Restriktive Lizenzbedingungen verhindern zudem den uneingeschränkten Einsatz der Software in Open-Source-Code, sodass die Wiederverwendung von KARDOS in dieser Arbeit nicht möglich war.

In den folgenden Abschnitten werden daher auf[60] basierende Verfahren für die Simulation der in Teil I eingeführten 1D-Modelle beschrieben und in Form einer objekt-orientierten Open-Source-Softwarebibliothek implementiert.

Dafür bringen wir die Modellgleichungen (3.97) und (3.98) in Abschnitt 6.1 zunächst in die allgemeinere Form eines PDAE-Systems. Im Anschluss erfolgt die Zeitdiskretisierung der PDAEs durch adaptive Rosenbrock-Methoden[60] (Abschnitt 6.2). Die resultierenden elliptischen PDEs sowie die Gleichungen des Fehlerschätzers werden in Abschnitt 6.3 behandelt. Ein wichtiger Aspekt für die Stabilität des Algorithmus ist die Wahl der Fehlerskalierung. Diese wird in Abschnitt 6.4 diskutiert. Eine Übersicht über die Struktur des Algorithmus wird in Abschnitt 6.5 gegeben.

Abschließend wird das objekt-orientierte Design des Algorithmus und insbesondere die Kopplung an den in Kapitel 4 vorgestellten Kinetik-Compiler besprochen (Abschnitt 6.6).

Einige Konzepte des Algorithmus wurden von uns bereits beschrieben[184]. Neben einer Veränderung der Fehlerskalierung, wird im Folgenden der Algorithmus insbesondere durch die Kontrolle des Fehlers in den Observablen erweitert.

6.1. Die partielle differentiell-algebraische Formulierung

Für die Entwicklung eines adaptiven Verfahrens, welches den Fehler sowohl in den Konzentrationen als auch im Strom oder dem Elektrodenpotential kontrolliert, wird die Observable o (potentialkontrolliert: $o = \chi$, stromkontrolliert: $o = p$) als zusätzliche Lösungsvariable in die PDEs (3.97) eingebracht ($V1$). Mit $v = (o, w, u)$ und unter Ver-

wendung des Strom-Potential-Residuums (3.98) als zusätzlicher Randgleichung lässt sich das Gesamtproblem allgemein als PDAE-System

$$\begin{cases} H \cdot \partial_t v(t, x) = F(v(t, x)) & x \in \mathcal{I}, \quad t > 0 \\ G \cdot \partial_t v(t, x) = g(v(t, x)) - B \cdot v(t, x) & x \in \Gamma, \quad t > 0 \\ v(0, x) = v^0(x) & x \in \bar{\mathcal{I}} \end{cases} \quad (6.1)$$

darstellen. Das spezielle Aussehen der Gleichungen wird durch die Matrizen H , G und B sowie die F - und g -Terme bestimmt. Über eine geeignete Definition der Matrizen H und G können algebraische Bedingungen an die Lösung im Inneren der Zelle bzw. an deren Ränder eingebracht werden. Für die hier betrachteten Modelle gilt

$$H = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \end{pmatrix} \quad \text{und} \quad G = \begin{pmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (6.2)$$

mit den $N \times N$ -Einheitsmatrizen I . In H steht die Dimension N für die Anzahl der gelösten Teilchen (N_{diss}), in G für die der adsorbierten Formen (N_{ads}).

Der F -Term in (6.1) beschreibt Transport- und Reaktionsterme im Inneren.

$$F(v) = \begin{pmatrix} 0 \\ 0 \\ \rho\beta\partial_{xx}u + R(u) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \rho\beta\partial_{xx}u + S_{\text{hom}}^T \cdot \mathbf{r}_{\text{hom}} \end{pmatrix} \quad (6.3)$$

Die Randgleichungen werden mittels

$$g(v) = \begin{pmatrix} g_o \\ g_w \\ g_u \end{pmatrix} = \begin{pmatrix} \chi - \sum_{j=0}^{N_{\text{ET}}-1} \nu_{j,e} \cdot r_{\text{ET},j} \\ S_{\text{ads}}^T \cdot \mathbf{r}_{\text{ads}} \\ S_{\text{ET}}^T \cdot \mathbf{r}_{\text{ET}} + \tilde{S}_{\text{ads}}^T \cdot \tilde{\mathbf{r}}_{\text{ads}} \end{pmatrix} \quad (6.4)$$

und

$$B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\beta\partial_x \end{pmatrix} \quad (6.5)$$

definiert. Die oberen zwei Zeilen und Spalten in den Ausdrücken (6.2) – (6.5) beschreiben die Gleichungen der Observablen und der adsorbierten Teilchen, die letzte Zeile und Spalte die der gelösten Spezies. Es sei betont, dass die Variablen w und u , je nach Reaktionsmechanismus, sämtliche Oberflächen- und Volumenkonzentrationen beinhalten:

$$\begin{aligned} w &= (w^{(0)}, w^{(1)}, \dots, w^{(N_{\text{ads}}-1)}) \\ u &= (u^{(0)}, u^{(1)}, \dots, u^{(N_{\text{diss}}-1)}) \end{aligned}$$

Die Variable o hingegen steht für die skalare Größe der Observablen einer Elektrode. Der Anfangswert $o^0(x)$ der Observablen wird mit Hilfe des Strom-Potential-Residuums (3.98) bestimmt.

Falls der Strom anstatt über das Strom-Potential-Residuum mit Hilfe der Strom-Massenfluss-Beziehung aus Abschnitt 3.11 berechnet werden soll ($V2$), können einfachere Modellgleichungen betrachtet werden. Nun wird die Observable nicht mehr als Variable des Systems (6.1) behandelt: $v = (w, u)$. Für H, G, B und für die F - und g -Terme gelten statt (6.2) – (6.5) die Beziehungen

$$H = \begin{pmatrix} 0 & 0 \\ 0 & I \end{pmatrix} \quad G = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} \quad (6.6)$$

$$F(v) = \begin{pmatrix} 0 \\ \rho\beta\partial_{xx}u + R(u) \end{pmatrix} = \begin{pmatrix} 0 \\ \rho\beta\partial_{xx}u + S_{\text{hom}}^T \cdot \mathbf{r}_{\text{hom}} \end{pmatrix} \quad (6.7)$$

$$g(v) = \begin{pmatrix} g_w \\ g_u \end{pmatrix} = \begin{pmatrix} S_{\text{ads}}^T \cdot \mathbf{r}_{\text{ads}} \\ S_{\text{ET}}^T \cdot \mathbf{r}_{\text{ET}} + \tilde{S}_{\text{ads}}^T \cdot \tilde{\mathbf{r}}_{\text{ads}} \end{pmatrix} \quad (6.8)$$

und

$$B = \begin{pmatrix} 0 & 0 \\ 0 & -\beta\partial_x \end{pmatrix} \quad (6.9)$$

6.2. Rosenbrock-Methoden

In der Rothe-Methode werden zeitabhängige PDEs als ein System von ODEs der allgemeinen Form

$$\partial_t y = f(t, y), \quad t > 0, \quad y(0) = y_0 \quad (6.10)$$

betrachtet und mit Hilfe wohlbekannter, numerischer Verfahren zur Behandlung von ODEs in der Zeit diskretisiert. Häufig werden aus Stabilitätsgründen dafür implizite Methoden verwendet, wobei im Allgemeinen nicht-lineare Gleichungssysteme entstehen, die durch Newton-Verfahren gelöst werden müssen. Als einfachstes Beispiel einer impliziten Methode sei das implizite Euler-Verfahren genannt, welches zur Berechnung der diskreten Lösung y_{n+1} auf die Vorschrift

$$y_{n+1} = y_n + \tau \cdot f(t_{n+1}, y_{n+1}) \quad (6.11)$$

führt.

Rosenbrock-Methoden beschreiben linearisierte, implizite Verfahren, in denen die Schritte der Newton-Iteration in die Vorschrift (6.11) integriert werden [139, 185, 186]. Folglich

bieten Rosenbrock-Verfahren den Vorteil, die Newton-Iterationen zur Lösung der Gleichungssysteme nicht explizit kontrollieren zu müssen. Sie erlauben eine schnelle Veränderung der Schrittweite τ und sind relativ einfach zu implementieren[60]. Darüber hinaus ermöglicht die enge Verwandtschaft zwischen verschiedenen Rosenbrock-Verfahren eine elegante objekt-orientierte Strukturierung sowie eine effiziente Implementierung der Algorithmen mit Hilfe des Strategie-Musters (siehe Abschnitt 6.6.1).

6.2.1. Der ε -eingebettete Ansatz

Das PDAE-System (6.1) wird durch Rosenbrock-Methoden in der Zeit diskretisiert. Wir wollen hier dem sogenannten ε -eingebetteten Ansatz aus[60] folgen, der auf Ideen von Roche und Lubich basiert[187, 188]. Dafür wird eine neue Variable $z = \partial_t v$ eingeführt und (6.1) als Grenzfall ($\varepsilon \rightarrow 0$) des Störungsproblems

$$\begin{aligned}\partial_t v &= z \\ \varepsilon \partial_t z &= F(v) - H \cdot z\end{aligned}\tag{6.12}$$

betrachtet. Für Probleme der Form (6.1) erhält man daraus das Rosenbrock-Schema (vgl. Gleichungen (V.10) – (V.14) in[60])

$$\begin{aligned}v_{n+1} &= v_n + \sum_{i=1}^s m_i V_{ni} \\ \hat{v}_{n+1} &= v_n + \sum_{i=1}^s \hat{m}_i V_{ni} \\ z_{n+1} &= z_n + \sum_{i=1}^s m_i \left(\frac{1}{\tau} \sum_{j=1}^i (c_{ij} - s_{ij}) V_{nj} + (\sigma_i - 1) z_n \right) \\ \hat{z}_{n+1} &= z_n + \sum_{i=1}^s \hat{m}_i \left(\frac{1}{\tau} \sum_{j=1}^i (c_{ij} - s_{ij}) V_{nj} + (\sigma_i - 1) z_n \right)\end{aligned}\tag{6.13}$$

mit

$$\left(\frac{1}{\tau\gamma} H - A_n \right) V_{ni} = F(V_i) - H \cdot \sum_{j=1}^{i-1} \frac{c_{ij}}{\tau} V_{nj}\tag{6.14}$$

und den internen Größen

$$t_i = t_n + \alpha_i \tau, \quad V_i = v_n + \sum_{j=1}^{i-1} a_{ij} V_{nj}\tag{6.15}$$

Die Randgleichungen von (6.1) müssen auf analoge Weise linearisiert werden. Man erhält

$$\left(\frac{1}{\tau\gamma} G - D_n \right) V_{ni} = g(V_i) - B \cdot V_i - G \cdot \sum_{j=1}^{i-1} \frac{c_{ij}}{\tau} V_{nj}\tag{6.16}$$

In den Gleichungen (6.14) und (6.16) stehen die Symbole A_n und D_n für die Jacobi-Matrizen der rechten Seiten

$$\begin{aligned} A_n &= \partial_v F(v)|_{v=v_n} \\ D_n &= \partial_v (g(v) - B \cdot v)|_{v=v_n} \end{aligned} \quad (6.17)$$

Die verschiedenen Rosenbrock-Verfahren werden durch die Koeffizienten m_i , \hat{m}_i , c_{ij} , s_{ij} , σ_i , γ , α_i und a_{ij} der Formeln (6.13) – (6.16) charakterisiert. Zusammen mit der Anzahl der Integrationsstufen s bestimmen die Koeffizienten die Konvergenzordnung, Stabilität sowie die Effizienz einer Methode. Die Werte der Koeffizienten sind tabellarisch in der Literatur angegeben [60, 185, 186, 189, 190].

In den Ausdrücken (6.13) werden die Lösungswerte v_{n+1} und z_{n+1} eines diskreten Zeitpunkts t_{n+1} aus den vorherigen Lösungen v_n , z_n und den Stufenwerten V_{ni} gewonnen. Letztere müssen zuvor durch die Lösung der linearen Gleichungssysteme (6.14) und (6.16) berechnet werden. Das Symbol τ beschreibt die aktuelle Schrittweite des Rosenbrock-Schrittes, welche mit Hilfe der Lösungen \hat{v} und \hat{z} des eingebetteten Verfahrens³ vorhergesagt wird (siehe Abschnitt 6.2.2). Zur Berechnung der rechten Seiten der linearen Gleichungssysteme (6.14) und (6.16) werden die internen Stufenwerte V_i benötigt (siehe Gleichung (6.15)). Die internen Zeitstufen t_i werden zur Auswertung der Anregungsfunktionen (siehe Abschnitt 3.2) verwendet.

Es sei betont, dass durch die Anwendung des ε -eingebetteten Ansatzes (6.12) die Berechnung der Zeitableitungen $z = \partial_t v$ alleine durch eine zusätzliche Summenbildung in (6.13) erreicht wird. Die Rosenbrock-Verfahren stellen somit die $\partial_t w$ -Werte zur Berechnung der Observablen gemäß Abschnitt 3.11 (V2) auf eine sehr effiziente Weise zur Verfügung.

Für V1 wendet man die Zeitdiskretisierung (6.14) und (6.16) mit dem Komponentenvektor

$$V = \begin{pmatrix} O \\ W \\ U \end{pmatrix}$$

auf die elektrochemischen Modelle (6.1) – (6.5) an. Mit (6.14) folgt

$$\begin{aligned} \left(\frac{1}{\tau\gamma} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \end{pmatrix} - \left[\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \rho\beta\partial_{xx} \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \partial_u R(u_n) \end{pmatrix} \right] \right) V_{ni} = \\ \begin{pmatrix} 0 \\ 0 \\ \rho\beta\partial_{xx} U_i \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ R(U_i) \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & I \end{pmatrix} \cdot \sum_{j=1}^{i-1} \frac{c_{ij}}{\tau} V_{nj} \end{aligned} \quad (6.18)$$

was den diskretisierten, rein parabolischen PDEs (16) in [184] entspricht. Gleichung (6.18) beschreibt ein elliptisches PDE-System, welches in Abschnitt 6.3 mittels finiter Elemente

³Das eingebettete Verfahren ist durch die Koeffizienten \hat{m} definiert (siehe Gleichung (6.13)).

behandelt wird.

Die diskreten Randgleichungen (6.16) ergeben sich gemäß

$$\left(\frac{1}{\tau\gamma} \begin{pmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{pmatrix} - \left[\begin{pmatrix} \partial_o g_o & \partial_w g_o & \partial_u g_o \\ \partial_o g_w & \partial_w g_w & \partial_u g_w \\ \partial_o g_u & \partial_w g_u & \partial_u g_u \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\beta\partial_x \end{pmatrix} \right] \right) V_{ni} =$$

$$\begin{pmatrix} g_o(V_i) \\ g_w(V_i) \\ g_u(V_i) \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ -\beta\partial_x U_i \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & 0 \end{pmatrix} \cdot \sum_{j=1}^{i-1} \frac{c_{ij}}{\tau} V_{nj} \quad (6.19)$$

wobei auf der linken Seite die partiellen Ableitungen der g -Terme mittels der Lösung v_n ausgewertet werden: $g = g(v_n)$.

Für $V\mathcal{2}$ (siehe (6.6) – (6.9)) gelten mit dem Komponentenvektor

$$V = \begin{pmatrix} W \\ U \end{pmatrix}$$

weiterhin die Gleichungen (6.18) und (6.19), jedoch ohne die erste Zeile und Spalte.

6.2.2. Adaptive Schrittweitensteuerung

Die Rosenbrock-Methoden erlauben in der Tradition von Runge-Kutta-Verfahren eine effiziente Steuerung der Zeitschrittweiten. Für jede Rosenbrock-Methode ist eine eingebettete Formel von geringerer Konvergenzordnung durch die \hat{m}_i -Koeffizienten in (6.13) gegeben. Der lokale Diskretisierungsfehler in den Lösungskomponenten eines Zeitschrittes kann somit durch die Differenz $r = v - \hat{v}$ abgeschätzt werden (für Details siehe [60, 139]). Die Residuen r lassen sich zudem aus (6.13) direkt mittels

$$r_{n+1} = \sum_{i=1}^s (m_i - \hat{m}_i) V_{ni} \quad (6.20)$$

berechnen, wobei die Differenzen $m_i - \hat{m}_i$ bereits vor der Simulation bestimmt werden können. Für den Zeitfehler eines Rosenbrock-Schrittes gilt⁴

$$\varepsilon_{n+1} = \| r_{n+1} \| \quad (6.21)$$

wobei das Symbol $\| \cdot \|$ für eine skalierte Norm steht (siehe Abschnitt 6.4).

Durch die Größe des Fehlers wird die Adaption der Zeitschrittweite kontrolliert. Liegt ε oberhalb der gewünschten Fehlertoleranz, wird die Lösung verworfen und erneut ein Rosenbrock-Schritt mit reduzierter Schrittweite τ durchgeführt. Liegt er unterhalb des

⁴Im Folgenden soll ε stets eine Fehlergröße symbolisieren. Sie steht in keiner Beziehung zum ε -eingebetteten Ansatz (Abschnitt 6.2.1, Gleichung (6.12)).

Toleranzwertes, wird die Lösung akzeptiert und die Simulation fortgesetzt. In diesem Fall wird die Schrittweite gegebenenfalls vergrößert. Die Vorhersage einer möglichst optimalen Schrittweite wird durch den Ansatz von Gustafsson erreicht[191], welcher von Lang nur geringfügig verändert wurde[60, 192]. Es gilt

$$\tau_{\text{pred}} = \min \left(\frac{\tau_n}{\tau_{n-1}} \left(\frac{\varepsilon_n}{\varepsilon_{n+1}\varepsilon_{n+1}} \right)^{1/p}, \left(\frac{1}{\varepsilon_n} \right)^{1/p} \right) \cdot \tau_n \quad (6.22)$$

Eine ausführliche Beschreibung des sogenannten PI-Controllers findet man in[191]. Zusätzlich zu der Vorhersage (6.22) werden üblicherweise heuristische Sicherheitsfaktoren verwendet, um eine möglichst gleichmäßige Veränderung in τ zu erreichen. Für die Simulation der im Rahmen dieser Arbeit untersuchten elektrochemischen Modelle hat sich die Strategie

$$\begin{aligned} \tau_{n+1} &= \min(1.5 \cdot \tau_n, 0.9 \cdot \tau_{\text{pred}}), & \text{falls } \tau_n \text{ akzeptiert} \\ \tau_n &= \max(0.5 \cdot \tau_n, 0.9 \cdot \tau_{\text{pred}}), & \text{falls } \tau_n \text{ verworfen} \end{aligned} \quad (6.23)$$

bewährt.

6.3. Adaptive finite Elemente

Zur Diskretisierung der elliptischen PDEs (6.18) folgen wir ansatzweise dem Multilevel-FE-Verfahren, welches von Yserentant, Deuffhard und Leinen entwickelt[182, 193] und von Lang[60] verwendet wurde.⁵ Die zugrunde liegenden theoretischen Konzepte des hier implementierten Algorithmus werden in der mathematischen Literatur in verschiedenen Werken beschrieben[173, 182, 189, 190, 192–195] und sollen in den folgenden Absätzen zusammengefasst werden.⁶

Die FE-Diskretisierung der PDEs (6.18) führt zu linearen Gleichungssystemen (LGS), die in jedem Zeitschritt der Simulation gelöst werden müssen. Die Bezeichnung *Multilevel* deutet dabei zumeist auf die Verwendung eines *iterativen* Verfahrens zur Behandlung der LGS hin, in dem die Lösung eines gröbereren FE-Gitters zur Berechnung der Lösung des verfeinerten Gitters eingesetzt wird.⁷ Iterative Methoden werden aus Effizienzgründen insbesondere zur Lösung von linearen Gleichungssystemen für höherdimensionale Modellgeometrien eingesetzt. Wie für 1D-Modelle üblich, verwenden wir hier stattdessen ein *direktes* Verfahren (siehe z.B. die Diskussion in[60] S.65).

⁵Die Implementierung der KARDOS-Software[60] basiert auf dem hierarchischen KASKADE-Löser[182] für elliptische PDEs.

⁶Zur Lösung der PDEs werden keine hierarchischen Basen gemäß[182, 193] verwendet. Die Diskretisierung erfolgt durch den klassischen, linearen Interpolationsansatz (Abschnitt 6.3.1). Der *a posteriori* Fehlerschätzer basiert somit, im Gegensatz zu[182], auf einer hierarchischen Erweiterung des *klassischen* Interpolationsansatzes (siehe Abschnitt 6.3.2).

⁷Die Aussage basiert auf einer E-Mail-Korrespondenz mit Herrn Dr. P. Leinen.

6.3.1. Variationelle Formulierung und FE-Interpolation

Der Grundansatz der Finite-Elemente-Methoden ist die Überführung partieller Differentialgleichungen in eine sogenannte *variationelle* Formulierung (siehe z.B.[154]). Wir betrachten dazu zunächst das zu (6.18) äquivalente System

$$\left(\frac{1}{\tau\gamma} I - [\rho\beta\partial_{xx} + \partial_u R(u_n)] \right) U_{ni} = \rho\beta\partial_{xx}U_i + R(U_i) - I \cdot \sum_{j=1}^{i-1} \frac{c_{ij}}{\tau} U_{nj} \quad (6.24)$$

Durch Multiplikation der Gleichung mittels einer sogenannten Testfunktion, δ , und anschließender Integration über das Intervall $\bar{\mathcal{I}} = [0, 1]$, erhält man

$$\begin{aligned} \frac{1}{\tau\gamma} \int_0^1 U_{ni} \cdot \delta \, dx - \rho\beta \int_0^1 \partial_{xx}U_{ni} \cdot \delta \, dx - \int_0^1 \partial_u R(u_n)U_{ni} \cdot \delta \, dx \\ = \rho\beta \int_0^1 \partial_{xx}U_i \cdot \delta \, dx + \int_0^1 R(U_i) \cdot \delta \, dx - \sum_{j=1}^{i-1} \frac{c_{ij}}{\tau} \int_0^1 U_{nj} \cdot \delta \, dx \end{aligned} \quad (6.25)$$

woraus sich durch partielle Integration die variationelle Formulierung

$$\begin{aligned} \frac{1}{\tau\gamma} \int_0^1 U_{ni} \cdot \delta \, dx + \rho\beta \int_0^1 \partial_x U_{ni} \cdot \partial_x \delta \, dx - \int_0^1 R_u(u_n)U_{ni} \cdot \delta \, dx \\ - \underbrace{\rho\beta (\partial_x U_{ni} \cdot \delta|_{x=1} - \partial_x U_{ni} \cdot \delta|_{x=0})}_{(*)} \\ = -\rho\beta \int_0^1 \partial_x U_i \cdot \partial_x \delta \, dx + \int_0^1 R(U_i) \cdot \delta \, dx - \sum_{j=1}^{i-1} \frac{c_{ij}}{\tau} \int_0^1 U_{nj} \cdot \delta \, dx \\ + \underbrace{\rho\beta (\partial_x U_i \cdot \delta|_{x=1} - \partial_x U_i \cdot \delta|_{x=0})}_{(\star)} \end{aligned} \quad (6.26)$$

ergibt.⁸ Anstelle der Randbedingungen (*) und (★) müssen für $V1$ die Gleichungen (6.19) eingebracht werden. Dazu betrachten wir ohne Bedenken der Allgemeinheit den Rand $x = 0$ und lösen die letzte Zeile von (6.19) nach $-\beta\partial_x U_{ni}$ auf

$$-\beta\partial_x U_{ni} = \left(\partial_o g_u \quad \partial_w g_u \quad \partial_u g_u \right) \cdot V_{ni} + g_u(V_i) + \beta\partial_x U_i \quad (6.27)$$

Setzt man (6.27) in (6.26) ein, erhält man für den Rand $x = 0$ die Terme

$$\dots - \rho \left(\partial_o g_u \quad \partial_w g_u \quad \partial_u g_u \right) \cdot V_{ni} - \rho g_u(V_i) - \rho\beta\partial_x U_i = \dots - \rho\beta\partial_x U_i \quad (6.28)$$

⁸Damit die variationelle Formulierung (6.26) wohldefiniert ist, müssen die Funktionen U_{ni} , U_i und δ im Sobolev-Raum $\mathcal{H}^1(\mathcal{I}) = \{u \in \mathcal{L}^2(\mathcal{I}) : \partial_x u \in \mathcal{L}^2(\mathcal{I})\}$ aus *quadratisch integrierbaren* Funktionen liegen, wobei $\mathcal{L}^2(\mathcal{I}) = \{u : \int_0^1 u^2 dx < \infty\}$ [154, 156].

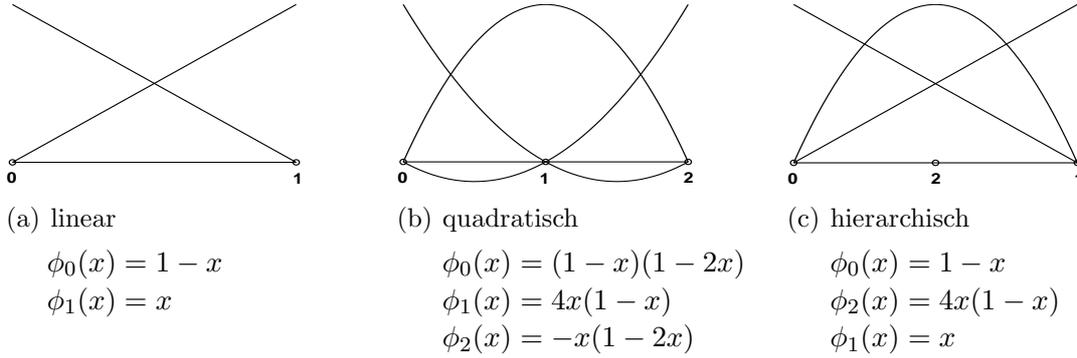


Abbildung 6.1: 1D finite Elemente mit Knotennummerierung: Linear (a) und quadratische Interpolation (b) sowie das in dieser Arbeit verwendete hierarchische Element mit quadratischer *Bubble*-Funktion (c).

und somit

$$\dots - \rho \left(\partial_o g_u \quad \partial_w g_u \quad \partial_u g_u \right) \cdot V_{ni} = \dots + \rho g_u(V_i) \quad (6.29)$$

Die Kopplung zwischen der Observablen, den adsorbierten und den gelösten Spezies am Rand ist durch die Terme der linken Seite von (6.29) festgelegt und muss bei der Aufstellung der voll-diskretisierten Gleichungssysteme berücksichtigt werden. Gleichzeitig gelten am Rand weiterhin die oberen zwei Zeilen von (6.19), welche die Gleichung der Observablen und die Gleichungen der adsorbierten Formen beschreiben.

Es sei erwähnt, dass im Falle homogener Neumann-Randbedingungen (3.72) die Terme auf beiden Seiten von (6.29) entfallen.

Die Idee des Multilevel-Prozesses ist es, zur Berechnung der diskreten Rosenbrock-Lösungen $u_{n+1,k}$ und $z_{n+1,k}$ gemäß (6.13), die dazugehörigen Stufenwerte $V_{ni,k}$ in einer Sequenz von FE-Unterräumen zu suchen

$$\mathcal{V}_{n+1,1} \subset \mathcal{V}_{n+1,2} \subset \dots \subset \mathcal{V}_{n+1,k} \quad (6.30)$$

welche sich durch die lokal verfeinerten Gitter

$$\mathcal{I}_{n+1,1} \subset \mathcal{I}_{n+1,2} \subset \dots \subset \mathcal{I}_{n+1,k} \quad (6.31)$$

ergeben. Beginnend mit dem größten Gitter $\mathcal{I}_{n+1,1}$ des Zeitschrittes $t_n \rightarrow t_{n+1}$, zählt Index k die Verfeinerungslevel. Die Gitter setzen sich jeweils aus den hierarchischen Elementen (Abbildung 6.1(c)) zusammen.

Zur Berechnung der Stufenwerte auf einem Level k approximieren wir U_{ni} , U_i und δ der variationellen Formulierung mit den linearen Interpolationspolynomen, die den FE-Raum $\mathcal{V}_{n+1,k}$ aufspannen. Auf einem Verfeinerungslevel mit N_k Gitterknoten gilt der

Interpolationsansatz

$$\begin{aligned} U_{ni} &\approx \sum_{l=1}^{N_k} \mu_{ni,l} \phi_l \\ U_i &\approx \sum_{l=1}^{N_k} \mu_{i,l} \phi_l \\ \delta &\approx \sum_{m=1}^{N_k} \nu_{ni,m} \phi_m \end{aligned} \quad (6.32)$$

Setzt man (6.32) in die variationelle Formulierung (6.26) ein, ist deren Lösung äquivalent zur Lösung des linearen Gleichungssystems

$$\underbrace{\left(\frac{1}{\tau\gamma} \mathbf{M} + \rho\beta \cdot \mathbf{S} - \mathbf{J}_n \right)}_{=\mathbf{L}_n} U_{ni,k} = \underbrace{-\rho\beta \cdot \mathbf{S} \cdot U_{i,k} + \mathbf{R}_i - \sum_{j=1}^{i-1} \frac{c_{ij}}{\tau} \mathbf{M} \cdot U_{nj,k}}_{=\mathbf{b}_i} \quad (6.33)$$

(wir vernachlässigen hier zunächst die Randgleichungen) mit den Massen- (\mathbf{M}) and Steifigkeitsmatrizen (\mathbf{S}),

$$\mathbf{M} = I \otimes \left(\int_0^1 \phi_l \phi_m dx \right)_{l,m=1}^{N_k} \quad \mathbf{S} = I \otimes \left(\int_0^1 \partial_x \phi_l \partial_x \phi_m dx \right)_{l,m=1}^{N_k} \quad (6.34)$$

die sich aus dem Kronecker-Produkt (Symbol \otimes)[196] mit der $N_{\text{diss}} \times N_{\text{diss}}$ - Einheitsmatrix ergeben. Das Symbol \mathbf{J}_n beschreibt die Jacobi-Matrix

$$\mathbf{J}_n = \left(\mathbf{J}_n^{(p,q)} \right)_{p,q=0}^{N_{\text{diss}}-1} \quad \mathbf{J}_n^{(p,q)} = \left(\int_0^1 \frac{\partial R_n^{(p)}}{\partial u^{(q)}} \phi_l \phi_j dx \right)_{l,m=1}^{N_k} \quad (6.35)$$

wobei $\mathbf{J}_n^{(p,q)}$ den Kopplungsterm zwischen den gelösten Spezies der Indizes p und q beschreibt. Vektor

$$\mathbf{R}_i = \left(\mathbf{R}_i^{(p)} \right)_{p=0}^{N_{\text{diss}}-1} \quad \mathbf{R}_i^{(p)} = \left(\int_0^1 R^{(p)}(U_i) \phi_l dx \right)_{l=1}^{N_k} \quad (6.36)$$

steht für die diskretisierten Reaktionsterme. Die Integrale der Jacobi-Matrizen und der Reaktionsterme müssen numerisch berechnet werden. Wie verwenden dafür Newton-Cotes-Formeln[63]. Der Vektor $U_{ni,k}$ beinhaltet die gesuchten Knotenkoeffizienten der diskretisierten Stufenlösungen

$$U_{ni,k} = \left(\mu_{ni,1}^{(0)}, \dots, \mu_{ni,N_k}^{(0)}, \mu_{ni,1}^{(1)}, \dots, \mu_{ni,N_k}^{(1)}, \dots, \mu_{ni,1}^{(N_{\text{diss}}-1)}, \dots, \mu_{ni,N_k}^{(N_{\text{diss}}-1)} \right)^T \quad (6.37)$$

Es sei betont, dass für die praktische Implementierung des Algorithmus eine andere Indizierung der Knotenkoeffizienten gewählt wird. Dazu werden die Einträge des Vektors $U_{ni,k}$ anstatt nach Speziesindizes, gemäß

$$U_{ni,k} = \left(\mu_{ni,1}^{(0)}, \mu_{ni,1}^{(1)}, \dots, \mu_{ni,1}^{(N_{\text{diss}}-1)}, \mu_{ni,2}^{(0)}, \dots, \mu_{ni,2}^{(N_{\text{diss}}-1)}, \dots, \mu_{ni,N_k}^{(N_{\text{diss}}-1)}, \dots, \mu_{ni,N_k}^{(N_{\text{diss}}-1)} \right)^T \quad (6.38)$$

aufsteigend nach den Knotenindizes sortiert. Die Zeilen und Spalten in (6.33) müssen entsprechend umgeordnet werden. Eine solche Indizierung erleichtert unter anderem die Einbringung der Randgleichungen in die linearen Gleichungssysteme.

Für die Behandlung der Randwerte muss das System (6.33) erweitert und anschließend mit den Beiträgen von (6.19) und (6.29) modifiziert werden. Wir betrachten wie zuvor exemplarisch den Rand $x = 0$, welcher dem Knotenindex 1 entspricht, und erweitern den Vektor (6.38) mit den Unbekannten der Observablen und der adsorbierten Spezies des Randknotens

$$V_{ni,k} = \underbrace{\left(0_{ni,1}, w_{ni,1}^{(0)}, \dots, w_{ni,1}^{(N_{\text{ads}}^{x=0}-1)}, \mu_{ni,1}^{(0)}, \mu_{ni,1}^{(1)}, \dots, \mu_{ni,1}^{(N_{\text{diss}}-1)}, \dots \right)^T}_{Q=N_{\text{ads}}^{x=0}+N_{\text{diss}}+1} \quad (6.39)$$

Somit müssen zur Matrix \mathbf{L}_n am Rand $x = 0$ insgesamt $N_{\text{ads}}^{x=0} + 1$ Zeilen und Spalten hinzugefügt werden.

$$\left(\begin{array}{cccc} * & * & \cdots & * \\ * & * & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \cdots & \mathbf{L}_n \end{array} \right) \left. \vphantom{\begin{array}{cccc} * & * & \cdots & * \\ * & * & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \cdots & \mathbf{L}_n \end{array}} \right\} \begin{array}{l} N_{\text{ads}}^{x=0}+1 \\ \\ \\ N_{\text{ads}}^{x=0}+1 \end{array} \quad (6.40)$$

Die neuen Einträge in (6.40) ergeben sich aus den Einträgen der Rosenbrock-Diskretisierung (6.19) bzw. den Kopplungstermen der linken Seite von (6.29). Für den oberen $Q \times Q$ -Block der erweiterten Systemmatrix (6.40) folgt mit $M = N_{\text{ads}}^{x=0} - 1$ und $N = N_{\text{diss}} - 1$

$$\left(\begin{array}{cccccc} -\partial_o g_o & -\partial_{w^{(0)}} g_o & \cdots & -\partial_{w^{(M)}} g_o & -\partial_{u^{(0)}} g_o & \cdots & -\partial_{u^{(N)}} g_o \\ -\partial_o g_{w^{(0)}} & \frac{1}{\tau\gamma} - \partial_{w^{(0)}} g_{w^{(0)}} & \cdots & -\partial_{w^{(M)}} g_{w^{(0)}} & -\partial_{u^{(0)}} g_{w^{(0)}} & \cdots & -\partial_{u^{(N)}} g_{w^{(0)}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -\partial_o g_{w^{(M)}} & -\partial_{w^{(0)}} g_{w^{(M)}} & \cdots & \frac{1}{\tau\gamma} - \partial_{w^{(M)}} g_{w^{(M)}} & -\partial_{u^{(0)}} g_{w^{(M)}} & \cdots & -\partial_{u^{(N)}} g_{w^{(M)}} \\ -\rho \partial_o g_{u^{(0)}} & -\rho \partial_{w^{(0)}} g_{u^{(0)}} & \cdots & -\rho \partial_{w^{(M)}} g_{u^{(0)}} & \cdots & \ddots & \cdots \\ \vdots & \vdots & \ddots & \vdots & \cdots & \ddots & \cdots \\ -\rho \partial_o g_{u^{(N)}} & -\rho \partial_{w^{(0)}} g_{u^{(N)}} & \cdots & -\rho \partial_{w^{(M)}} g_{u^{(N)}} & \cdots & \ddots & \cdots \end{array} \right) \left. \vphantom{\begin{array}{cccccc} -\partial_o g_o & -\partial_{w^{(0)}} g_o & \cdots & -\partial_{w^{(M)}} g_o & -\partial_{u^{(0)}} g_o & \cdots & -\partial_{u^{(N)}} g_o \\ -\partial_o g_{w^{(0)}} & \frac{1}{\tau\gamma} - \partial_{w^{(0)}} g_{w^{(0)}} & \cdots & -\partial_{w^{(M)}} g_{w^{(0)}} & -\partial_{u^{(0)}} g_{w^{(0)}} & \cdots & -\partial_{u^{(N)}} g_{w^{(0)}} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -\partial_o g_{w^{(M)}} & -\partial_{w^{(0)}} g_{w^{(M)}} & \cdots & \frac{1}{\tau\gamma} - \partial_{w^{(M)}} g_{w^{(M)}} & -\partial_{u^{(0)}} g_{w^{(M)}} & \cdots & -\partial_{u^{(N)}} g_{w^{(M)}} \\ -\rho \partial_o g_{u^{(0)}} & -\rho \partial_{w^{(0)}} g_{u^{(0)}} & \cdots & -\rho \partial_{w^{(M)}} g_{u^{(0)}} & \cdots & \ddots & \cdots \\ \vdots & \vdots & \ddots & \vdots & \cdots & \ddots & \cdots \\ -\rho \partial_o g_{u^{(N)}} & -\rho \partial_{w^{(0)}} g_{u^{(N)}} & \cdots & -\rho \partial_{w^{(M)}} g_{u^{(N)}} & \cdots & \ddots & \cdots \end{array}} \right\} \begin{array}{l} \\ \\ \\ \\ \tilde{\mathbf{L}}_n \\ \\ \end{array} \quad (6.41)$$

wobei sich die Einträge des $\tilde{\mathbf{L}}_n$ -Blocks aus den ersten N_{diss} Zeilen und Spalten der Systemmatrix \mathbf{L}_n und den entsprechenden Einträgen der Randterme (6.29) ergeben.

$$\tilde{\mathbf{L}}_n = \begin{pmatrix} l_{00} & \cdots & l_{0N} \\ \vdots & \ddots & \vdots \\ l_{N0} & \cdots & l_{NN} \end{pmatrix} - \rho \begin{pmatrix} \partial_{u^{(0)}} g_{u^{(0)}} & \cdots & \partial_{u^{(N)}} g_{u^{(0)}} \\ \vdots & \ddots & \vdots \\ \partial_{u^{(0)}} g_{u^{(N)}} & \cdots & \partial_{u^{(N)}} g_{u^{(N)}} \end{pmatrix} \quad (6.42)$$

Die Einbringung der Randgleichungen auf der rechten Seite \mathbf{b}_i des Gleichungssystems (6.33) erfolgt auf ähnliche Weise. Zunächst wird \mathbf{b}_i wieder mit $N_{\text{ads}}^{x=0} + 1$ Zeilen erweitert und den Randgleichungen (6.19) und (6.29) entsprechend verändert. Für die oberen Q Zeilen folgt

$$\left. \begin{pmatrix} g_o(V_{i,k}) \\ g_{w^{(0)}}(V_{i,k}) - \sum_{j=1}^{i-1} \frac{c_{ij}}{\tau} W_{nj,k}^{(0)} \\ \vdots \\ g_{w^{(M)}}(V_{i,k}) - \sum_{j=1}^{i-1} \frac{c_{ij}}{\tau} W_{nj,k}^{(M)} \\ \mathbf{b}_i \end{pmatrix} \right\} N_{\text{ads}}^{x=0} + 1 \quad (6.43)$$

mit den modifizierten, ersten N_{diss} Zeilen von \mathbf{b}_i

$$\tilde{\mathbf{b}}_i = \begin{pmatrix} b_0 \\ \vdots \\ b_N \end{pmatrix} + \rho \begin{pmatrix} g_{u^{(0)}}(V_{i,k}) \\ \vdots \\ g_{u^{(N)}}(V_{i,k}) \end{pmatrix} \quad (6.44)$$

Die Einbringung der Randsysteme anderer Ränder erfolgt auf analoge Weise. Für $V\mathcal{Q}$ müssen an den Randknoten in den obigen Ausdrücken die zur Variablen o gehörenden Zeilen und Spalten entfernt werden (gestrichelte Linien).

Die modifizierten linearen Gleichungssysteme (6.33) werden auf jeder Stufe eines Zeitschrittes durch ein numerisches Verfahren gelöst. Wie bereits in der Einleitung dieses Kapitels erwähnt, eignen sich für 1D-Modelle direkte Verfahren. Wir verwenden daher in der Implementierung des Algorithmus die LU-Zerlegung, welche von der GMM++-Bibliothek[197] angeboten wird. Die Systemmatrix \mathbf{L}_n muss dabei in jedem Zeitschritt nur einmal zerlegt werden.⁹

Wir werden im Folgenden, anstelle der variationellen Formulierung (6.26) bzw. der linearen Gleichungssysteme der Form (6.33) mit eingebrachten Randgleichungen, die kompaktere Formulierung

$$(L_{n,k} V_{ni,k}, \delta) = (b_{i,k}, \delta) \quad \forall \delta \in \mathcal{V}_{n+1,k} \quad i = 1, \dots, s \quad (6.45)$$

wählen, mit den diskreten Stufenlösungen $V_{ni,k} \in \mathcal{V}_{n+1,k}$. Aus (6.13) folgen die voll-diskretisierten Lösungen eines Rosenbrock-Schrittes gemäß

$$\begin{aligned} v_{n+1,k} &= v_{n,k} + \sum_{i=1}^s m_i V_{ni,k} \\ z_{n+1,k} &= z_{n,k} + \sum_{i=1}^s m_i \left(\frac{1}{\tau} \sum_{j=1}^i (c_{ij} - s_{ij}) V_{nj,k} + (\sigma_i - 1) z_{n,k} \right) \end{aligned} \quad (6.46)$$

⁹Die Systemmatrix \mathbf{L}_n ist schwach besetzt und besitzt durch die gewählte Indizierung zudem Bandstruktur, sodass prinzipiell auch effizientere Verfahren eingesetzt werden können (siehe z.B.[198]).

Die Lösungen des vorherigen Gitters $u_{n,k}$ und $z_{n,k}$ werden dafür auf verfeinerten Gittern interpoliert. Für die lokalen, diskreten Fehler der Zeitdiskretisierung folgt mit (6.20)

$$r_{n+1,k} = \sum_{i=1}^s (m_i - \hat{m}_i) V_{ni,k} \quad (6.47)$$

Die Verfeinerung der Gitter (6.31) wird mit Hilfe eines hierarchischen Fehlerschätzers gesteuert, der im folgenden Abschnitt besprochen wird.

6.3.2. Der hierarchische Fehlerschätzer

Zur Kontrolle des lokalen Verfeinerungsprozesses (6.31) ist eine Abschätzung des Fehlers nötig, welcher durch die FE-Diskretisierung entsteht. In jedem Rosenbrock-Schritt resultieren auf den Stufen (Index i) die Fehlerwerte

$$E_{ni,k} = V_{ni} - V_{ni,k} \quad i = 1, \dots, s \quad (6.48)$$

die sich aus der Differenz zwischen exakter Lösung V_{ni} und der FE-Approximation $V_{ni,k}$ ergeben. Die exakte Lösung ist *a priori* nicht bekannt, sodass eine sogenannte *a posteriori* Abschätzung des Fehlers $E_{ni,k}$ durchgeführt werden muss.

Ein möglicher Ansatz besteht darin, die unbekannte Lösung V_{ni} in (6.48) durch eine Approximation $\bar{V}_{ni,k}$ zu ersetzen, die aus einer FE-Interpolation mit stückweise quadratischen Polynomen berechnet wird (siehe Abbildung 6.1(b)). Sei $\mathcal{W}_{n+1,k}$ der FE-Raum, der durch die quadratischen Polynome aufgespannt wird, so erhält man

$$E_{ni,k} \approx \bar{V}_{ni,k} - V_{ni,k} \quad \text{mit } \bar{V}_{ni,k} \in \mathcal{W}_{n+1,k} \quad (6.49)$$

Die Berechnung der Fehlerwerte erfolgt dann auf jeder Stufe mit Hilfe sogenannter Residuumprobleme

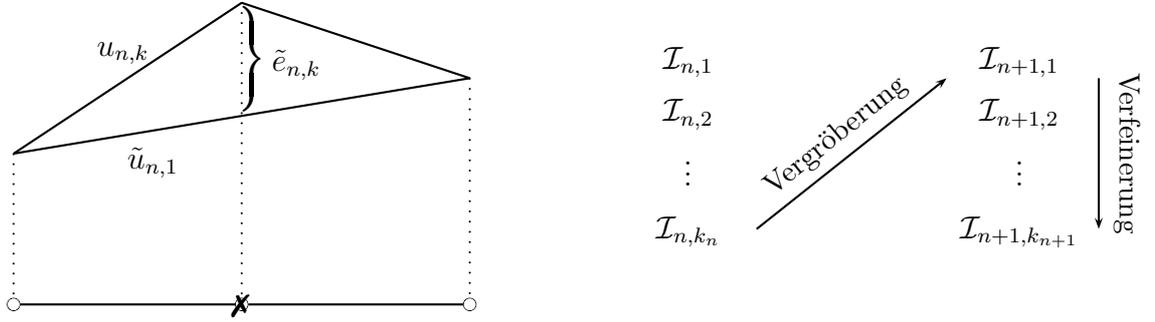
$$(L_{n,k} E_{ni,k}, \omega) = (\hat{b}_{i,k}, \omega) \quad \forall \omega \in \mathcal{W}_{n+1,k} \quad i = 1, \dots, s \quad (6.50)$$

mit

$$\hat{b}_{i,k} = b_{i,k}(E_{n1,k} + V_{n1,k}, \dots, E_{ni-1,k} + V_{ni-1,k}) - L_{n,k} V_{ni,k} \quad (6.51)$$

Die Gültigkeit der Gleichungssysteme (6.50) folgt aus (6.45) und (6.48) gemäß

$$\begin{aligned} (L_{n,k} E_{ni,k}, \omega) &= (L_{n,k}(V_{ni} - V_{ni,k}), \omega) \\ &= (L_{n,k} V_{ni}, \omega) - (L_{n,k} V_{ni,k}, \omega) \\ &= (b_{i,k}(V_{n1}, \dots, V_{ni-1}), \omega) - (L_{n,k} V_{ni,k}, \omega) \\ &= (b_{i,k}(E_{n1,k} + V_{n1,k}, \dots, E_{ni-1,k} + V_{ni-1,k}), \omega) - (L_{n,k} V_{ni,k}, \omega) \\ &= (\hat{b}_{i,k}, \omega) \end{aligned}$$



(a) Fehler $\tilde{e}_{n,k} = u_{n,k} - \tilde{u}_{n,1}$, der durch die Vergrößerung zweier Elemente entsteht.

(b) Dynamische Gitterevolution im Zeitschritt $t_n \rightarrow t_{n+1}$ (vgl. Abbildung V.6. in[60])

Abbildung 6.2: Fehler $\tilde{e}_{n,k}$, der durch die Vergrößerung $\mathcal{I}_{n,k_n} \rightarrow \mathcal{I}_{n+1,1}$ entsteht.

Der vorhergesagte Diskretisierungsfehler $e_{n+1,k}$ der Lösung $v_{n+1,k} \in \mathcal{V}_{n+1,k}$ ergibt sich schließlich aus den Stufenlösungen der Residuumprobleme

$$e_{n+1,k} = \tilde{e}_{n,k} + \sum_{i=1}^s m_i E_{ni,k} \quad (6.52)$$

Der Fehler

$$\tilde{e}_{n,k} = u_{n,k} - \tilde{u}_{n,1} \quad (6.53)$$

entsteht durch eine Gittervergrößerung $\mathcal{I}_{n,k_n} \rightarrow \mathcal{I}_{n+1,1}$ vom optimalen Gitter \mathcal{I}_{n,k_n} des vorherigen Zeitschrittes zum Anfangsgitter $\mathcal{I}_{n+1,1}$ des Zeitschrittes $t_n \rightarrow t_{n+1}$ (siehe Abbildung 6.2).

Die Berechnung der Stufenwerte $E_{ni,k}$ mit Hilfe der Residuumprobleme (6.50), die aus einer quadratischen FE-Interpolation entstehen wäre sehr aufwendig. In jedem Rosenbrock-Schritt müssten neben den Gleichungssystemen (6.45) zusätzlich die globalen Residuumprobleme (6.50) zur Fehlerapproximation gelöst werden.

Einen effizienteren Weg bietet die Verwendung einer *hierarchischen Basis*[173, 182, 192]. Anstelle des quadratischen FE-Raums $\mathcal{W}_{n+1,k}$, verwenden wir einen äquivalenten Raum $\bar{\mathcal{V}}_{n+1,k}$, welcher sich aus den ursprünglichen, linearen Interpolationspolynomen und quadratischen *Bubble*-Funktionen ergibt (siehe Abbildung 6.1(c)). Der Raum $\bar{\mathcal{V}}_{n+1,k}$ kann als die direkte Summe (Symbol \oplus [199])

$$\bar{\mathcal{V}}_{n+1,k} = \mathcal{V}_{n+1,k} \oplus \mathcal{Z}_{n+1,k} \quad (6.54)$$

der Vektorräume $\mathcal{V}_{n+1,k}$ und $\mathcal{Z}_{n+1,k}$ betrachtet werden, wobei Letzterer durch die Bubble-Funktionen gebildet wird. Für ein Gitter $\mathcal{I}_{n+1,k}$, das sich aus $N_{n+1,k}$ finiten Elementen

S^e zusammensetzt (im Folgenden soll ein hochgestelltes e den Index eines Elements beschreiben)

$$\mathcal{I}_{n+1,k} = \bigcup_{e=1}^{N_{n+1,k}} S^e \quad (6.55)$$

ergeben sich dann *lokale* Fehlerwerte $E_{ni,k}^e$ als Lösungen kleiner, lokaler Residuump Probleme[192]

$$\begin{aligned} (L_{n,k}^e E_{ni,k}^e, z) &= (\hat{r}_{i,k}^e, z) \quad \text{in } S^e \quad \forall z \in \mathcal{Z}_{n+1,k} \\ E_{ni,k}^e &= 0 \quad \text{auf } \Gamma^e \quad i = 1, \dots, s \end{aligned} \quad (6.56)$$

für jedes Element des Gitters. Die Symbole $L_{n,k}^e$ und $\hat{r}_{i,k}^e$ beschreiben dabei die lokale Systemmatrix und rechte Seite eines Elementes. Da die Bubble-Funktionen auf den Rändern der Elemente (Γ^e) verschwinden, unterscheiden sich dort die Lösungen des linearen und die des hierarchischen FE-Ansatzes nur wenig. Der Fehler an den Randknoten wird daher durch das Einbringen homogener Dirichlet-Randbedingungen in (6.56) auf Null gesetzt. Zur Berechnung der Fehler $E_{ni,k}^e$ muss somit für jedes Element pro Stufe lediglich ein kleines lineares Gleichungssystem gelöst werden. Schließlich erhält man für jedes Element die lokalen Fehlergrößen

$$\eta_{n+1,k}^e = \|\tilde{e}_{n,k}^e + \sum_{i=1}^s m_i E_{ni,k}^e\|_{L^2(S^e)} \quad (6.57)$$

(zur Norm $\|\cdot\|_{L^2(S^e)}$ siehe Abschnitt 6.4). Für den Gesamtfehler der FE-Ortsdiskretisierung ϵ verwenden wir

$$\epsilon_{n+1,k} = \|\|\eta_{n+1,k}\|\| \quad (6.58)$$

mit einer zu (6.21) kompatiblen Norm $\|\|\cdot\|\|$ (siehe Abschnitt 6.4). In (6.58) beschreibt $\eta_{n+1,k}$ eine $N_{\text{diss}} \times N_{n+1,k}$ -Matrix, welche die Fehlergrößen aller gelösten Spezies auf den Elementen des Gitters $\mathcal{I}_{n+1,k}$ enthält.

6.4. Fehlerskalierung

Der in den vorherigen Abschnitten 6.2 und 6.3 beschriebene voll-adaptive Algorithmus liefert Abschätzungen an die absoluten Fehlerwerte, die zum einen durch die Rosenbrock-Zeitdiskretisierung, zum anderen durch die Diskretisierung des Ortes mittels finiter Elemente entstehen.

Aus Gleichung (6.47) erhalten wir für jede Lösungsvariable (Index i) den absoluten Zeitfehler an jedem Knotenpunkt (Index j) des FE-Gitters ($r_j^{(i)}$).¹⁰ Die Kontrolle des Fehlers

¹⁰Die Indizes n und k aus (6.47) werden im Folgenden zur besseren Übersicht weggelassen.

der Variablen eines Knotens

$$|r_j^{(i)}| \leq \text{ATOL}, \quad \text{bzw.} \quad \frac{|r_j^{(i)}|}{\text{ATOL}} \leq 1 \quad (6.59)$$

mit einer festen absoluten Fehlertoleranz ATOL ist aufgrund möglicher Größenunterschiede in den einzelnen Komponenten der Lösung v nicht sinnvoll. Falls möglich, sollte daher der relative Fehler in den Größen kontrolliert werden

$$\frac{|r_j^{(i)}|}{|v_j^{(i)}|} \leq \text{RTOL}, \quad \text{bzw.} \quad \frac{|r_j^{(i)}|}{\text{RTOL} \cdot |v_j^{(i)}|} \leq 1 \quad (6.60)$$

wobei RTOL die relative Fehlertoleranz beschreibt. Eine rein relative Fehlermessung scheitert jedoch für $v_j^{(i)} \approx 0$, sodass in der Literatur oft eine gemischte Form

$$\frac{|r_j^{(i)}|}{\text{ATOL}^{(i)} + \text{RTOL}^{(i)} \cdot |v_j^{(i)}|} \quad (6.61)$$

aus absoluter und relativer Fehlerskalierung eingesetzt wird [60, 185, 186, 195, 200]. Durch die Wahl der einzelnen Toleranzwerte kann die Skalierung der Fehler für jede Variable eingestellt werden.

Wir wollen hier an jedem Knotenpunkt des FE-Gitters den maximalen Fehler gemäß (6.61) beschränken, wählen also für den Zeitfehler (6.21) die Bedingung

$$\varepsilon = \|r\| = \max_i \max_j \left(\frac{|r_j^{(i)}|}{\text{ATOL}^{(i)} + \text{RTOL}^{(i)} \cdot |v_j^{(i)}|} \right) \leq 1 \quad (6.62)$$

Im Zusammenhang voll-adaptiver Algorithmen zur Lösung von zeitabhängigen PDEs wird angenommen, dass sich der Gesamtfehler der Simulation additiv aus den Zeit- und Ortsfehlern zusammensetzt. Für das Erreichen einer Gesamt toleranz TOL verwendet man daher häufig eigene Toleranzwerte für die Zeit- (TOL_t) und Ortsdiskretisierung (TOL_x), die über die Beziehungen

$$\text{TOL}_t = \text{TOL}/n \quad \text{und} \quad \text{TOL}_x = \text{TOL}/m \quad (6.63)$$

ausbalanciert werden [60, 143, 184]. Die Parameter n und m werden heuristisch bestimmt (z.B. $n = m = 2$ in [143, 184] oder $n = 2, m = 3$ in [60]). Wir wollen im Folgenden eine andere Strategie vorstellen, welche es ermöglicht, aus den an den Knoten vorgegebenen relativen Toleranzen der Zeitfehlerskalierung (6.61), entsprechende Toleranzwerte für die Skalierung des Ortsfehlers zu ermitteln. Dazu betrachten wir die relativen Toleranzen als (konstante) Funktionen in Zeit- und Ortskoordinate.

$$\text{RTOL}^{(i)} = \text{RTOL}^{(i)}(t, x) \quad (6.64)$$

Der Algorithmus liefert für jedes Element des FE-Gitters (Index e) und für jede Volumenkonzentration (Index i) die Fehlergrößen $\eta_e^{(i)}$, die sich nach Gleichung (6.57) aus den Lösungen der Residuump Probleme (6.56) ergeben. In Gleichung (6.57) steht $\|\cdot\|_{L^2(S^e)}$ für die L^2 -Norm¹¹ auf dem Element S^e

$$\|f\|_{L^2(S^e)} = \left(\int_{S^e} f^2 dx \right)^{1/2} \quad (6.65)$$

Mit

$$\tilde{\eta}_e^{(i)} = \frac{\eta_e^{(i)}}{\text{ATOL}^{(i)} + \text{RTOL}_e^{(i)} \cdot \|u^{(i)}\|_{L^2(S^e)}} \quad (6.66)$$

erhält man für die skalierten η -Werte ($\tilde{\eta}_e^{(i)}$) einen zu (6.61) äquivalenten Ausdruck. Die relativen Fehlertoleranzen eines Elementes $\text{RTOL}_e^{(i)}$ ergeben sich darin als L^2 -Norm der Toleranzfunktion (6.64)

$$\text{RTOL}_e^{(i)} = \|\text{RTOL}^{(i)}\|_{L^2(S^e)} = \left(h^e \cdot (\text{RTOL}^{(i)})^2 \right)^{1/2} \quad (6.67)$$

Die Werte $\text{RTOL}_e^{(i)}$ können somit als ein Äquivalent der Knotentoleranzen $\text{RTOL}^{(i)}$ betrachtet werden. Insgesamt wird der Ortsfehler (6.58) in der Maximumsnorm

$$\epsilon = \|\eta\| = \max_i \max_e (\tilde{\eta}_e^{(i)}) \leq 1 \quad (6.68)$$

beschränkt. Der Algorithmus lässt sich folglich alleine über die Vorgabe der Toleranzen $\text{ATOL}^{(i)}$, $\text{RTOL}^{(i)}$ kontrollieren. Eine zusätzliche, heuristische Annahme der Form (6.63) ist nicht mehr nötig.

Neben dieser Vereinfachung wird die Fehlermessung, im Vergleich zu[184], zudem mit anderen, restriktiveren Normen durchgeführt. Insbesondere für Modellsysteme, in denen lokal große Konzentrationsveränderungen auftreten, zeigen die Algorithmen nun ein stabileres Verhalten.¹² Die Vermutung liegt nahe, dass die Kontrolle der Zeitschrittweiten auf Basis der einzelnen Knotenwerte (Gleichung (6.62)) entscheidend zur verbesserten Stabilität beiträgt. In[184] wurde zur Kontrolle des Zeitfehlers stattdessen eine skalierte euklidische Norm verwendet, die jedoch große, lokal begrenzte Fehler weniger stark bewertet (siehe dazu[138]).

¹¹Zur Berechnung der L^2 -Norm verwenden wir für hierarchische Elemente

$$\int_{S^e} f^2 dx \approx \frac{h^e}{3} (f_0^2 + 4f_2^2 + f_1^2)$$

und für lineare Elemente

$$\int_{S^e} f^2 dx \approx \frac{h^e}{2} (f_0^2 + f_1^2)$$

die entsprechenden Newton-Cotes-Formeln (zur Knotennummerierung siehe Abbildungen 6.1(a) und (c))[63].

¹²Zuvor wurden Störungen in den numerischen Lösungskurven beobachtet[184].

```

1   $n = 0, t_n = 0, \tau_{n+1} = \tau_{\text{init}}$ 
2   $\mathcal{I}_{n+1,1} = \mathcal{I}_{\text{init}}$ 
3  solange  $t_n < 1$ 
4     $k = 1$ 
5    berechne  $v_{n+1,k}, r_{n+1,k}, \tilde{\eta}_{n+1,k}^e, \epsilon_{n+1,k}$ 
6      solange  $\epsilon_{n+1,k} > 1$  und  $k < k_{\text{max}}$ 
7        verfeinere alle Elemente mit  $\tilde{\eta}_{n+1,k}^e > \tilde{\eta}_{n+1,k}^{\text{barr}}$ 
8         $k = k + 1$ 
9        berechne  $v_{n+1,k}, r_{n+1,k}, \tilde{\eta}_{n+1,k}^e, \epsilon_{n+1,k}$ 
10   berechne  $\epsilon_{n+1}$ 
11   falls  $\epsilon_{n+1} \leq 1.0$ 
12     akzeptiere Lösung
13     falls  $\vartheta \cdot \epsilon_{n+1,k} < 1$ 
14       vergrößere alle Elemente mit  $2 \cdot \tilde{\eta}_{n+1,k}^e < \tilde{\eta}_{n+1,k}^{\text{barr}}$ 
15      $t_{n+1} = t_n + \tau_{n+1}, n = n + 1$ 
16      $\mathcal{I}_{n+1,1} = \mathcal{I}_{n,k_n}^{\text{coarsed}}$ 
17     bestimme  $\tau_{n+1}$ 
18   sonst
19     verwerfe Lösung
20     rekonstruiere  $\mathcal{I}_{n+1,1}$ 
21     reduziere  $\tau_{n+1}$ 

```

Abbildung 6.3: Übersicht über den Multilevel-Algorithmus, implementiert in den MFEMSolver-Klassen (siehe Abschnitt 6.6).

6.5. Gitterrevolution

Der Verfeinerungsprozess zusammen mit den Gittervergrößerungen (siehe Abbildung 6.2(b)) wird über die skalierten lokalen Fehlergrößen $\tilde{\eta}$ aus Gleichung (6.66) gesteuert. Abbildung 6.3 zeigt das Prinzip des gesamten Lösungsalgorithmus. Beginnend mit der Lösung $v_{n+1,1}$, wird mit Hilfe der dazugehörigen $\eta_{n+1,1}^e$ -Werte das grobe Gitter $\mathcal{I}_{n+1,1}$ sukzessive verfeinert, bis entweder die Fehlertoleranz oder eine maximale Anzahl an Verfeinerungsleveln k_{max} erreicht ist (Zeilen 5 – 9). Dazu wird für jeden Gitterlevel k eine Fehlerschranke

$$\tilde{\eta}_{n+1,k}^{\text{barr}} = \gamma \cdot \epsilon_{n+1,k} \quad (6.69)$$

berechnet, welche durch den Parameter $0 < \gamma < 1$ angepasst werden kann. Während der Gitterverfeinerung werden dann nur die Elemente unterteilt, die einen $\tilde{\eta}$ -Wert besitzen, der größer als $\tilde{\eta}^{\text{barr}}$ ist (Zeile 7).

Nachdem eine ausreichend genaue Lösung auf einem Gitter \mathcal{I}_{n+1,k_n} bestimmt wurde,

wird in Zeile 10 der Zeitfehler gemäß Gleichung (6.21) berechnet. Die Lösung wird akzeptiert, falls die Bedingung (6.62) erfüllt ist. Nun kann durch eine Vergrößerung des Anfangsgitters des nächsten Zeitschrittes bestimmt werden. Dazu heben wir die Unterteilung der Elemente auf, für die das Zweifache ihres $\tilde{\eta}$ -Wertes immer noch kleiner ist als $\tilde{\eta}^{\text{barr}}$ (Zeile 14). Diese Bedingung ist durch die Konvergenzeigenschaften einer linearen FE-Interpolation motiviert[60] und soll gewährleisten, dass nur die Elemente vergrößert werden, die im nächsten Zeitschritt nicht direkt wieder verfeinert werden müssen. Die Vergrößerung findet nur dann statt, falls zusätzlich gilt

$$\vartheta \cdot \epsilon_{n+1,k} < 1 \tag{6.70}$$

mit einem Parameter $\vartheta \geq 1$ (Zeile 13). Der ϑ -Wert bestimmt die Empfindlichkeit der Vergrößerungsschritte. Für $\vartheta = 1$ wird nach jedem akzeptierten Zeitschritt vergrößert, für einen größeren ϑ -Wert nur bei entsprechend kleinem Ortsfehler.

Falls der Zeitfehler die Toleranz nicht erfüllt, wird das ursprüngliche, grobe Gitter des Zeitschrittes wiederhergestellt und ein neuer Verfeinerungsprozess mit reduzierter Schrittweite gestartet (Zeilen 18 – 21).

Jede Veränderung des Gitters führt zu der Durchführung einer sogenannten *Massenbilanzierung*[157]

$$\frac{1}{\mu} \leq \frac{h_{e-1}}{h_e} \leq \mu \tag{6.71}$$

für ein festes $\mu \geq 1$, wobei h_e die Länge eines Elementes des Gitters beschreibt. Sie verhindert große Unterschiede in den Längen benachbarter Elemente, die zu schlechten Lösungseigenschaften der Gleichungen führen können.

Es sei erwähnt, dass durch die Verwendung der Schranke k_{max} für die Gitterverfeinerung (Zeile 6) der Fall eintreten kann, in dem der Ortsfehler die Bedingung (6.68) nicht erfüllt, während für den Zeitfehler $\epsilon_{n+1} \leq 1$ gilt. Wir wollen in diesem Fall trotzdem die Lösung akzeptieren. Es stellte sich heraus, dass dieser Fall für eine Begrenzung von $k_{\text{max}} = 10$ zum einen eher selten eintritt, zum anderen keine erkennbaren Auswirkungen auf die Lösungen zu beobachten waren. Der Anwender wird vom Algorithmus durch eine Warnung informiert.

6.6. Objekt-orientiertes Design und Implementierung

Für die Implementierung der in den Abschnitten (6.2) – (6.5) vorgestellten Verfahren verwenden wir erneut ein objekt-orientiertes Design, welches auf einem von Gamma et al.[27] beschriebenen Entwurfsmuster basiert (Abschnitt 6.6.1). Die Algorithmen werden anschließend (Abschnitt 6.6.2) an die Geschwindigkeitsgesetze von Ecco (siehe Kapitel 4) gekoppelt, wodurch die Simulation aller derzeit von Ecco unterstützter elektrochemischer Modelle ermöglicht wird (Kapitel 3).

6.6.1. Das Strategie-Muster

Durch die Koeffizienten in den Formeln (6.13) – (6.17) werden verschiedene Rosenbrock-Methoden definiert, die in Kombination mit dem Multilevel-FE-Prozess zur Simulation elektrochemischer Modelle untersucht werden (Kapitel 7). Tabelle 6.1 zeigt die Rosenbrock-Verfahren, welche in dieser Arbeit implementiert wurden. Es sei betont, dass noch zahlreiche weitere Rosenbrock-Schemata existieren, die zur Simulation elektrochemischer Modelle eingesetzt werden könnten (siehe z.B.[201]).

Um die verschiedenen Verfahren auf elegante Weise in eine Modellierung und Simulations-Software integrieren zu können, verwenden wir das sogenannte *Strategie*-Muster[27].

Dazu wurde die Klassenhierarchie aus Abbildung 6.4 gebildet. Jede Unterklasse steht für eines der Rosenbrock-Verfahren aus Tabelle 6.1 und implementiert in der `step()`-Methode den Zeitschritt gemäß den Formeln (6.13) – (6.17). Zudem werden die lokalen Zeitresiduen (6.20) und die η -Werte (6.57) berechnet. Die Koeffizienten der Rosenbrock-Verfahren werden in den jeweiligen Unterklassen gespeichert und zur Berechnung der Lösung in der `step()`-Methode eingesetzt. Die dabei zu lösenden Gleichungssysteme hängen insbesondere von dem zugrunde liegenden FE-Gitter ab. Ändert sich die Gitterstruktur, müssen die davon abhängigen Daten (z.B. die Massen- und Steifigkeitsmatrizen (6.34)) aktualisiert werden. Dies wird durch eine `updateMeshData()`-Funktion implementiert.

Alle Rosenbrock-Unterklassen aus Abbildung 6.4 besitzen die Schnittstelle der abstrakten Basisklasse `RotheRosenbrock`. Sie können somit entweder *dynamisch* zur Laufzeit des Programms ausgetauscht¹³ oder *statisch* als sogenannte *Policy*-Klassen in C++-Klassentemplates eingesetzt werden[27].

Die dynamische Variante besitzt den Nachteil, dass in jedem Zeitschritt der Simulation der Aufruf einer `step()`-Funktion über die rein-virtuelle Funktion der Basisklasse erfolgt. Dies kann bei Simulationen, in denen viele Zeitschritte benötigt werden, zu relevanten

¹³Es besteht somit sogar prinzipiell die Möglichkeit, während einer Simulation einzelne Zeitschritte auf elegante Weise von unterschiedlichen Verfahren berechnen zu lassen.

Tabelle 6.1: Ausgewählte Rosenbrock-Methoden (siehe[60] und darin enthaltenen Referenzen). Die Symbolik $p(\hat{p})$ beschreibt die Konvergenzordnung p der Methode und die des eingebetteten Verfahrens, \hat{p} (siehe Abschnitt 6.2.2).

Methode	Stufenanzahl	Funktionsauswertungen	Konvergenzordnungen
Ros2	2	2	2(1)
Rowda3	3	2	3(2)
Ros3P	3	2	3(2)
RodasP	6	6	4(3)

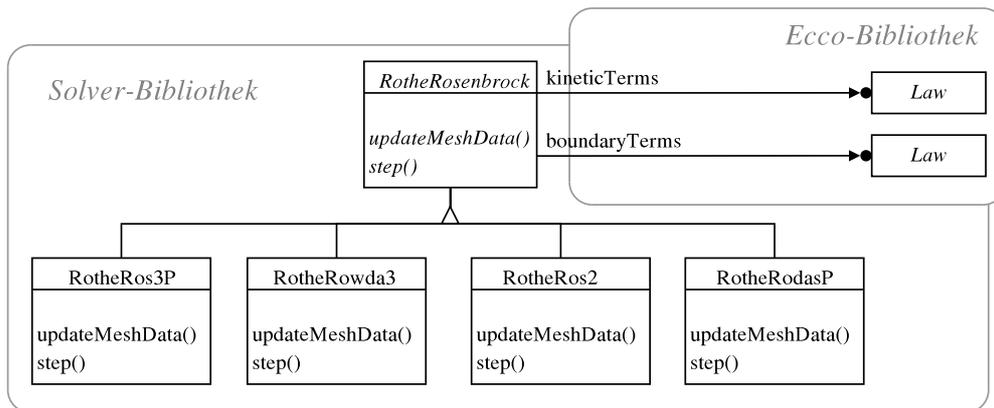


Abbildung 6.4: Rothe-Rosenbrock-Klassen (*Strategie-Muster*); jede Unterklasse repräsentiert ein spezifisches Rosenbrock-Schema. Die Kopplung an Ecco geschieht in der abstrakten Oberklasse. Die Klassen des Algorithmus sind in einer sogenannten *Solver-Bibliothek* zusammengefasst.

Zeitverlusten führen. Ein effizienteres Verhalten verspricht die statische Variante, in der die `step()`-Methode direkt auf den konkreten Unterklassen aufgerufen wird, wodurch das dynamische Binden entfällt.

Abbildung 6.5 zeigt das zugrunde liegende Klassendiagramm. Das `MFEMSolver`-Klassen-Template implementiert den Algorithmus aus Abbildung 6.3 in der `solve()`-Methode und erhält den speziellen Rosenbrock-Typ als Template-Parameter (als 'class R' wird eine Unterklasse aus Abbildung 6.4 erwartet). Zur Berechnung eines Zeitschrittes kommt dann in der `solve()`-Methode das gewählte Rosenbrock-Verfahren zum Einsatz.

Das statische Binden der Rosenbrock-Strategie zur Übersetzungszeit des Algorithmus garantiert einen effizienten Funktionsaufruf auf den Objekten. Um den dynamischen Austausch unterschiedlicher Verfahren zur Laufzeit eines Anwender-Programms zu ermöglichen, wird das `MFEMSolver`-Template von einer abstrakten `AFEMSolver`-Klasse abgeleitet. Ein Anwender-Code kann somit z.B. durch die Zeilen

```

AFEMSolver* solver;
...
// falls Ros3P verwendet werden soll
solver = new MFEMSolver<RotheRos3P,...>(...);

// falls Rowda3 verwendet werden soll
solver = new MFEMSolver<RotheRowda3,...>(...);
...
// starte Algorithmus
solver->solve();

```

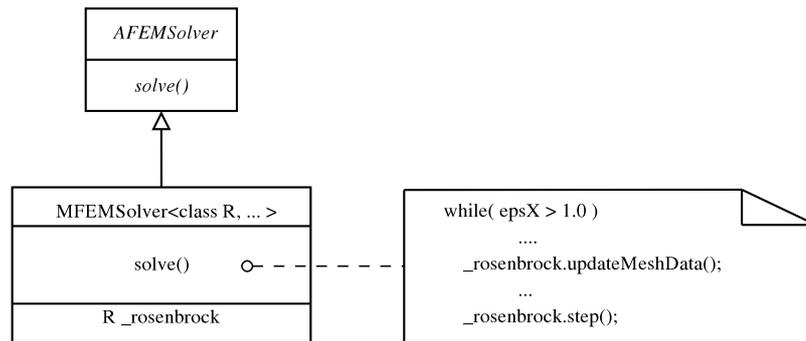


Abbildung 6.5: MFEMSolver-Klassen-Template. Die `solve()`-Methode implementiert den Multilevel-Algorithmus unter Verwendung der als Template-Parameter angegebenen Rosenbrock-Strategie. Die Pseudo-Implementierung zeigt einen Auszug aus dem Algorithmus (Abbildung 6.3, Zeilen 6 – 9).

die mit der gewünschten Rosenbrock-Strategie konfigurierten MFEMSolver-Objekte erzeugen. Der Aufruf der `solve()`-Methode startet anschließend die Simulation.

Abbildung 6.6 zeigt ein Interaktionsdiagramm, welches den Ablauf des Algorithmus in Form von Methodenaufrufen auf einigen Objekten verdeutlicht. Als Beispiel ist ein MFEMSolver-Objekt dargestellt, welches mit der `RotheRos3P`-Klasse konfiguriert wurde. Zudem wird zur Schrittweitenkontrolle eine Klasse eingesetzt, die auf der von Lang angegebenen Formel basiert (siehe Gleichung (6.22)). Eine Mesh-Klasse repräsentiert das FE-Gitter.

Zu Beginn werden zunächst alle relevanten Daten initialisiert. Dabei wird z.B. die Größe der Lösungsvektoren (6.39) bestimmt, die von der Anzahl der Elemente des Gitters abhängt. Letztere wird über die `nbElements()`-Methode der Mesh-Klasse abgefragt.

Nachdem auf dem größten Gitter eines Zeitschrittes eine erste Lösung berechnet wurde, wird, falls nötig, das Gitter verfeinert und eine neue Lösung bestimmt, bis der Ortsfehler die vorgegebene Toleranz erreicht hat. Falls zudem die Toleranz in der Zeit erfüllt ist, wird die Lösung akzeptiert und durch den Aufruf der `accept()`-Methode eine neue Zeitschrittweite berechnet. Das Gitter wird vergrößert und ein neuer Zeitschritt gestartet. Bei zu großem Zeitfehler verwirft der MFEMSolver die Lösung, berechnet mittels der `reject()`-Methode eine reduzierte Schrittweite, rekonstruiert das Gitter und startet den Zeitschritt erneut.

Die Gittermanipulationen verlangen nach einer Datenstruktur, welche auf effiziente Weise Verfeinerungen, Vergrößerungen sowie die Rekonstruktion alter Gitterstrukturen unterstützt. Für die Implementierung der Mesh-Klasse wurde daher ein STL-ähnlicher Baum-Datencontainer eingesetzt[202].

Es sei erwähnt, dass die MFEMSolver-Template-Klasse neben den Rosenbrock- und Schrittweiten-Policies mit Hilfe weiterer Template-Parameter konfiguriert werden muss.

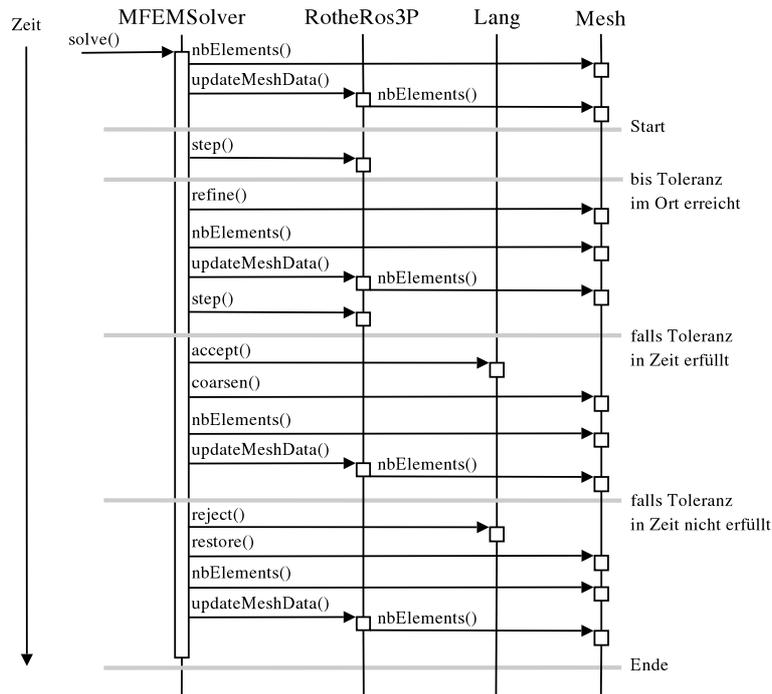


Abbildung 6.6: Interaktionsdiagramm des Algorithmus. Nur einzelne, ausgewählte Funktionsaufrufe werden gezeigt.

So dienen sogenannte *Processor*-Klassen zur Datenaufnahme und in $V2$ zur Stromberechnung. *Observer*-Klassen ermöglichen eine dynamische Visualisierung der Ergebnisse. *Sweep*- und *Step*-Klassen dienen zur Modifizierung der Schrittweitenkontrolle für komplexe Anregungsfunktionen (siehe Abschnitt 7.1.2).

6.6.2. Kopplung an Ecco

Der elektrochemische Compiler Ecco übersetzt einen durch den Anwender definierten elektrochemischen Reaktionsmechanismus in C++-Objekte (siehe Kapitel 4). Insbesondere werden Objekte gebildet, die die Reaktions- und Randgleichungen der PDAEs aus Abschnitt 6.1 repräsentieren. In den voll-diskretisierten Gleichungen (6.33) treten die Reaktionsterme in den Komponenten der diskreten Jacobi-Matrix (6.35) und des Reaktionsvektors (6.36) auf. Die Randgleichungen müssen in der modifizierten Systemmatrix (6.41) und rechten Seite (6.43) berücksichtigt werden.

Innerhalb der Ecco-Bibliothek werden die Reaktions- und Randterme durch die Law-Klassenhierarchie aus Abbildung 4.11 realisiert und bieten die in Abschnitt 4.3.4 beschriebene Schnittstelle zur Funktionsauswertung an. Durch die Ankopplung der Law-Klassen an die RotheRosenbrock-Klassen aus Abbildung 6.4 ist es möglich, alle von Ecco

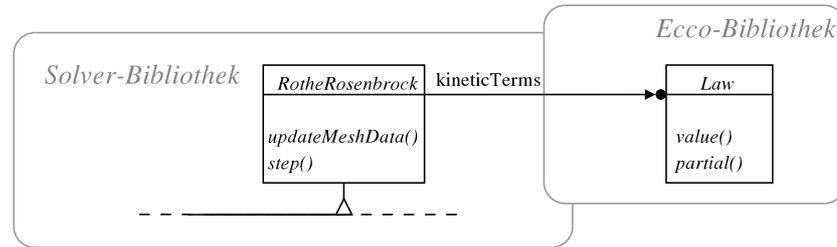


Abbildung 6.7: Kopplung der abstrakten RotheRosenbrock-Klasse an die Law-Klassenschnittstelle der Reaktionsterme.

unterstützten Mechanismen zu simulieren, ohne den Code des zuvor beschriebenen Algorithmus zu verändern. Die Auswertungen der Reaktions- und Randterme geschieht dabei in der Basisklasse über die Law-Klassenschnittstelle des Ecco-Compilers. Als Beispiel betrachten wir einen EC-Mechanismus



in dem ein VET und anschließend eine irreversible Folgereaktion des Produkts stattfindet. Für den Reaktionsterm des Mechanismus gilt

$$R(u) = \begin{pmatrix} R^{(0)}(u) \\ R^{(1)}(u) \\ R^{(2)}(u) \end{pmatrix} = \begin{pmatrix} 0 \\ -k_{f1} \cdot u^{(1)} \\ k_{f1} \cdot u^{(1)} \end{pmatrix} \equiv \begin{bmatrix} \text{Law*} \\ \text{Law*} \\ \text{Law*} \end{bmatrix} = \text{kineticTerms} \quad (6.73)$$

wobei $u = (u^{(0)}, u^{(1)}, u^{(2)})$ den Konzentrationsvektor der Spezies A, B und C, und k_{f1} die Geschwindigkeitskonstante der Folgereaktion beschreiben. In der C++-Implementierung wird der Reaktionsterm durch den kineticTerms-Vektor dargestellt, dessen Einträge den Zugriff auf die zusammengesetzten Law-Objekte ermöglicht. Zur Berechnung der Einträge des Reaktionsvektors (6.36) werden dann die value()-Funktionen der Objekte

```
kineticTerms[0]->value(Ui)
kineticTerms[1]->value(Ui)
kineticTerms[2]->value(Ui)
```

aufgerufen, wobei der Vektor U_i die Konzentrationen eines Knotens der Stufe i enthält. Neben den Funktionswerten müssen zur Berechnung der Jacobi-Matrix (6.35) die partiellen Ableitungen

$$\partial_u R(u) = \begin{pmatrix} \frac{\partial R^{(0)}}{\partial u^{(0)}} & \frac{\partial R^{(0)}}{\partial u^{(1)}} & \frac{\partial R^{(0)}}{\partial u^{(2)}} \\ \frac{\partial R^{(1)}}{\partial u^{(0)}} & \frac{\partial R^{(1)}}{\partial u^{(1)}} & \frac{\partial R^{(1)}}{\partial u^{(2)}} \\ \frac{\partial R^{(2)}}{\partial u^{(0)}} & \frac{\partial R^{(2)}}{\partial u^{(1)}} & \frac{\partial R^{(2)}}{\partial u^{(2)}} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ -k_{f1} & 0 & 0 \\ k_{f1} & 0 & 0 \end{pmatrix} \quad (6.74)$$

ausgewertet werden. Die Law-Klassenschnittstelle bietet dafür für jeden einzelnen Eintrag der Matrix die `partial()`-Methode an

```
kineticTerms[0]->partial(0,un)
kineticTerms[0]->partial(1,un)
kineticTerms[0]->partial(2,un)

kineticTerms[1]->partial(0,un)
kineticTerms[1]->partial(1,un)
kineticTerms[1]->partial(2,un) ...
```

wobei zum einen der Index der Spezieskonzentration angegeben werden muss, nach der die Ableitung gebildet wird. Zum anderen beschreibt u_n den Konzentrationsvektor eines Knotens zum Zeitpunkt t_n .

Die Auswertung der Randterme (`boundaryTerms`, Abbildung 6.4) erfolgt auf analoge Weise.

Es sei betont, dass üblicherweise jedes numerische Verfahren zur Lösung von Differentialgleichungen einen Mechanismus zur Auswertung der Funktionswerte der rechten Seite und deren Jacobi-Matrix benötigt. Folglich lassen sich die Law-Klassen von Ecco nicht nur im Zusammenhang der hier eingesetzten Kombination aus Rosenbrock-Verfahren und FE-Methode verwenden. Sie können auch an andere Diskretisierungsmethoden gekoppelt werden.

7. Anwendungen

Die Eigenschaften der in Kapitel 6 beschriebenen Algorithmen werden im Folgenden anhand ausgewählter Modellprobleme untersucht.

Die Kopplung der Algorithmen an Ecco (Abschnitt 6.6.2) ermöglicht die Simulation aller Elektrodenprozesse, die mit Hilfe des Compilers modelliert werden können. Dazu gehören heterogene Elektronentransfers zwischen Elektroden und gelösten oder adsorbierten Substanzen, homogene chemische Reaktionen, Adsorptionen sowie Oberflächenreaktionen (siehe Abbildung 3.1).

Die Algorithmen wurden für lineare Diffusionsprozesse in 1D-Modellgeometrien implementiert. Sie können daher zur Simulation semi-infiniten und finiten Diffusionsmodelle mit planaren Arbeitselektroden eingesetzt werden.

Durch die Einbindung der *Excitation Function Template Library* (Abschnitt 3.2) sind zudem potential- und stromkontrollierte elektrochemische Experimente mit unterschiedlichen Anregungsfunktionen zugänglich.

Die folgenden Abschnitte sollen anhand einiger Anwendungsbeispiele verdeutlichen, dass die hier implementierten Algorithmen für eine Vielfalt an elektrochemischen Modellen zuverlässige Resultate liefern.

Aufgrund der nahezu unendlichen Anzahl an möglichen Kombinationen aus elektrochemischen Anregungsfunktionen, Reaktionsmechanismen und der unterstützten Modellgeometrien ist die Auswahl der Beispiele schwierig. Wir wollen uns daher insbesondere auf solche Modelle konzentrieren, für die entweder analytische oder zuverlässige numerische Lösungen in der Literatur existieren, um die Qualität der berechneten Ergebnisse beurteilen zu können.

Zudem werden Modelle simuliert, die bestimmte Herausforderungen an die numerischen Verfahren stellen. Dazu gehören z.B. unstetige Anfangs- und Randwerte, wie sie in chronoamperometrischen Experimenten vorkommen, sowie sehr steile, sich bewegende Reaktionsfronten, welche oft durch Folgereaktionen in der Diffusionsschicht entstehen.

Neben diesen in der Literatur akzeptierten Testsystemen werden wir zusätzlich dynamische Adsorptions- und Oberflächenprozesse betrachten, wie sie bisher erst vereinzelt und unter spezielleren Annahmen beschrieben wurden.

Sämtliche Simulationen wurden mit dem in dieser Arbeit entwickelten *ModSim*-Programm (Version 0.3) durchgeführt[30]. Die Möglichkeit zur Reproduktion der Ergebnisse ist somit gegeben.

7.1. Potentialkontrollierte Experimente

In der Simulation von potentialkontrollierten Experimenten liefert eine Potential-Zeit-Anregungsfunktion den Verlauf des Elektrodenpotentials, welches zur Berechnung der Randwerte des Modells benötigt wird.

Darüber hinaus wird in der *Variante 1* ($V1$, siehe Abschnitt 6.1) der Strom über das Strom-Potential-Residuum aus dem vorgegebenen Elektrodenpotential und der Konzentrationsverteilung am Rand berechnet. Das Strom-Potential-Residuum bringt dabei bezüglich des Stroms eine zusätzliche *lineare* algebraische Gleichung in das Modell ein. Diese muss durch die Rosenbrockverfahren simultan zu den PDEs gelöst werden.

In der *Variante 2* ($V2$) dagegen erfolgt die Stromberechnung in einem nachverarbeitenden Schritt durch die Approximation der Konzentrationsableitungen an der Elektrodenoberfläche.

Die Verwendung der $V1$ zur Stromberechnung stellt eine neue Alternative zu den bisher in der Literatur beschriebenen Methoden dar (siehe Abschnitt 3.11). Sie ermöglicht, im Vergleich zur klassischen $V2$, eine zusätzliche Kontrolle der Zeitschrittweiten auf der Basis einer Fehlerschätzung im Strom. Die Kontrolle der Konzentrationsfehler bleibt zudem uneingeschränkt erhalten. Der Schwerpunkt der folgenden Betrachtung liegt daher stets auf der Untersuchung der Lösungseigenschaften von $V1$.

In[184] wurden die in Abschnitt 6.6.1 nach $V2$ implementierten Rosenbrock-Methoden bereits anhand der Simulation eines einfachen Testsystems (Semi-infinite Diffusion, cyclische Voltammetrie, einfacher Elektronentransfer) verglichen. Es stellte sich heraus, dass Ros3P und RodasP dem Rowda3 und dem Ros2-Verfahren überlegen sind. Für die Behandlung partieller Differentialgleichungen ist die Überlegenheit der ersten beiden Methoden gegenüber Rowda3 und Ros2 bekannt[201]. Auch in weiteren Testsimulation potentialkontrollierter Experimente zeigte sich, dass insbesondere das Ros3P-Verfahren, welches speziell für parabolische Differentialgleichungen entwickelt wurde[190], in beiden Varianten qualitativ stets die besten Ergebnisse lieferte. Ros3P ist zudem häufig effizienter als die drei anderen Verfahren. Wir werden daher meistens die Simulationsergebnisse der *RotheRos3P*-Klasse präsentieren. Ein Vergleich der Methoden wird im ersten Beispiel diskutiert (Abschnitt 7.1.1).

Es sei bereits hier darauf hingewiesen, dass sich im Falle *stromkontrollierter* Experimente das Verhalten der vier Rosenbrock-Methoden relativ zueinander verändert. Dies wird später in Abschnitt 7.2 diskutiert.

7.1.1. *Beispiel 1: Chronoamperometrie — Einelektronentransfer*

In einem klassischen chronoamperometrischen Experiment (CA) wird ein konstantes Potential an der Arbeitselektrode vorgegeben und der resultierende Strom gemessen. Die dazugehörige Anregungsfunktion beinhaltet ein einzelnes konstantes Segment (Abbildung 7.1 (links)).

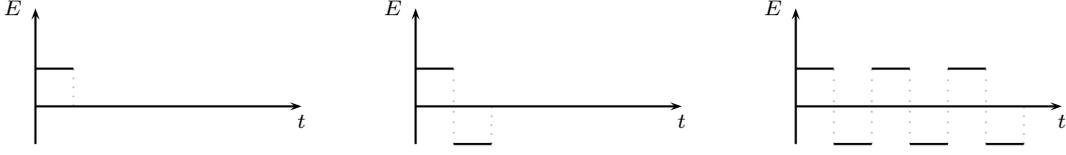


Abbildung 7.1: Skizzierte Anregungsfunktionen: CA (links), Doppel-Puls-CA (Mitte), Multi-Puls-CA (rechts).

Wir betrachten hier zunächst den einfachen Elektronentransfer



bei einer Temperatur $T = 293.15$ K und mit einer Elektrodenoberfläche von $A = 6$ mm². Als Speziesparameter verwenden wir die Anfangskonzentrationen $c_A^0 = 1 \cdot 10^{-4}$, $c_B^0 = 0$ mol · l⁻¹ und die Diffusionskoeffizienten $D_A = D_B = 1 \cdot 10^{-5}$ cm²s⁻¹. Für die Kinetik des VETs soll das Butler-Volmer-Modell mit Formalpotential $E^0 = 0.25$ V, Transferkoeffizient $\alpha = 0.5$ und Geschwindigkeitskonstante $k_h = 0.1$ cm · s⁻¹ gelten.

Bei einem Elektrodenpotential, welches weit über dem Formalpotential des Elektronentransfers liegt, wird der Strom alleine durch den Massentransfer der elektroaktiven Spezies zur Elektrode hin bestimmt und kann für reversible VETs über die *Cottrell-Gleichung*[51, 203]

$$\chi_{\text{Cottrell}}(t) = \frac{L_{\text{ref}}}{D_{\text{max}} c_{\text{max}}} \cdot \frac{D_A^{1/2} c_A^0}{\pi^{1/2} t^{1/2}} \quad (7.2)$$

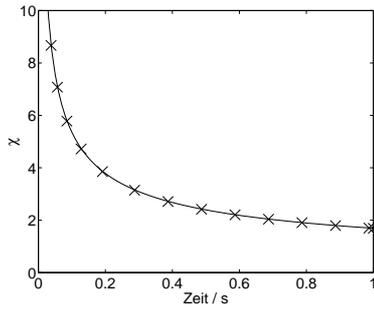
(hier in dimensionsloser Form) für $t > 0$ analytisch berechnet werden. Das dimensionslose Konzentrationsprofil der Spezies A ist durch[51, 203]

$$u_{\text{Cottrell}}^{(0)}(t, x) = \text{erf} \left(\frac{L_{\text{ref}} \cdot x}{2(D_A t)^{1/2}} \right) \quad (7.3)$$

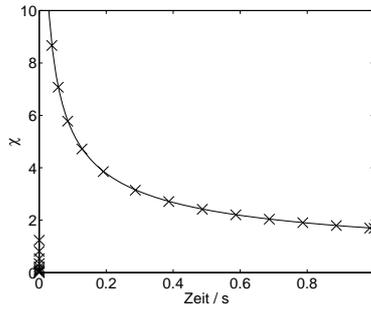
gegeben. Gleichungen (7.2) und (7.3) liefern somit in jedem Zeitschritt Referenzwerte für die simulierten Ergebnisse.

An der Stelle $t = 0$ besitzt der Stromverlauf (7.2) eine Zeit-Singularität, das Konzentrationsprofil ist für $t = 0$ und $x = 0$ unstetig. Die Simulation des Modells ist somit insbesondere in den ersten Zeitschritten zu Beginn des Experiments erschwert. Die Verlässlichkeit eines numerischen Algorithmus in Bezug auf die Simulation elektrochemischer Experimente wird daher häufig an seiner Fähigkeit gemessen, wie schnell große Fehler zu Beginn eines Cottrell-Experiments in den folgenden Zeitschritten gedämpft werden können (siehe z.B. die Diskussion in[204–206]).

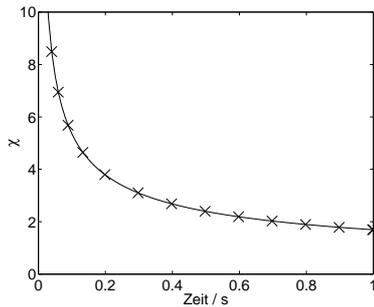
Abbildung 7.2 zeigt die simulierten Ergebnisse der vier Rosenbrock-Klassen aus Abbildung 6.4 für beide Varianten des Simulationsmodells. Als Anregungsfunktion wurde



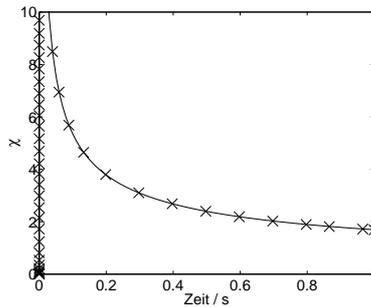
(a) Ros3P, $V1$ (0.88 s)



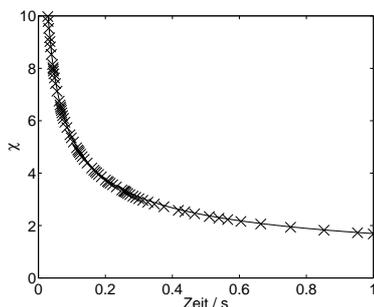
(b) Ros3P, $V2$ (0.89 s)



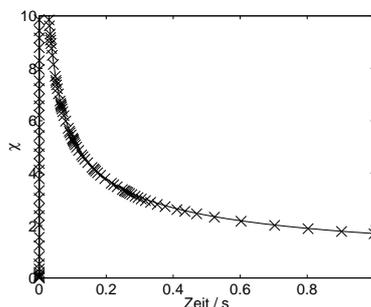
(c) RodasP, $V1$ (2.03 s)



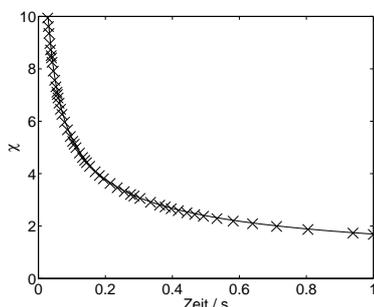
(d) RodasP, $V2$ (2.15 s)



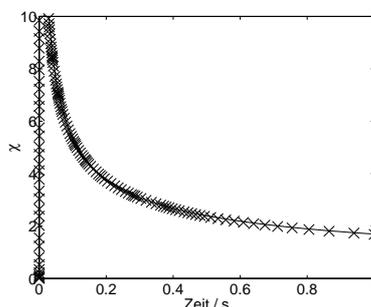
(e) Rowda3, $V1$ (5.31 s)



(f) Rowda3, $V2$ (5.22 s)



(g) Ros2, $V1$ (7.57 s)



(h) Ros2, $V2$ (7.22 s)

Abbildung 7.2: (*Beispiel 1*) Numerische (\times) und Cottrellsche ($-$) CA-Ströme. Die Simulationsdauer ist in Klammern angegeben. Weitere Details siehe Tabelle 7.1 und Text.

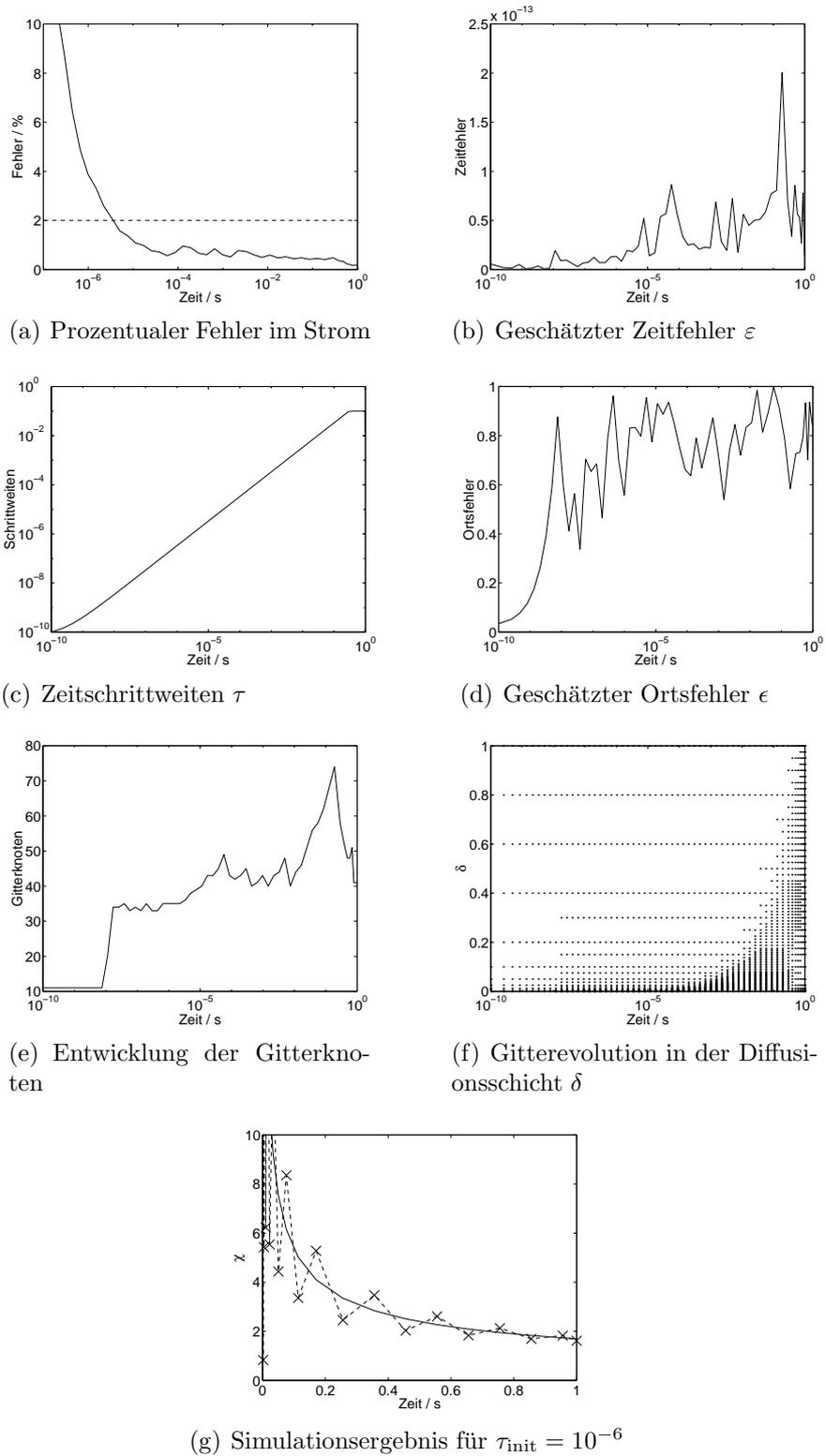


Abbildung 7.3: (*Beispiel 1*) Methoden-Charakteristik von Ros3P nach *V1* für $\tau_{\text{init}} = 10^{-10}$ (a) – (f) und CA-Ströme für $\tau_{\text{init}} = 10^{-6}$ (g).

ein konstantes E - t -Segment mit $E = 0.5$ V und $t_{\max} = 1$ s verwendet. Die einzelnen Toleranzwerte der Lösungsvariablen sind in Tabelle 7.1 angegeben. Beide Varianten liefern für eine relative Fehlertoleranz von 2% sehr gute Approximationen an die Cottrell-Lösung. Es fällt auf, dass Ros3P und RodasP (Abbildungen 7.2(a) – (d)) weniger Zeitschritte benötigen, als Ros2 und Rowda3 (Abbildungen 7.2(e) – (h)). Dies macht sich insbesondere in der Geschwindigkeit der Verfahren bemerkbar. Während Ros3P und RodasP 1-2 s zur Lösung des Modells benötigen, dauern die Simulationen mit Rowda3 und Ros2 bereits 5-7 s.

In $V1$ (Abbildung 7.2, linke Spalte) erfolgt die Berechnung des Anfangsstroms $\chi(0)$ über das Strom-Potential-Residuum (3.98). In $V2$ (Abbildung 7.2, rechte Spalte) wird der Strom hingegen durch die Approximation

$$\partial_x u^{(0)}|_{x=0} \approx \frac{u_1^{(0)} - u_0^{(0)}}{h_0} \quad (7.4)$$

des Konzentrationsgradienten des Edukts A an der Elektrodenoberfläche berechnet, wobei $u_i^{(0)}$ ($i = 1, 2$) die numerische Lösung der Konzentration von A an den ersten beiden Knoten des Gitters und h_0 die Länge des ersten finiten Elements beschreiben. Zu Beginn des Experiments wird eine konstante Konzentrationsverteilung im Diffusionsraum vor der Elektrode angenommen, sodass sich aus (7.4) der Anfangsstrom $\chi(0) = 0$ ergibt. In den ersten Zeitschritten erfolgt die rapide Abnahme der Konzentration $u^{(0)}$ in einer dünnen Schicht vor der Elektrode. Die Gradienten- und folglich die Stromwerte steigen schlagartig an (Abbildung 7.2, rechte Spalte, $t \approx 0$).

Das Gesamtverhalten des Algorithmus ist exemplarisch für die RotheRos3P-Klasse und $V1$ in den Abbildungen 7.3 und 7.4 dargestellt. Betrachtet man den zeitlichen Verlauf des prozentualen Fehlers

$$\frac{|\chi_{\text{Cottrell}}(t) - \chi_{\text{Ros3P}}(t)|}{|\chi_{\text{Cottrell}}(t)|} \times 100 \quad (7.5)$$

zwischen Cottrell- und Ros3P-Lösung (Abbildung 7.3(a)), sieht man, dass die Fehlertoleranz von 2% zu Beginn der Simulation bis zu einem Zeitpunkt von $t \approx 0.5 \cdot 10^{-5}$ nicht

Tabelle 7.1: (*Beispiel 1*) Fehlertoleranzen der Simulation.

Variable	ATOL	RTOL
χ	$1 \cdot 10^{-10}$	0.02
$u^{(0)}$	$1 \cdot 10^{-4}$	0.02
$u^{(1)}$	$1 \cdot 10^{-4}$	0.02

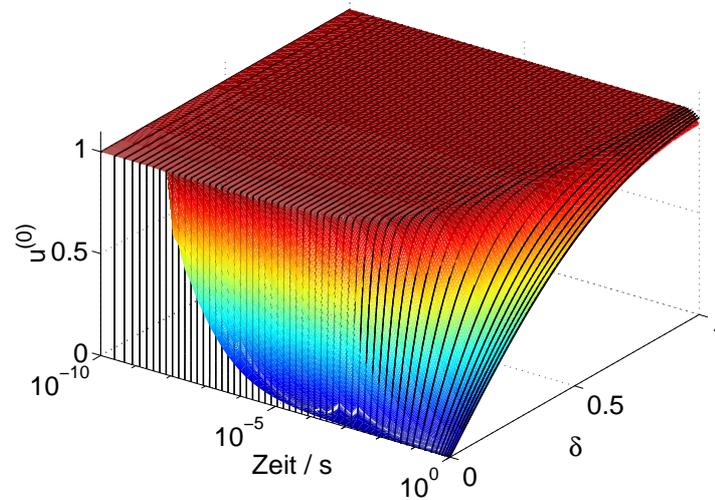


Abbildung 7.4: (*Beispiel 1*) Cottrellsches (schwarze Linien) und mit der Ros3P-Methode nach *V1* simuliertes Konzentrationsprofil (farbig-skalierte Oberfläche).

eingehalten wird. Es sei betont, dass die Cottrell-Gleichung mittels einer homogenen Dirichlet-Randbedingung

$$c_A(t, 0) = 0, \quad t > 0 \quad (7.6)$$

abgeleitet wird[51]. Das zur Simulation verwendete Butler-Volmer-Modell liefert hingegen eine Bedingung an den Massenfluss der Substanzen an der Elektrodenoberfläche (Neumann-Randbedingung). Somit ist eine Abweichung des berechneten Stroms von der Cottrell-Gleichung in den ersten Zeitschritten nicht überraschend. Mit der Wahl der heterogenen Geschwindigkeitskonstanten $k_h = 0.1 \text{ cm} \cdot \text{s}^{-1}$ und einem relativ hohen Elektrodenpotential von $E = 0.5 \text{ V}$ werden die Annahmen der Cottrell-Gleichung jedoch näherungsweise erfüllt, sodass bereits ab $t \approx 0.5 \cdot 10^{-5}$ die vorgegebene Toleranz von 2% (gestrichelte Linie) unterschritten wird.

Der geschätzte Zeitfehler bleibt während der gesamten Simulation klein (Abbildung 7.3(b)), sodass sich die Zeitschrittweiten nach jedem Integrationsschritt stetig vergrößern (Abbildung 7.3(c)). Der hierarchische Fehlerschätzer im Ort bewirkt hingegen bei $t \approx 1 \cdot 10^{-8}$ eine erste Verfeinerung des FE-Gitters (Abbildungen 7.3(d) – (f)), welches sich in den folgenden Zeitschritten je nach Fehlerindikation dem Konzentrationsprofil anpasst. Am Ende der Simulation (Abbildung 7.3(f), $t \approx 1$) erkennt man grobe Gitterbereiche in der Diffusionsschicht für $0 < \delta < 0.3$, die dem nahezu linearen Verlauf der Konzentrationsprofile in dieser Region entsprechen.

Um Oszillationen in der numerischen Lösung zu Beginn zu vermeiden, muss eine sehr kleine Anfangsschrittweite gewählt werden. In den Simulationen genügte ein Wert von

$\tau_{\text{init}} = 1 \cdot 10^{-10}$. Abbildung 7.3(g) zeigt ein Simulationsergebnis, bei dem $\tau_{\text{init}} = 1 \cdot 10^{-6}$ gewählt wurde. Es treten starke Oszillationen in der berechneten Stromkurve auf.

In Abbildung 7.4 ist das analytische und numerisch berechnete Konzentrationsprofil dargestellt. Man erkennt deutlich den sehr steilen Bereich zu Beginn des Experiments $10^{-10} \leq t \leq 10^{-5}$. Ab $t \approx 10^{-8}$ konvergiert die numerische Lösung an der Elektrodenoberfläche gegen den Cottrell-Wert $u_{\text{Cottrell}}^{(0)}(t, 0) = 0$. Der Zeitpunkt entspricht der ersten Gitterverfeinerung aus Abbildung 7.3. Ab $t \approx 10^{-5}$ ist der Cottrell-Konzentrationswert an der Elektrodenoberfläche näherungsweise erreicht, wodurch der Fehler im Strom die vorgegebene Toleranz erfüllt (Abbildung 7.3(a)).

Am Ende der Diffusionsschicht ($\delta \rightarrow 1$) fallen für $t \approx 1$ leichte Abweichungen zwischen simulierter und Cottrell-Lösung auf. Hier zeigt sich erneut der Unterschied in den Randwerten des analytischen und numerischen Modells. Während Ersteres die Randwerte der Konzentrationen im Grenzwert

$$\lim_{x \rightarrow \infty} u^{(0)}(t, x) = 1 \quad (7.7)$$

vorschreibt, verwenden wir zur Simulation am Ende der Diffusionsschicht homogene Neumann-Randbedingungen (siehe Gleichung (3.72)). Die Abweichungen lassen sich durch die Wahl eines größeren Sicherheitsfaktor ζ in Gleichung (3.76) minimieren. In den vorliegenden Simulationen wurde $\zeta = 3$ gewählt.

Die Cottrell-Gleichung liefert den exakten Stromverlauf eines CA-Experiments für vollständig diffusionskontrollierte (hohe Durchtrittsüberspannung) und *reversible* VETs. Bei geringer Durchtrittsüberspannung und für *quasi-reversible* und *irreversible* ET-Kinetik beschreibt Gleichung

$$\chi_{\text{qr}}(t) = \frac{L_{\text{ref}}}{D_{\text{max}}} \cdot k_{\text{f}} \cdot \exp(H^2 \cdot t) \cdot \text{erfc}(H \cdot t^{1/2}) \quad (7.8)$$

die analytische Stromkurve (vgl.[51], S.192) mit

$$H = \frac{k_{\text{f}}}{D_A^{1/2}} + \frac{k_{\text{b}}}{D_B^{1/2}}$$

und

$$\begin{aligned} k_{\text{f}} &= k_{\text{h}} \cdot \exp(\alpha(p - p^0)) \\ k_{\text{b}} &= k_{\text{h}} \cdot \exp(-(1 - \alpha)(p - p^0)) \end{aligned}$$

In diesem Fall stimmen die Randwerte an der Elektrodenoberfläche des analytischen mit denen des numerischen Modells überein. Wir erwarten nun auch zu Beginn des Experiments einen geringeren Fehler in den berechneten Kurven. Zur Simulation wurden als Modell-Parameter die vorherigen Werte übernommen und eine konstante Anregungsfunktion mit $E = 0.3$ V und $t_{\text{max}} = 1$ s eingesetzt. Abbildung 7.5 zeigt die Simulationsergebnisse der RotheRos3P-Klasse nach *V1* für unterschiedliche Geschwindigkeitskonstanten. Man sieht in allen drei Fällen eine exzellente Übereinstimmung zwischen simulierten und analytischen Strömen. Die Fehlertoleranz von 2% wird zu allen Zeiten der Simulation deutlich unterschritten.

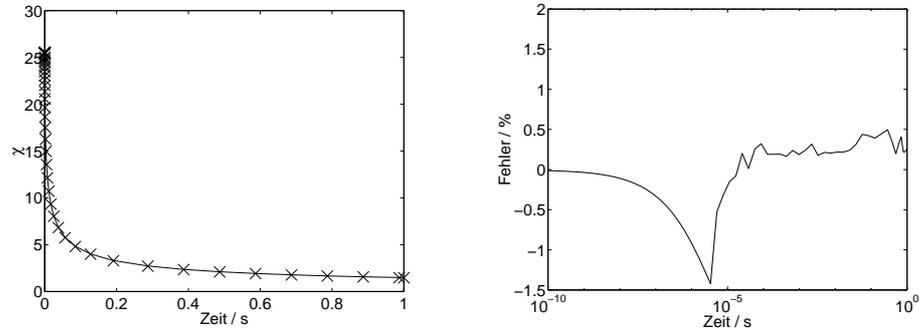
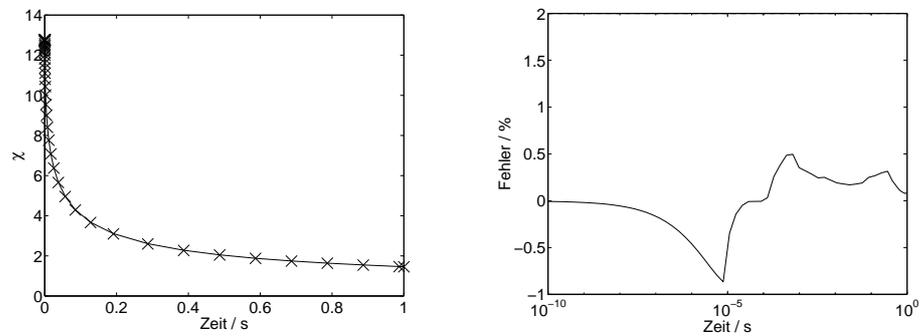
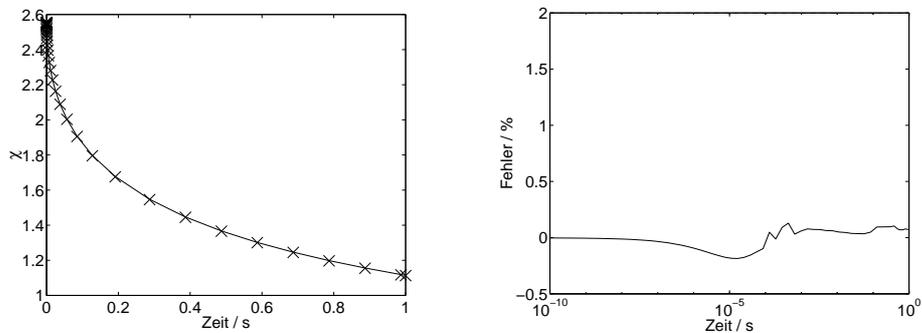
(a) CA-Ströme (links) und prozentualer Fehler (rechts) für $k_h = 0.01 \text{ cm} \cdot \text{s}^{-1}$ (b) CA-Ströme (links) und prozentualer Fehler (rechts) für $k_h = 0.005 \text{ cm} \cdot \text{s}^{-1}$ (c) CA-Ströme (links) und prozentualer Fehler (rechts) für $k_h = 0.001 \text{ cm} \cdot \text{s}^{-1}$

Abbildung 7.5: (*Beispiel 1*) Analytische (—) und simulierte (×) CA-Ergebnisse für quasi-reversible bis irreversible ET-Kinetik (Ros3P, V1, weitere Details siehe Text und Tabelle 7.1).

7.1.2. *Beispiel 2: Doppel- und Multi-Puls-Chronoamperometrie — Eielektronentransfer*

Wir betrachten erneut den VET (7.1) aus *Beispiel 1* mit den auf Seite 109 angegebenen Spezies- und Reaktionsparametern.

In der Doppel-Puls-Chronoamperometrie (DPCA) wird in einem ersten Zeitintervall $[0, t_{\max,1}]$ bei konstantem Elektrodenpotential E_1 das Edukt an der Arbeitselektrode oxidiert (oder reduziert). Anschließend erfolgt durch das Umschalten des Elektrodenpotentials auf einen Wert E_2 für die Dauer $t_{\max,2}$ die Reduktion (Oxidation) des im ersten Schritt erzeugten Produkts. Für den dimensionslosen Strom des ersten Schrittes gilt[51]

$$\chi_1(t) = \frac{L_{\text{ref}}}{D_{\max} c_{\max}} \cdot \frac{D_A^{1/2} c_A^0}{\pi^{1/2} t^{1/2} (1 + \xi \theta')} \quad (7.9)$$

was der verallgemeinerten Cottrell-Gleichung entspricht. Im zweiten Schritt folgt[51]

$$\chi_2(t) = -\frac{L_{\text{ref}}}{D_{\max} c_{\max}} \cdot \frac{D_A^{1/2} c_A^0}{\pi^{1/2}} \left[\left(\frac{1}{1 + \xi \theta'} - \frac{1}{1 + \xi \theta''} \right) \cdot \left(\frac{1}{(t - t_{\max,1})^{1/2}} \right) - \frac{1}{(1 + \xi \theta') t^{1/2}} \right] \quad (7.10)$$

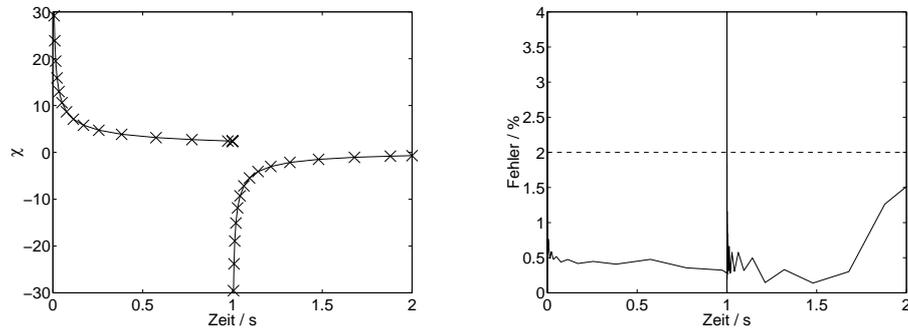
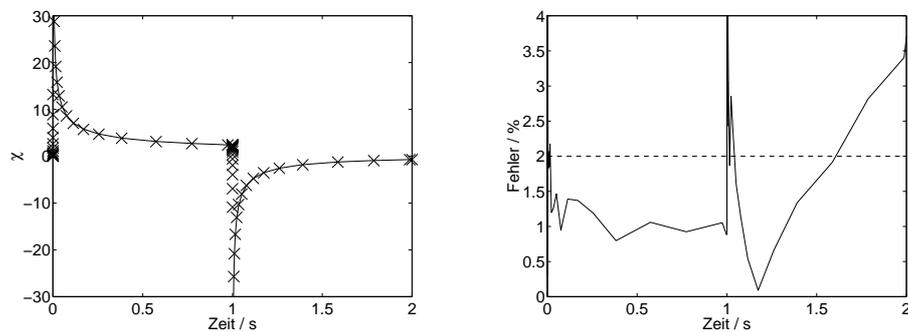
mit den Größen

$$\begin{aligned} \xi &= \left(\frac{D_A}{D_B} \right)^{1/2} \\ \theta' &= \exp \left(-\frac{F}{RT} (E_1 - E^0) \right) \\ \theta'' &= \exp \left(-\frac{F}{RT} (E_2 - E^0) \right) \end{aligned}$$

Um in beiden Schritten eine hohe Durchtrittsüberspannung zu erzeugen, verwenden wir in den Simulationen eine Anregungsfunktion aus zwei Segmenten, mit $E_1 = 0.5 \text{ V}$,

Tabelle 7.2: (*Beispiel 2*) Fehlertoleranzen der Simulation.

(a) <i>V1</i>			(b) <i>V2</i>		
Variable	ATOL	RTOL	Variable	ATOL	RTOL
χ	$1 \cdot 10^{-10}$	0.02	$u^{(0)}$	$1 \cdot 10^{-3}$	0.02
$u^{(0)}$	$1 \cdot 10^{-4}$	0.02	$u^{(1)}$	$1 \cdot 10^{-3}$	0.02
$u^{(1)}$	$1 \cdot 10^{-4}$	0.02			

(a) DPCA-Ströme (links) und prozentualer Fehler (rechts) nach $V1$ (b) DPCA-Ströme (links) und prozentualer Fehler (rechts) nach $V2$ Abbildung 7.6: (*Beispiel 2*) Analytische (–) und simulierte (×) DPCA-Ergebnisse (Ros3P, Toleranzen siehe Tabelle 7.2)

$t_{\max,1} = 1$ s und $E_2 = 0.0$ V, $t_{\max,2} = 1$ s (Abbildung 7.1).

Aus numerischer Sicht stellt das rapide Umschalten des Elektrodenpotentials während der Simulation eine besondere Schwierigkeit dar. Neben der Polstelle zu Beginn des Experiments entsteht eine zweite Polstelle in der Stromkurve am Umschaltzeitpunkt $t_{\max,1}$. Auch die Konzentrationsprofile weisen dort erneut ein unstetiges Verhalten auf.

Um die Algorithmen auf eine sprunghafte Veränderung in der Lösung vorzubereiten, wurde die Schrittweitensteuerung des MFEMSolver-Templates (Abbildung 6.5) durch sogenannte *Sweep* und *Step*-Policies erweitert. Die Sweep-Klasse berücksichtigt keine Sprünge in der Anregungsfunktion. Die Step-Klasse hingegen erkennt das Überschreiten der Segment-Endpunkte. In der Nähe eines Endpunkts wird eine Reduktion der Schrittweite erzwungen, sodass sich der Algorithmus trotz geringer Fehlerschätzung vorsichtig einer möglichen Sprungstelle nähert. Nachdem ein Endpunkt erreicht ist, setzt die Step-Klasse die Simulation mit der gewählten Anfangsschrittweite von $\tau_{\text{init}} = 1 \cdot 10^{-10}$ fort. Des Weiteren wird in $V1$ der ATOL-Wert des Stroms für den ersten Schritt nach der Polstelle auf den größtmöglichen Gleitkommawert des Systems gesetzt. Die Fehlerkon-

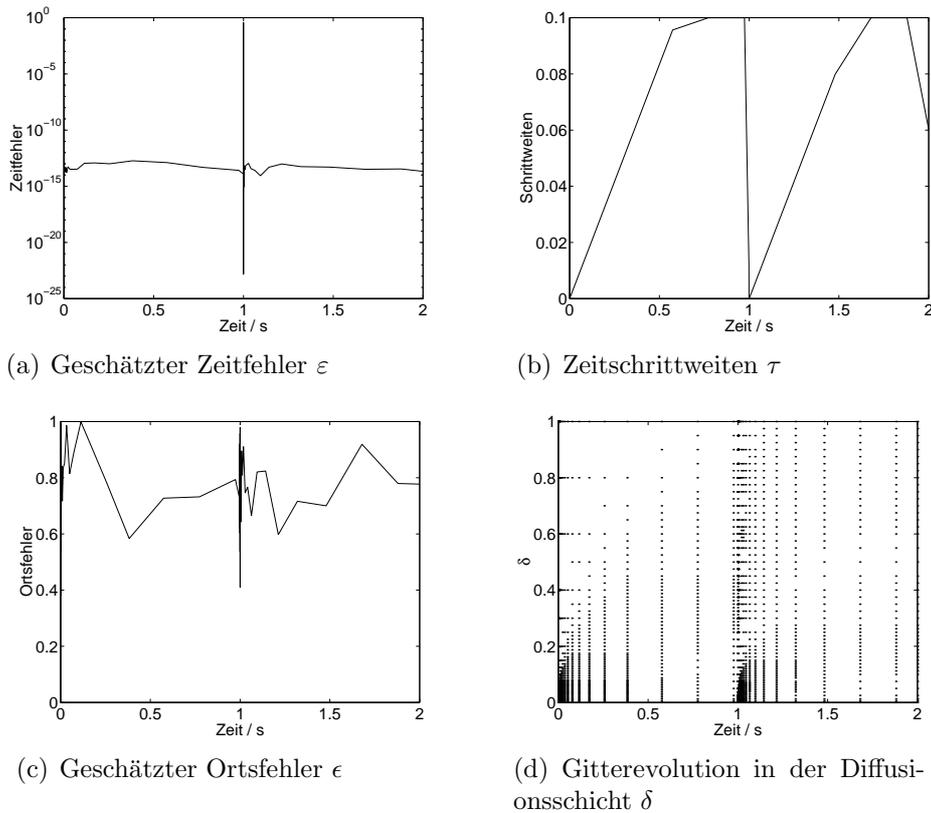


Abbildung 7.7: (*Beispiel 2*) Methoden-Charakteristik des Ros3P-Verfahrens nach *V1* für das DPCA-Experiment.

trolle wird somit für einen Zeitschritt ausgeschaltet. Andernfalls bricht der Algorithmus wegen zu klein gewählten Schrittweiten an der Polstelle ab.

Abbildung 7.6(a) zeigt die Simulationsergebnisse einer MFEMSolver-Klasse der *V1*, welche mit der *RotheRos3P*- und *Step*-Policies konfiguriert wurde. Man sieht eine sehr gute Übereinstimmung zwischen den simulierten und analytischen Strömen in beiden Potentialschritten. Zu Beginn und kurz nach dem Umschaltzeitpunkt ($t = 1$ s) treten wie schon im Cottrell-Experiment (*Beispiel 1*) große Fehler auf. Für alle weiteren Zeitschritte wird die Fehlertoleranz von 2% eingehalten. Wie schon zuvor erwähnt, stimmen die Randwerte des analytischen und numerischen Modells nicht überein, sodass ein Vergleich der beiden Modelle, insbesondere zu Beginn und am Umschaltzeitpunkt des Experiments, zu großen Abweichungen führt. Zudem wird die Fehlerkontrolle im Strom an der mittleren Polstelle durch die *Step*-Policy unterbunden, sodass hier ein großer Fehlerwert ignoriert wird.

In Abbildung 7.6(b) wurde dasselbe Experiment nach *V2* simuliert, jedoch ohne Berücksichtigung der Potentialsprünge in der Anregungsfunktion (*Sweep*-Policy). In diesem

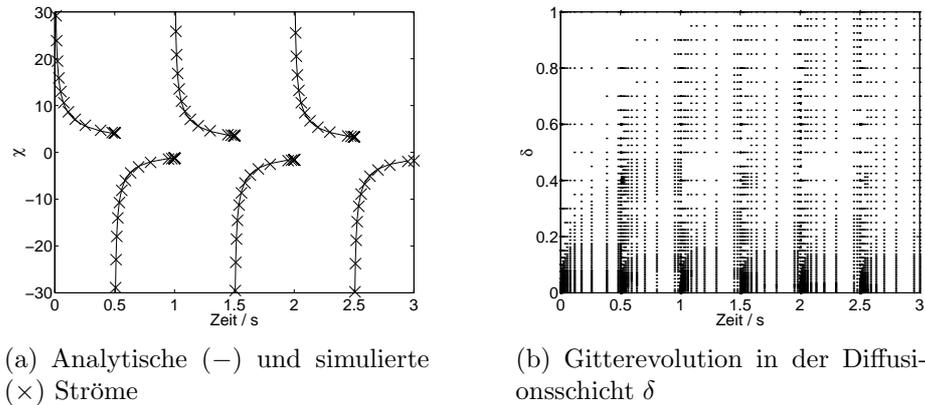


Abbildung 7.8: (*Beispiel 2*) Ros3P-Simulationsergebnisse nach *V1* der Multi-Puls-Chronoamperometrie (MPCA) mit (vgl. Abbildung 7.1) $E_i = 0.5$ ($i = 1, 3, 5$), $E_j = 0.0$ V ($j = 2, 4, 6$), $t_{\max,k} = 0.5$ s ($k = 1, \dots, 5$) (alle weiteren Parameter wie zuvor). Die analytische Lösung wurde aus[144] übernommen.

Fall müssen größere absolute Fehlertoleranzen eingesetzt werden, um die Simulation ohne Abbruch am Umschaltzeitpunkt durchführen zu können (siehe Tabelle 7.2). Dort gilt $u^{(1)}(t_{\max,1}, 0) \approx 0$, was durch die Skalierung (6.62) mit zu kleinen ATOL-Werten zu einer deutlichen Vergrößerung des geschätzten Fehlers ε führt. Die Verwendung einer Step-Policy führte in *V2* zu keiner Verbesserung der Ergebnisse.

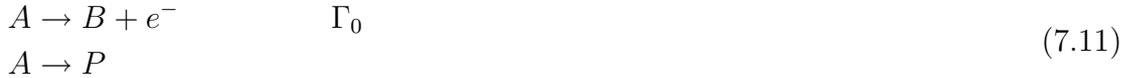
Es fällt auf, dass in *V2*, in der nur der Fehler in den Konzentrationen kontrolliert wird, auch gegen Ende des Experiments die Toleranz im Strom nicht eingehalten werden kann. Hier erweist sich die Berechnung des Stroms als zusätzliche Lösungsvariable (*V1*) als die bessere Alternative.

In den Abbildungen 7.7(a) – (d) wird die Auswirkung des Potentialsprunges in der Anregungsfunktion auf das Verhalten des Algorithmus deutlich. Abbildung 7.7(b) zeigt das Einwirken der Step-Policy auf die Schrittweiten. Kurz vor $t = 1$ s wird die Schrittweite trotz eines geringen Zeitfehlers (Abbildung 7.7(a)) verringert, bis der Endpunkt des ersten E - t -Segments erreicht ist. Anschließend wird mit der Anfangsschrittweite neu gestartet. Die Unstetigkeit in den Konzentrationsprofilen führt auch zu einer großen Fehlerschätzung im Ort, sodass in der Nähe der Elektrode am Umschaltzeitpunkt neue Gitterpunkte eingeführt werden.

Abbildung 7.8 zeigt Simulationsergebnisse für mehrere Potentialpulse (vgl.[144]). Mit der oben beschriebenen Modifizierung der Schrittweitenkontrolle kann auch hier eine sehr gute Übereinstimmung zwischen den simulierten und analytischen Strömen erzielt werden.

7.1.3. Beispiel 3: Chronoamperometrie — Das Reinert-Berg-System

Für den CA-Strom des Reinert-Berg-Systems



gilt[1]

$$\chi_{\text{RB}}(t) = \zeta \cdot \exp(-k_f \cdot t) \cdot \frac{1}{\sqrt{\pi(t/t_{\text{max}})}} \quad (7.12)$$

wobei ζ den Sicherheitsfaktor zur Berechnung der Dicke der Diffusionsschicht (siehe Gleichung (3.76)) und k_f / s^{-1} die Geschwindigkeitskonstante der homogenen Reaktion beschreiben. Zusätzlich zu den Modellparametern aus *Beispiel 1* (siehe S.109) wählen wir für das Produkt P die Anfangskonzentration $c_P^0 = 0 \text{ mol} \cdot \text{l}^{-1}$ und $D_P = 1 \cdot 10^{-5} \text{ cm}^2 \text{ s}^{-1}$. Die Kinetik der homogenen Reaktion des Systems wird durch die IrrPowerRateLaw-Klasse von Ecco (siehe Abbildung 4.11) wiedergegeben. Eine irreversible Butler-Volmer-Kinetik wird zur Zeit von Ecco nicht unterstützt. Gleichung (7.12) gilt jedoch unter Cottrell-Bedingungen. Wir wählen also, wie schon in *Beispiel 1*, eine konstante Anregungsfunktion mit $E = 0.5 \text{ V}$ und $t_{\text{max}} = 1 \text{ s}$. Unter diesen Bedingungen wird der zweite Term in Gleichung (3.38) sehr klein, sodass die von Ecco gebildete reversible Butler-Volmer-Gleichung (näherungsweise) verwendet werden kann.

Die Simulation des Reinert-Berg-Systems soll zeigen, dass die Geschwindigkeitsgesetze des Ecco-Compilers für moderat gewähltes k_f korrekte Resultate liefern. Für großes k_f machen sich die unterschiedlichen Randterme der Modellansätze in den Ergebnissen bemerkbar, sodass ein direkter Vergleich nicht mehr sinnvoll ist.

In der Eingabesprache des Compilers lautet der Reaktionsmechanismus

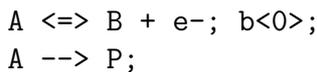


Abbildung 7.9 zeigt die Simulationsergebnisse der *RotheRos3P*-Klasse nach *V1*. Es wurden vier verschiedene Werte für die Geschwindigkeitskonstante k_f verwendet und die Fehlertoleranzen des einfachen CA-Experiments eingesetzt (Tabelle 7.1). Die linke Abbildung zeigt eine sehr gute Übereinstimmung zwischen simulierten und analytischen Kurven. Auf der rechten Seite sind die relativen Fehler in den CA-Strömen

$$\frac{|\chi_{\text{RB}}(t) - \chi_{\text{Ros3P}}(t)|}{|\chi_{\text{RB}}(t)|} \times 100 \quad (7.13)$$

für die ersten beiden Geschwindigkeitskonstanten abgebildet. Bis auf den aus den vorherigen Beispielen bekannten Fehler am Anfang des Experiments wird die Fehlertoleranz zu allen Zeiten eingehalten. Bei den höheren Werten von k_f erreicht der Strom den Nullpunkt, sodass eine rein relative Fehlerberechnung nach Gleichung (7.13) nicht mehr möglich ist.

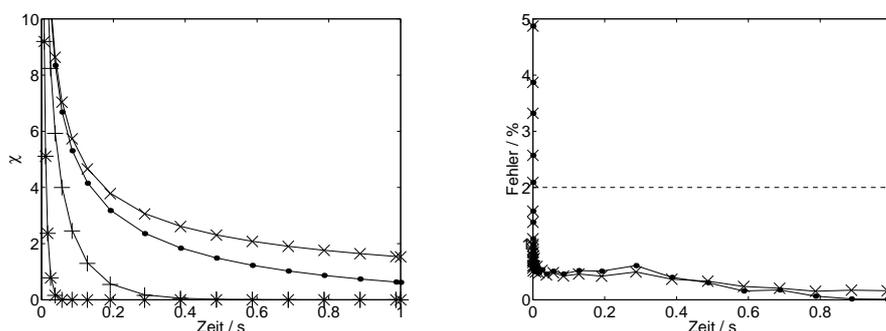


Abbildung 7.9: (*Beispiel 3*) Analytische (–) und simulierte CA-Ströme (links) des Reinert-Berg-Systems für vier unterschiedliche Werte von $k_f = 0.1$ (×), 1 (●), 10 (+), 100 s^{-1} (*). Rechts sind die relativen Fehler für $k_f = 0.1$ und 1 s^{-1} abgebildet.

7.1.4. *Beispiel 4*: Cyclische Voltammetrie — Einelektronentransfer

Die Simulation eines einfachen Elektronentransfers der Form (7.1) mit einer Anregungsfunktion der cyclischen Voltammetrie (CV) wurde nach $V2$ in[184] durchgeführt. Wie bereits in der Einleitung dieses Kapitels erwähnt, stellte sich heraus, dass insbesondere das Rosenbrock-Schema Ros3P den anderen Verfahren hinsichtlich der erzielten Genauigkeit und Effizienz überlegen war.

Ein Nachteil der $V2$ ist es, dass die Fehlerkontrolle lediglich in den Konzentrationen durchgeführt wird. Die Berechnung des Stroms erfolgt dabei über die zusätzliche Approximation des Gradienten der elektroaktiven Substanz an der Elektrodenoberfläche (siehe z.B. Gleichung (7.4)). Die Genauigkeit der simulierten Ströme hängt somit auch von der Qualität der Gradientenapproximation ab. Anhand des Testsystems aus[184] konnte zwar für einen speziellen Fall gezeigt werden, dass eine Verringerung der Fehler-toleranz in den Konzentrationen auch eine Verbesserung der Stromkurven zur Folge hat. Eine direkte Kontrolle des Stromfehlers ist allerdings nach $V2$ nicht möglich.

In[184] musste zudem die Unterteilung des Randelementes des FE-Gitters verboten werden, um Oszillationen in der Stromkurve nach einer Gitterverfeinerung zur Elektrodenoberfläche hin zu unterbinden, die in manchen Fällen auftraten. Dieser Eingriff in die Gitteradaption führte bei RodasP zu Konvergenzproblemen. Durch die veränderte Fehlerskalierung (siehe Abschnitt 6.4) konnte auf ein Verfeinerungsverbot der Randelemente in beiden Varianten vollständig verzichtet werden.¹

Darüber hinaus wurde der Algorithmus aus[184] durch das Einbringen des Stroms als Lösungsvariable erweitert (hier $V1$), wodurch zum einen die zusätzliche Fehlerkontrol-

¹Eine mögliche Erklärung für das verbesserte Verhalten der Algorithmen wird am Ende des Abschnitts 6.4 gegeben.

le im Strom ermöglicht ist. Zum anderen folgen die Stromwerte unmittelbar aus dem Strom-Potential-Residuum (3.98), sodass eine zusätzliche Approximation des Gradienten nicht mehr notwendig ist.

Wir wollen daher in diesem Abschnitt das Verhalten von *V1* zur Simulation des Testsystems aus[184]



betrachten. Dazu wurden Simulationen mit den Modellparametern aus *Beispiel 1* (S.109) für vier verschiedene Werte der heterogenen Geschwindigkeitskonstante $k_h = 0.1, 0.01, 0.001, 0.0001 \text{ cm} \cdot \text{s}^{-1}$ durchgeführt. Im Grenzfall eines reversiblen Elektronentransfers (hier $k_h \approx 0.1$) können die Oxidationsströme mit den (numerisch berechneten) $\sqrt{\pi}\chi$ -Werten aus[207] gemäß

$$\chi_{\text{NS}} = \frac{L_{\text{ref}} \sqrt{a}}{\sqrt{D_{\text{max}}}} \cdot \sqrt{\pi} \chi \quad (7.14)$$

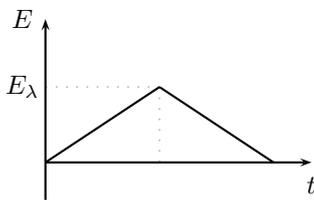
verglichen werden ($a = Fv/(RT)$). Eine geschlossene, analytische Lösung der CV-Ströme existiert nicht.

Die Anregungsfunktion der Simulation bestand aus zwei linearen E - t -Segmenten, mit Anfangspotential $E_{\text{init}} = 0.0 \text{ V}$ und Umschaltpotential $E_\lambda = 0.5 \text{ V}$, bei einer Spannungsvorschubgeschwindigkeit von $v = 0.2 \text{ Vs}^{-1}$ (Abbildung 7.10(a)).

Abbildung 7.11 zeigt die CV-Ströme der Simulationen. Beide Varianten ergeben bei vergleichbarer Geschwindigkeit (siehe Zeitangaben unter den Abbildungen) nahezu äquivalente Ergebnisse. Die Stromkurven konvergieren für großes k_h gegen die Referenzwerte. Für $k_h = 0.1 \text{ cm} \cdot \text{s}^{-1}$ stimmt die nach *V1* simulierte Kurve mit χ_{NS} näherungsweise überein, während die nach *V2* gemäß (7.4) approximierten Ströme etwas geringere Werte annehmen (vgl. Abbildung 7.11(c)).

Erwartungsgemäß wandern die Peakpotentiale bei kleinem k_h auseinander, während die Peakströme abnehmen.

Die zur Simulation eingesetzten Fehlertoleranzen sind in Abbildung 7.10(b) angegeben.

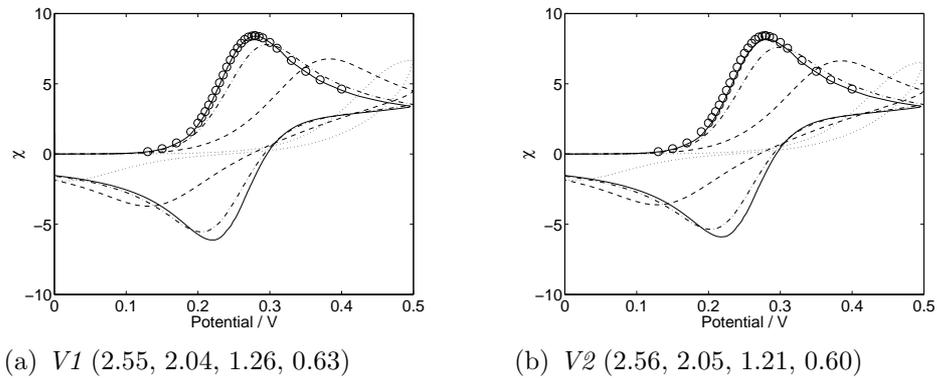


Variable	ATOL	RTOL
χ	1	0.02
$u^{(0)}$	$1 \cdot 10^{-4}$	0.02
$u^{(1)}$	$1 \cdot 10^{-4}$	0.02

(a) Skizzierte CV-Anregungsfunktion

(b) Fehlertoleranzen der Simulation.

Abbildung 7.10: (*Beispiel 4*) Anregungsfunktion und Fehlertoleranzen.

(a) $V1$ (2.55, 2.04, 1.26, 0.63)(b) $V2$ (2.56, 2.05, 1.21, 0.60)

E_{NS}	χ_{NS}	E_{V1}	χ_{V1}	E_{V2}	χ_{V2}
0.1300	0.1699	0.1297	0.1640	0.1297	0.1540
0.2100	3.0200	0.2112	2.9589	0.2112	2.7905
0.2250	4.5300	0.2258	4.4563	0.2258	4.2229
0.2400	6.1910	0.2411	6.1625	0.2411	5.8762
0.2550	7.5500	0.2558	7.5496	0.2558	7.2502
0.2800	8.4182	0.2812	8.3924	0.2812	8.1682
0.3500	5.8890	0.3500	5.8561	0.3500	5.8282

(c) Auswahl an numerischen Werten für $k_h = 0.1 \text{ cm} \cdot \text{s}^{-1}$

Abbildung 7.11: (*Beispiel 4*) Simulierte CV-Ströme (Ros3P-Methode) für $k_h = 0.1$ (—), 0.01 (---), 0.001 (— · —), 0.0001 (····) $\text{cm} \cdot \text{s}^{-1}$ und χ_{NS} -Werte (\circ), (siehe Gleichung (7.14)). Die Simulationszeiten (in s) sind, nach abnehmenden k_h -Werten geordnet, in Klammern angegeben.

Im Gegensatz zu den Simulationen der Chronoamperometrie durchschreitet die Stromkurve eines Cyclovoltammogramms im Reduktionsschritt die Nulllinie. In $V1$ führt die Skalierung des Stromfehlers mit kleinem ATOL-Wert für $\chi \rightarrow 0$ jedoch zu einer unnötig starken Verkleinerung der Schrittweite (Gleichung (6.62)), wodurch sich die Simulationen deutlich verlangsamen; mit $\text{ATOL}_\chi = 1 \cdot 10^{-3}$ dauerte die Simulation für $k_h = 0.1 \text{ cm} \cdot \text{s}^{-1}$ bereits 9–10 s. Alle Simulationen wurden daher mit einem Wert von $\text{ATOL}_\chi = 1$ durchgeführt.² Die Verkleinerung von ATOL_χ führte zu keiner wesentlichen Verbesserung der Ergebnisse.

Abbildung 7.12 zeigt die Eigenschaften des Ros3P-Verfahrens nach $V1$ für den Grenzfall des reversiblen Elektronentransfers. Wie schon bei der Chronoamperometrie wurde zu Beginn eine Schrittweite von $\tau_{\text{init}} = 1 \cdot 10^{-10}$ gewählt, um inkonsistente Anfangswerte

²Mit $\text{ATOL}_\chi = 1$ wird die Schrittweite überwiegend durch den Fehler in den Konzentrationsprofilen kontrolliert. In beiden Varianten werden folglich dieselben Zeitschritte gewählt: In Abbildung 7.11(c) stimmen die Potentialwerte E_{V1} und E_{V2} überein.

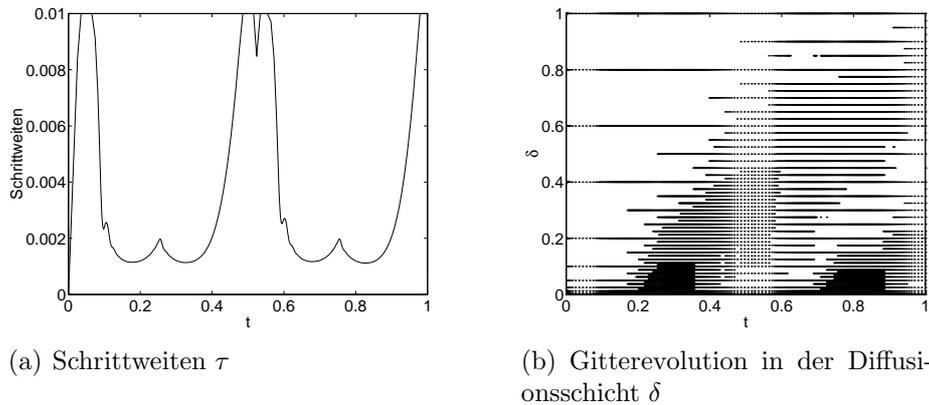


Abbildung 7.12: (*Beispiel 4*) Methoden-Charakteristik von Ros3P nach *V1* für $k_h = 0.1 \text{ cm} \cdot \text{s}^{-1}$. Die Zeitachsen sind normiert angegeben.

auszugleichen. Während in *V2* stets der Anfangsstrom $\chi(0) = 0$ angenommen wird, erfolgt in *V1* die Berechnung von $\chi(0)$ aus den Anfangskonzentrationen $u^{(0)}(0, 0) = 1$ und $u^{(1)}(0, 0) = 0$ und dem Startpotential der Anregungsfunktion über das Strom-Potential-Residuum. Anhand der Gleichungen (3.85) und (3.98) sieht man, dass mit einem Startpotential $E_{\text{init}} \ll E^0$ ($E_{\text{init}} - E^0 \rightarrow -\infty$) für den Anfangsstrom einer Oxidation ($\nu_e > 0$) $\chi(0) \rightarrow 0$ folgt. Die Störungen der Anfangswerte sind daher gering, sodass sie insbesondere durch die Wahl der kleinen Anfangsschrittweite zu späteren Zeitpunkten keinen erkennbaren Einfluss auf die Stromkurve besitzen. Für $E_{\text{init}} \rightarrow E^0$ werden zu Beginn in beiden Varianten zunehmend CA-ähnliche Stromkurven erhalten.

Die kleine Anfangsschrittweite wird rasch vergrößert (Abbildung 7.12(a)). Bei $t \approx 0.1$ ($E \approx 0.1 \text{ V}$ in Abbildung 7.11(a)) erfolgt daraufhin eine erste Verkleinerung der Schrittweite, was dem Beginn der zunehmenden Produktbildung an der Elektrodenoberfläche entspricht. Insgesamt spiegelt die Variation der Schrittweiten den zeitlichen Verlauf des Cyclovoltammogramms wieder (vgl. Abbildung 7.11(a), durchgezogene Linie), welcher im Bereich $0.1 \leq E \leq 0.4 \text{ V}$ ($0.1 \leq t \leq 0.4$), bzw. im zweiten Segment $0.4 \geq E \geq 0.1 \text{ V}$ ($0.6 \leq t \leq 0.9$) die größte Veränderung erfährt. Die zwischenzeitliche Vergrößerung der Schrittweite für $0.2 \leq t \leq 0.3$ und $0.7 \leq t \leq 0.8$ entspricht den Potentialbereichen, in denen die erste Ableitung des Stroms den Nullpunkt durchläuft.

Die Gitterveränderungen sind in Abbildung 7.12(b) abgebildet. Hier erkennt man die Veränderungen in den Konzentrationsprofilen der Substanzen bezüglich des Ortes, welche an der Elektrodenoberfläche in den Anstiegszonen des Cyclovoltammogramms besonders groß sind.

7.1.5. Beispiel 5: Linear Sweep Voltammetry — Katalytische Folgereaktion pseudo-erster Ordnung

Ein numerisch anspruchsvolles System stellt der katalytische Reaktionsmechanismus[51]



dar, wobei die Kinetik der Folgereaktion nach einem Geschwindigkeitsgesetz erster Ordnung verläuft. Mit zunehmender Reaktionsgeschwindigkeit der Folgereaktion entstehen extrem dünne Reaktionsschichten an der Elektrodenoberfläche[1, 144].

Zu Systemen der Form (7.15) gehören insbesondere *medierte* Elektrodenreaktionen, in denen die Elektronenübertragung auf das Substrat S indirekt über einen sogenannten *Mediator* (hier das Redoxsystem A/B) verläuft[208]. Das Substrat liegt dabei im hohen Überschuss vor, sodass die Folgereaktion pseudo-erster Ordnung ist.

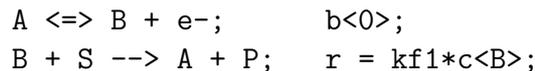
Analytische Lösungen liegen für große k_f -Werte nur für reversibel verlaufende VETs vor[207, 209, 210]. Für die Stromkurve gilt[207]

$$\chi_{\text{NS}} = \frac{L_{\text{ref}}}{D_{\text{max}} c_{\text{max}}} \cdot \frac{\sqrt{D_A k_f c_A^0}}{1 + \exp\left(-\frac{F}{RT}(E - E_{1/2})\right)} \quad (7.16)$$

Der Grenzstrom ist proportional zur Wurzel der Geschwindigkeitskonstanten $\kappa_f = k_f \cdot t_{\text{max}}$

$$\chi_G \propto \sqrt{\kappa_f} \quad (7.17)$$

Das Halbpeakpotential $E_{1/2}$ entspricht im reversiblen Fall dem Formalpotential des VETs. In der Eingabesprache von Ecco lautet Mechanismus (7.15)



wobei $\mathbf{kf1}$ die *dimensionslose* Geschwindigkeitskonstante κ_f der Folgereaktion beschreibt. Zusätzlich zu den Speziesparametern des ersten Beispiels (S.109) wurde $c_S^0 = 1 \cdot 10^{-2}$, $c_P^0 = 0 \text{ mol} \cdot \text{l}^{-1}$ und $D_S = D_P = 1 \cdot 10^{-5} \text{ cm}^2 \text{ s}^{-1}$ eingesetzt. Mit $L_{\text{ref}} = 3 \cdot \sqrt{D_{\text{max}} \cdot t_{\text{max}}}$ und $c_{\text{max}} = c_S^0$ ergibt sich in Gleichung (7.17) der Proportionalitätsfaktor

$$\chi_G = 3 \cdot 10^{-2} \cdot \sqrt{\kappa_f} \quad (7.18)$$

Zur Variation des Potentials an der Elektrodenoberfläche wurde eine Anregungsfunktion mit einem einzelnen linearen E - t -Segment ($E_{\text{init}} = 0$, $E_{\text{max}} = 1 \text{ V}$, $v = 0.2 \text{ Vs}^{-1}$) verwendet (*linear sweep voltammetry*, LSV).

Abbildung 7.13 zeigt einige Simulationsergebnisse der RotheRos3P-Klasse nach $V1$ für unterschiedliche Geschwindigkeitskonstanten κ_f . Mit zunehmenden κ_f erkennt man den Übergang des Voltammogramms in eine S-förmige Welle, wobei für große κ_f -Werte die

7. Anwendungen

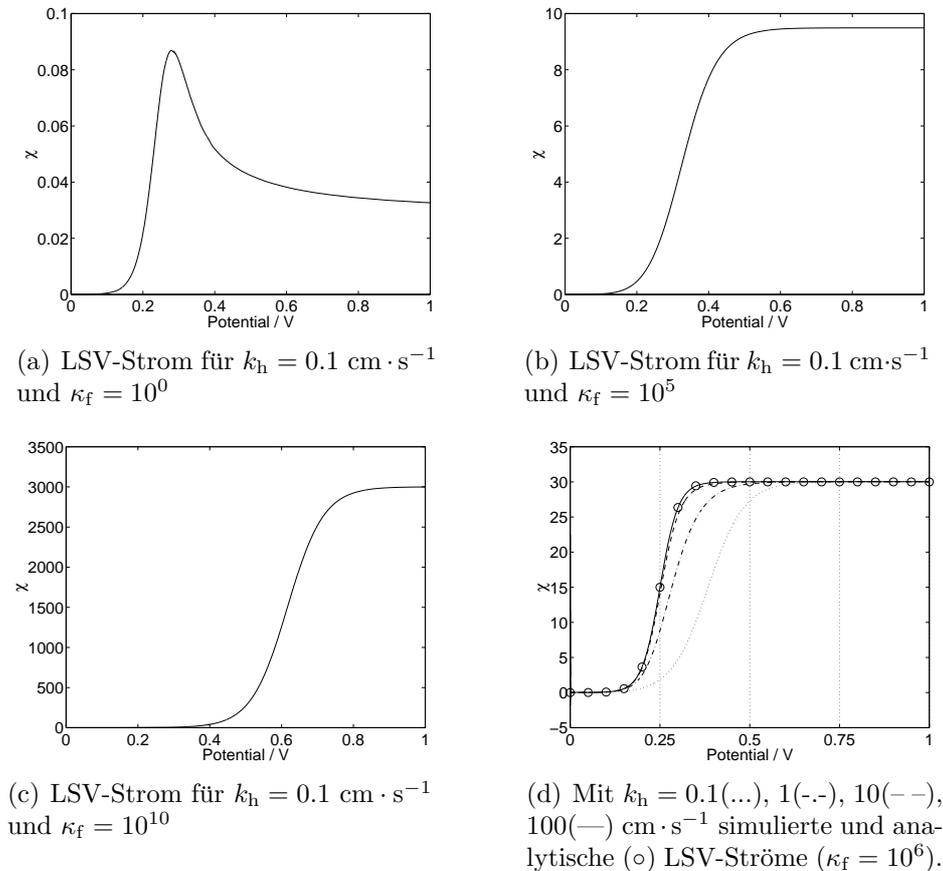


Abbildung 7.13: (*Beispiel 5*) Simulationsergebnisse der Ros3P-Methode nach VI

Grenzströme Gleichung (7.18) folgen. In allen mit quasi-reversibler ET-Kinetik ($k_h < 10 \text{ cm} \cdot \text{s}^{-1}$) simulierten Kurven erfolgt der Stromanstieg jedoch später als im reversiblen Grenzfall. Letzterer ist ab $k_h = 100 \text{ cm} \cdot \text{s}^{-1}$ erreicht (Abbildung 7.13(d)). Die simulierte Stromkurve folgt dann der analytischen Lösung (7.16) und das Halbpeakpotential entspricht dem für den VET gewählten Formalpotential $E^0 = 0.25 \text{ V}$.

Abbildung 7.14 zeigt die Eigenschaften des Ros3P-Verfahrens, aufgenommen während einer Simulation mit $k_h = 0.1 \text{ cm} \cdot \text{s}^{-1}$ und $\kappa_f = 10^5$. In den Abbildungen 7.14(a) und 7.14(b) erkennt man, dass kleine Zeitschrittweiten gewählt werden und die Anzahl der Gitterknoten mit Beginn des Stromanstiegs (vgl. Abbildung 7.13(b)) stetig zunimmt. Vor der Elektrode entsteht eine extrem dünne Reaktionsschicht, welche dort zu einer starken Gitterverfeinerung führt (Abbildung 7.14(c)).

In Abbildung 7.14(d) sind die eingesetzten Fehlertoleranzen angegeben. Im Gegensatz zu den vorherigen Beispielen muss hier über eine geeignete Wahl der ATOL-Werte den Größenunterschieden in den dimensionslosen Lösungskonzentrationen Rechnung getragen werden. Es stellte sich heraus, dass für die Berechnung des Stroms kleine ATOL-

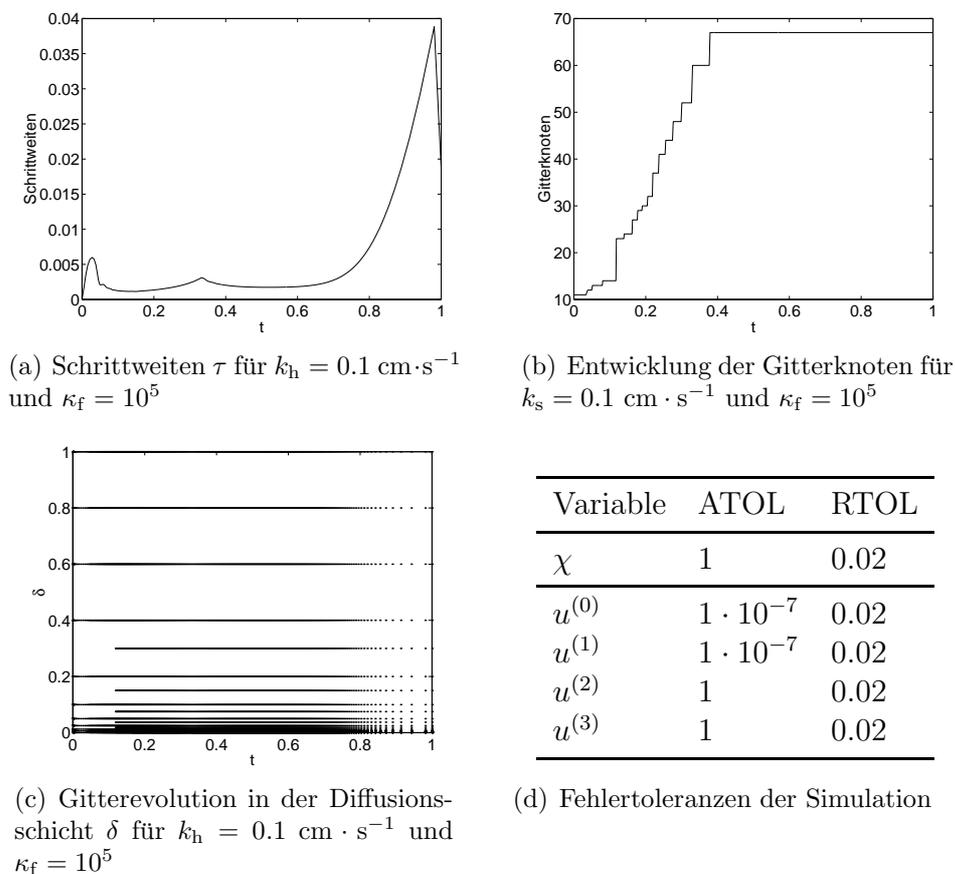


Abbildung 7.14: (*Beispiel 5*) Methoden-Charakteristik und Fehlertoleranzen von Ros3P nach V1.

Werte für die Konzentrationen ($u^{(0)}$ und $u^{(1)}$) des Redox-Paares A/B benötigt wurden, wobei die Fehlerkontrolle in den Substrat und Produkt-Konzentrationen ($u^{(2)}$ und $u^{(3)}$) weitestgehend ausgeschaltet werden konnte. Die steile Reaktionsfront bildet sich zwischen A/B unmittelbar an der Elektrodenoberfläche. In weiterer Entfernung von der Elektrodenoberfläche benötigt man für eine genaue Approximation der Konzentrationsprofile von S und P kleinere ATOL-Werte. Eine Simulation mit den Werten $\text{ATOL}_S = \text{ATOL}_P = 1 \cdot 10^{-3}$, $k_h = 0.1 \text{ cm} \cdot \text{s}^{-1}$ und $\kappa_f = 10^5$, führte bei einer Rechenzeit von ca. 40 Sekunden zu derselben Stromkurve aus Abbildung 7.13(b), welche zuvor innerhalb von 30 Sekunden erreicht wurde. Das Gitter aus Abbildung 7.14(c) wurde dabei zur Lösung hin stärker verfeinert.

7.1.6. *Beispiel 6: Cyclische Voltammetrie — Zweielektronentransfer mit Disproportionierungsreaktion*

Neben unstetigen Randwerten (*Beispiele 1 – 3*) und extrem dünnen Reaktionsschichten an der Elektrodenoberfläche (*Beispiel 5*) stellen steile, sich in der Diffusionsschicht bewegende Reaktionsfronten eine weitere Herausforderung an numerische Verfahren dar. Ein System, in dem sich eine solche Front ausbilden kann, ist ein Zweielektronentransfer mit gekoppelter Disproportionierungsreaktion des ersten Redoxprodukts (siehe z.B.[115,145,211–215])

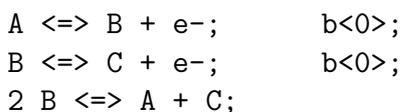


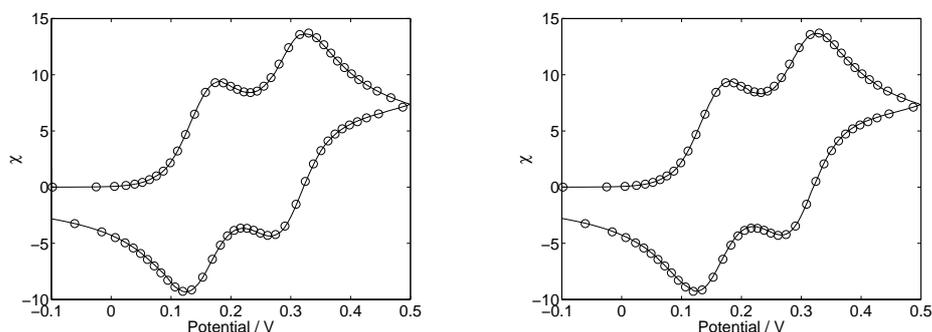
Die elektrochemischen Eigenschaften dieses EE-Disp-Systems werden insbesondere durch die Lage der Formalpotentiale E_0^0 und E_1^0 des ersten und zweiten VETs bestimmt. Die Gleichgewichtskonstante der Disproportionierung hängt über die Beziehung (siehe auch Gleichung (4.45) in Abschnitt 4.3.7)

$$K = \exp\left[\frac{F}{RT}(E_0^0 - E_1^0)\right] \quad (7.20)$$

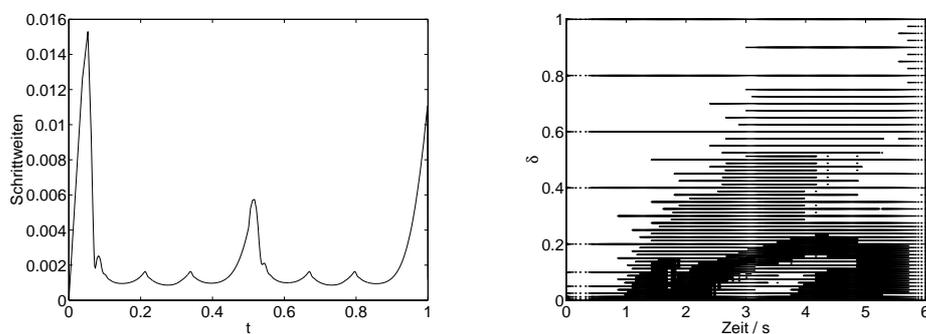
von den Formalpotentialen ab. Gilt $E_0^0 < E_1^0$, folgen die VETs in voneinander separierten Schritten. Bei einer linearen Variation des Elektrodenpotentials (z.B. in der CV) mit $E_{\text{init}} < E_0^0$ und $E_{\text{max}} > E_1^0$ entsteht zunächst die Spezies B an der Elektrodenoberfläche und diffundiert in die Lösung. Steigt das Elektrodenpotential weiter, erfolgt die Oxidation von B zu C , wodurch die Konzentration von B an der Elektrodenoberfläche wieder abnimmt. Es entsteht somit eine laufende Konzentrationswelle der Spezies B in die Diffusionsschicht hinein, deren Ausprägung von der Größe der Geschwindigkeitskonstante der Disproportionierung k_f bestimmt wird. Für großes k_f ist sie besonders steil.

Analytische Lösungen existieren nicht. Es ist jedoch bekannt, dass für reversibel verlaufende VETs die Disproportionierung keinen Einfluss auf die Voltammogramme besitzen kann[115]. Wir können daher im reversiblen Grenzfall die simulierte Stromkurve mit den simulierten Ergebnissen eines Zweielektronentransfers *ohne* Berücksichtigung der Disproportionierungsreaktion vergleichen. Dabei sollten keine Abweichungen auftreten. Zur Simulation wurde eine CV-Anregungsfunktion mit $E_{\text{init}} = -0.1$ und $E_{\text{max}} = 0.5$ V, bei $v = 0.2$ Vs⁻¹ gewählt. In der Eingabesprache des Compilers lautet der EE-Disp-Mechanismus





(a) Simulierte Voltammogramme mit (—) und ohne (○) Berücksichtigung der Disproportionierungsreaktion bei $k_f = 10$ (links) und $k_f = 10^9 \text{ l} \cdot \text{mol}^{-1} \text{ s}^{-1}$ (rechts).



(b) Schrittweiten τ (links) und Gitterrevolution (rechts) für $k_f = 10^9 \text{ l} \cdot \text{mol}^{-1} \text{ s}^{-1}$

Abbildung 7.15: (*Beispiel 6*) Simulierte Voltammogramme und Methodencharakteristik von Ros3P nach *V1* für den EE-Disp-Mechanismus.

Die Speziesparameter betragen $c_A^0 = 1 \cdot 10^{-4}$, $c_B^0 = c_C^0 = 0 \text{ mol} \cdot \text{l}^{-1}$ und $D_A = D_B = D_C = 1 \cdot 10^{-5} \text{ cm}^2 \text{ s}^{-1}$. Als Reaktionsparameter wurden $E_0^0 = 0.15$, $E_1^0 = 0.3 \text{ V}$, $\alpha_0 = \alpha_1 = 0.5$ und $k_{h,0} = k_{h,1} = 0.1 \text{ cm} \cdot \text{s}^{-1}$, sowie $k_f = 10$ bzw. $10^9 \text{ l} \cdot \text{mol}^{-1} \text{ s}^{-1}$ eingesetzt. Aus den Formalpotentialen und mit $T = 293.15 \text{ K}$ ergibt sich eine Disproportionierungskonstante von $K = 0.00263781$. Alle Simulationen wurden mit der *RotheRos3P*-Klasse und den Fehlertoleranzen aus *Beispiel 4* (siehe Abbildung 7.10(b)) durchgeführt.

Abbildung 7.15(a) zeigt zwei Voltammogramme, die mit einer langsamen und schnellen Disproportionierungskinetik simuliert wurden. Der Vergleich mit einem entsprechenden Voltammogramm, welches ohne Berücksichtigung der Disproportionierung berechnet wurde (Kreise), zeigt die erwartete Unabhängigkeit der Stromkurven von k_f .

In Abbildung 7.15(b) sind die Schrittweiten- und die Gitterrevolution während der Simulation mit $k_f = 10^9 \text{ l} \cdot \text{mol}^{-1} \text{ s}^{-1}$ abgebildet. Die Variation der Schrittweite spiegelt, wie schon in *Beispiel 4* (Abbildung 7.12(a)), den Stromverlauf des Cyclovoltammogramms wieder. Des Weiteren deutet die Gitterstruktur an, dass ab dem Zeitpunkt $t \approx 2 \text{ s}$ eine

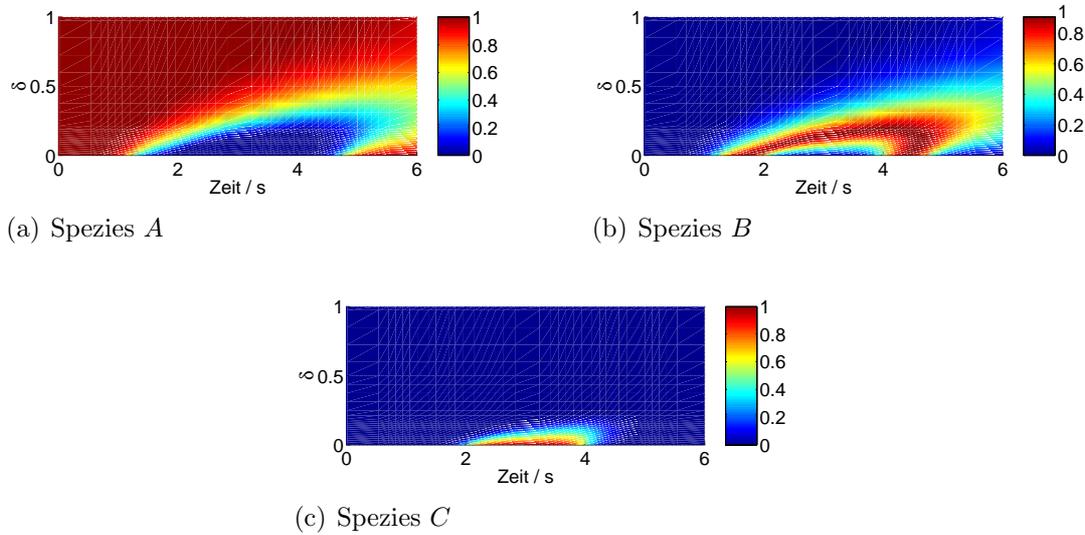


Abbildung 7.16: (*Beispiel 6*) Simulierte Konzentrationsprofile des EE-Disp-Mechanismus.

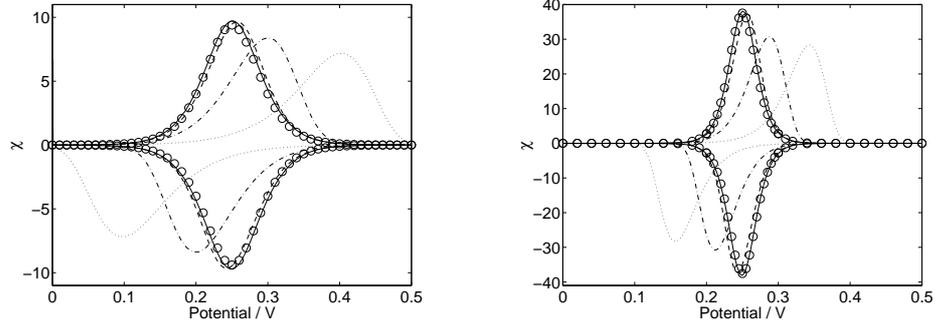
Konzentrationsbewegung in der Diffusionsschicht stattfindet.

Abbildung 7.16 zeigt die dazugehörigen Konzentrationsprofile der Spezies. Man erkennt, dass sich ab $t \approx 2$ s die Konzentration von B (Abbildung 7.16(b)) wellenförmig in die Diffusionsschicht ausbreitet. Der wandernde Wellenberg bewirkt die in Abbildung 7.15(b) gezeigten, sich zunehmend von der Elektrodenoberfläche entfernenden Gitterverfeinerungen.

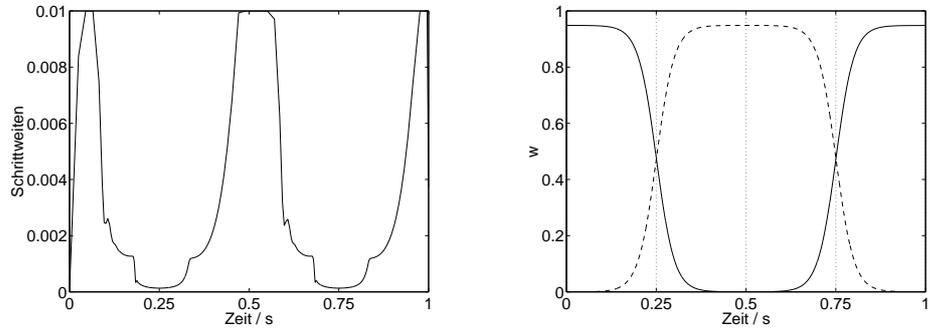
7.1.7. *Beispiel 7*: Cyclische Voltammetrie — Elektroaktive Adsorptionsschicht

Wie bereits in Abschnitt 3.7 erwähnt, existieren in der Literatur bisher nur weniger systematische Modellansätze für Adsorptionssysteme. Die allgemeine Modellierung von Adsorptions- und Oberflächenreaktionen aus Kapitel 3, die Erzeugung sämtlicher Modellgleichungen durch Ecco sowie die Kopplung der generierten Gleichungen an die numerischen Algorithmen eröffnen die Möglichkeit, unter den in Abschnitt 3.7 beschriebenen Annahmen, Adsorptionsprobleme in nahezu unbegrenzter Vielfalt zu simulieren.

Eine einfache, quantitative Analyse der Ergebnisse auf der Basis sogenannter “Arbeitskurven” ist allerdings aufgrund der hohen Komplexität der Modelle unmöglich. Chevalier et al.[112] bildeten daher kürzlich in einer Simulationsstudie auf 12 Seiten ihrer Publikation insgesamt 216 Cyclovoltammogramme ab (!), um die Einflüsse diverser kinetischer Parameter auf CV-Kurven zu beschreiben. Die Darstellung umfangreicher Simulations-



(a) Analytische (Gleichung (7.22), \circ) und simulierte CV-Kurven für den Ein- (links) und Zweielektronentransfer (rechts) bei $k_{s,1} = 1(\cdots)$, $10(-\cdots)$, $10^2(-\cdot-)$ und $10^3 \text{ s}^{-1}(\text{—})$



(b) Schrittweiten τ (links) und Variation der Oberflächenkonzentrationen (rechts) von A_{ads} (—) und B_{ads} (---) für $k_{s,1} = 10^3 \text{ s}^{-1}$ und $n = 1$.

Abbildung 7.17: (*Beispiel 7*) Simulationsergebnisse der Ros3P-Methode nach *V1*.

ergebnisse zur besseren Übersicht ist sicherlich sinnvoll. Wir wollen dem Beispiel von Chevallier et al. in dieser Arbeit jedoch nicht folgen, sondern interessieren uns zunächst, wie schon in den Beispielen zuvor, für die Verlässlichkeit der numerischen Ergebnisse. Dafür betrachten wir für $n = 1$ und $n = 2$ eine elektroaktive, adsorbierte Schicht



einer Spezies A .

Das System (7.21) besitzt, zumindest für einen Grenzfall, eine analytische Lösung: Unter den Annahmen, dass

- \Rightarrow sich vor Beginn des Experiments die Adsorptionsschicht von A_{ads} vollständig gebildet hat ($c_{A_{\text{ads}}}^0 = c_s$, wobei c_s die Sättigungskonzentration beschreibt)
- \Rightarrow die Adsorption im Gleichgewicht vorliegt, und dieses nahezu vollständig auf der Seite der adsorbierten Form liegt ($k_{f,0} \gg 0$, $K_0 \gg 0$)

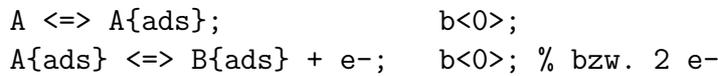
⇒ der Oberflächenelektronentransfer (OET) reversibel verläuft (großes $k_{s,1}$)

lassen sich die simulierten CV-Kurven mit der Lösung (Gleichung (14.3.11) in[51])

$$\chi_{\text{rev}} = \frac{L_{\text{ref}}}{D_{\text{max}}c_{\text{max}}} \cdot \frac{n^2F}{RT} \cdot \frac{vc_{A_{\text{ads}}}^0 (b_{A_{\text{ads}}}/b_{B_{\text{ads}}}) \exp[nF/(RT)(E - E^0)]}{1 + (b_{A_{\text{ads}}}/b_{B_{\text{ads}}}) \exp[nF/(RT)(E - E^0)]} \quad (7.22)$$

vergleichen, wobei die “Stärke” der Adsorption von A_{ads} und B_{ads} näherungsweise identisch ist: $b_{A_{\text{ads}}}/b_{B_{\text{ads}}} \approx 1$ (zur genauen Bedeutung von $b_{A_{\text{ads}}}$ und $b_{B_{\text{ads}}}$ siehe Abschnitt 7.1.8). In diesem Fall muss das Peakpotential des Stromsignals dem Formalpotential des OETs entsprechen[51].

In der Eingabsprache des Compilers lautet der Mechanismus



Die Parameter der Simulation betragen $c_A^0 = 10^{-4} \text{ mol} \cdot \text{l}^{-1}$, $D_A = 10^{-5} \text{ cm}^2\text{s}^{-1}$, $c_{A_{\text{ads}}}^0 = c_s = 10^{-10}$, $c_{B_{\text{ads}}}^0 = 0 \text{ mol} \cdot \text{cm}^{-2}$, $K_0 = 10^5 \text{ cm}^3\text{mol}^{-1}$, $k_{f,0} = 10^2 \text{ cm}^3\text{mol}^{-1}\text{s}^{-1}$ (und somit $k_{b,0} = 10^{-3} \text{ s}^{-1}$), $E_1^0 = 0.25 \text{ V}$ und $\alpha_1 = 0.5$.

Für den OET wurden unterschiedliche heterogene Geschwindigkeitskonstanten $k_{s,1} = 1, 10, 10^2, 10^3 \text{ s}^{-1}$ eingesetzt, und das Elektrodenpotential wurde bei $T = 293.15 \text{ K}$ von $E_{\text{init}} = 0.0 \text{ V}$ bis $E_{\text{max}} = 0.5 \text{ V}$ mit $v = 1 \text{ Vs}^{-1}$ in einem Zyklus linear variiert.

Abbildung 7.17 zeigt die Simulationsergebnisse der Ros3P-Methode nach *V1*, wobei für alle Variablen die für die CV üblichen Fehlertoleranzen (siehe Abbildung 7.10(b)) eingesetzt wurden. Für beide Modelle erkennt man eine hervorragende Übereinstimmung zwischen den analytischen und numerischen Lösungen für $k_{s,1} = 10^3 \text{ s}^{-1}$ (Abbildung 7.17(a)). Für kleineres k_s wandern die Peaks erwartungsgemäß auseinander und verändern dabei ihre zuvor symmetrische Form (vgl. Abbildung 5 in[78]).

In Abbildung 7.17(b) ist die Variation der Schrittweiten und der Konzentrationsverlauf der adsorbierten Formen auf der Elektrodenoberfläche für $k_{s,1} = 10^3 \text{ s}^{-1}$ abgebildet. Unter den oben beschreibenden Bedingungen findet die elektrochemische Umsetzung von A_{ads} kurz vor Erreichen des Formalpotentials statt (hier bei $t \approx 0.2 \text{ s}$). Hier nimmt die Konzentration ab, B_{ads} wird gebildet. Es sei erwähnt, dass die Oberflächenkonzentrationen in unserem dimensionslosen Modell mit Hilfe des zusätzlichen Faktors φ normiert werden (siehe Abschnitt 3.10). Der Wert der dimensionslosen Sättigungskonzentration liegt hier somit knapp unterhalb des Wertes 1.0.

7.1.8. *Beispiel 8: Cyclische Voltammetrie — Adsorptionssystem ohne und mit Folgereaktionen auf der Elektrodenoberfläche*

In *Beispiel 7* (Abschnitt 7.1.7) erfolgt die Ladungsübertragung ausschließlich zwischen der Elektrode und den Teilchen in der Adsorptionsschicht. Man erhält einen einzelnen, symmetrischen Adsorptionspeak (siehe Abbildung 7.17(a)).

Wir wollen nun ein komplexeres System betrachten, in dem sich der Gesamtstrom aus den Teilströmen der adsorbierten und der gelösten Spezies zusammensetzt.



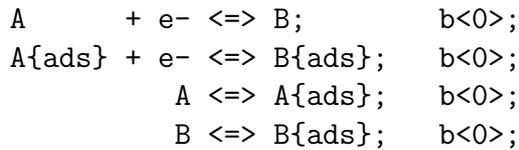
Analytische Lösungen sind nicht verfügbar. Wopshall und Shain[93] (siehe auch[51] S.595 ff.) und später Laviron[78] behandelten das System jedoch numerisch. Beide Arbeiten gehen von ausreichend schnellen Adsorptionsreaktionen aus, sodass deren Kinetik keinen Einfluss auf die Voltammogramme besitzt. Mit den von Ecco unterstützten Modellgleichungen und unter den Bedingungen

- ⇒ die elektrochemischen Reaktionen verlaufen reversibel (große Werte für $k_{h,0}$ und $k_{s,1}$)
- ⇒ beide Adsorptionen befinden sich im Gleichgewichtszustand (große Werte für $k_{f,2}$ und $k_{f,3}$)

können wir die Simulationen zumindest qualitativ mit den Ergebnissen dieser frühen Arbeiten vergleichen.

In der Literatur werden im Wesentlichen vier Fälle unterschieden, in denen entweder das Edukt A stark (I) oder schwach (II), oder das Produkt B stark (III) oder schwach (IV) adsorbiert. Die ‘‘Stärke’’ der Adsorptionen wird dabei durch die Adsorptionskoeffizienten $b_{A_{\text{ads}}}$ und $b_{B_{\text{ads}}}$ beschrieben (siehe Gleichung (7.22)), die den Gleichgewichtskonstanten K_2 und K_3 der beiden Adsorptionen entsprechen (vgl. Gleichungen (1) und (2) in[78]). In unserem kinetischen Adsorptionsmodell werden die Fälle demnach durch (I) $K_2 \gg K_3$, (II) $K_2 \geq K_3$, (III) $K_2 \ll K_3$ und (IV) $K_2 \leq K_3$ angenähert.

In der Eingabesprache des Compilers lautet der Mechanismus (7.23)



Die (festen) Parameter der Simulation betragen $c_A^0 = 10^{-4}$, $c_B^0 = 0 \text{ mol} \cdot \text{l}^{-1}$, $D_A = D_B = 10^{-5} \text{ cm}^2 \text{s}^{-1}$, $c_s = 10^{-10}$, $c_{B_{\text{ads}}}^0 = 0 \text{ mol} \cdot \text{cm}^{-2}$, $k_{f,2} = k_{f,3} = 10^5 \text{ cm}^3 \text{mol}^{-1} \text{s}^{-1}$, $E_0^0 = 0.0 \text{ V}$, $\alpha_0 = \alpha_1 = 0.5$, $k_{h,0} = 1 \text{ cm} \cdot \text{s}^{-1}$ und $k_{s,1} = 10^3 \text{ s}^{-1}$.

Das Elektrodenpotential wurde bei $T = 293.15 \text{ K}$ von $E_{\text{init}} = 0.6 \text{ V}$ bis $E_\lambda = -0.3 \text{ V}$ mit $v = 0.05 \text{ Vs}^{-1}$ in einem Zyklus linear variiert.

Abbildung 7.18 zeigt simulierte Cyclovoltammogramme bei verschiedenen Kombinationen der Gleichgewichtskonstanten K_2 und K_3 . Die dazugehörigen Modellparameter sind in Tabelle 7.3 aufgeführt. Das Formalpotential des OETs ergibt sich dabei nach[78]

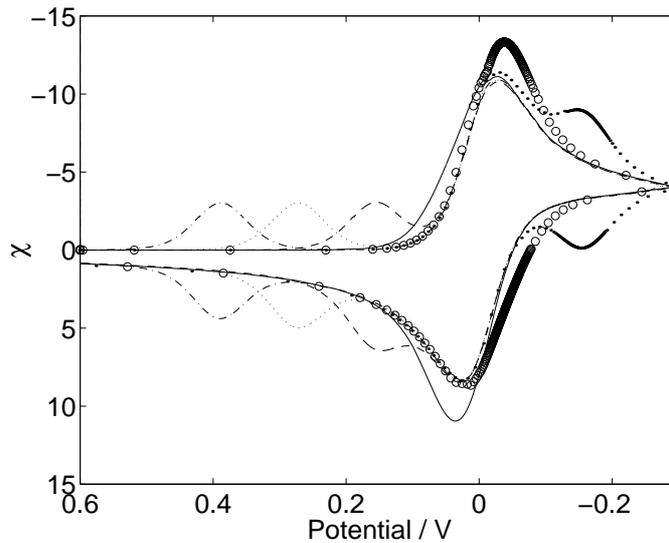


Abbildung 7.18: (*Beispiel 8*) Simulierte CVs (Ros3P, V1) des Adsorptionssystems mit $K_2/K_3 = 10^3$ (●), 10 (○), 10^{-1} (—), 10^{-3} (- -), 10^{-5} (···) und 10^{-7} (- · -). Weitere Parameter siehe Text und Tabelle 7.3.

$$E_1^0 = E_0^0 - \frac{RT}{F} \ln \frac{K_2}{K_3} \quad (7.24)$$

und stellt somit einen gemäß Abschnitt 4.3.7 abhängigen thermodynamischen Parameter dar. Die Werte von E_1^0 in Tabelle 7.3 wurden von Ecco automatisch berechnet.

Die verschiedenen Anfangskonzentrationen von A_{ads} ergeben sich als Gleichgewichtskonzentrationen aus der Lösung eines ODE-Systems der Form (3.73), wobei die Konzentration von A als näherungsweise konstant ($c_A(t) = c_A^0$) angenommen wird.

Die simulierten CVs in Abbildung 7.18 weisen das Verhalten der Kurven von Wopshall und Shain auf. Mit abnehmenden Verhältnis K_2/K_3 erkennt man gut die Übergänge zwischen den vier Fällen (I) – (IV). Für $K_2/K_3 = 10^3$ (Symbol ●) liegt eine starke Adsorption des Edukts vor (Fall I). Hier befindet sich das Formalpotential des OETs hinter dem des VETs und wir beobachten einen *Postpeak*. Bei schwächerer Adsorption von A ($K_2/K_3 = 10$, Fall II, Symbol ○) wird der Adsorptionspeak von dem stärkeren Signal des VETs überlagert und bewirkt eine Vergrößerung des Stromsignals im Vorwärtsscan. Bei $K_2/K_3 = 10^{-1}$ (Fall III, durchgezogene Linie) liegt eine schwache Adsorption des Produkts vor, wodurch sich der Rückpeak des Stromsignals erhöht. Mit abnehmenden Verhältnis K_2/K_3 nimmt die Adsorptionsstärke des Produkts zu und das Formalpotential des OETs verschiebt sich zu höheren Potentialen hin. Man beobachtet dort einen *Präpeak*.

Chemische Folgereaktionen auf der Elektrodenoberfläche wurden bisher in der elektrochemischen Literatur nur sehr selten theoretisch behandelt. Laviron eröffnete die Dis-

Tabelle 7.3: (*Beispiel 8*) Abhängige Modellparameter der Simulation ohne Folgereaktion (obere Zeilen) und mit Folgereaktion (letzte Zeile)

K_2/K_3	K_2	$K_3/ \text{cm}^3\text{mol}^{-1}$	E_1^0/ V	$c_{A_{\text{ads}}}^0$	$c_{X_{\text{ads}}}^0/ \text{mol} \cdot \text{cm}^{-2}$
10^3	10^4	10	-0.1745	$1.0 \cdot 10^{-10}$	
10	10^2	10	-0.0582	$9.9 \cdot 10^{-11}$	
10^{-1}	10	10^2	0.0582	$5.0 \cdot 10^{-11}$	
10^{-3}	10	10^4	0.1745	$5.0 \cdot 10^{-11}$	
10^{-5}	10	10^6	0.2908	$5.0 \cdot 10^{-11}$	
10^{-7}	10	10^8	0.4071	$5.0 \cdot 10^{-11}$	
10^{-5}	10	10^6	0.2908	$3.3 \cdot 10^{-11}$	$3.3 \cdot 10^{-11}$

kussion, ob voltammetrische Methoden überhaupt in der Lage sind, alleine zwischen Folgereaktionen in Lösung oder auf der Elektrodenoberfläche zu unterscheiden[106]. Das von Laviron abgeleitete analytische Modell kann das Gesamtsystem allerdings nur für die Grenzfälle beschreiben, in denen die Ladungsübertragung entweder über einen scheinbar heterogenen Prozess, oder nur über einen Oberflächenprozess eintritt. Die Betrachtung des Gesamtsystems war aufgrund der hohen Komplexität analytisch nicht möglich.

Mirčeski und Lovrić erweiterten das Adsorptionssystem (7.23) durch Folgereaktionen der Produkte B und B_{ads} und untersuchten in einer Simulationsstudie das elektrochemische Verhalten des Systems unter den Bedingungen der *square wave voltammetry* (SWV)[104]. Mit der in[104] eingesetzten numerischen Methode, konnte das Modell nur für Folgereaktionen gelöst werden, die nach einem Geschwindigkeitsgesetz erster Ordnung verlaufen. Mirčeski und Lovrić nahmen dabei unter anderem an, dass sich die Adsorptionen von A und B stets im Gleichgewicht befinden und jeweils einer linearen Adsorptionsisotherme gehorchen. Weiterhin betrachteten sie den Fall, in dem beide Reaktanden mit derselben Stärke adsorbieren, sodass sich die Lage der Formalpotentiale des VETs und OETs nicht unterscheiden. Die Resultate von Mirčeski und Lovrić zeigten, dass mit Hilfe der SWV und unter den oben genannten Annahmen, tatsächlich zwischen Oberflächen- und Volumenreaktionen unterschieden werden kann.

Es liegt auf der Hand, dass eine genaue Untersuchung des Systems mit Hilfe des allgemeineren Modells von Ecco, welches dynamische Adsorptionsprozesse, verschiedene Adsorptionsstärken und nahezu beliebige Geschwindigkeitsgesetze der Folgereaktionen in Lösung und auf der Elektrodenoberfläche unterstützt, zu weiteren aussagekräftigen Resultaten führen kann. Die Komplexität des Systems ist jedoch so groß, dass eine ausführliche Analyse im Rahmen dieser Arbeit nicht vorgenommen wurde.

Stattdessen soll im Folgenden die Fähigkeit des Ecco-Compilers demonstriert werden,

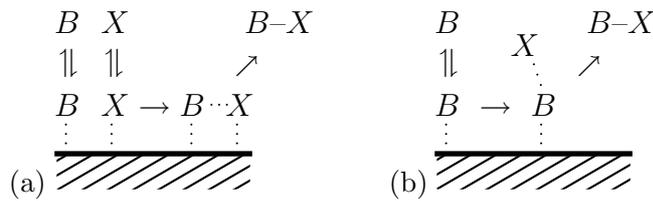
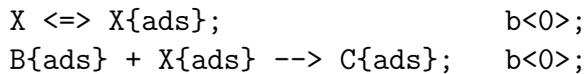


Abbildung 7.19: Mechanismen einer Oberflächenreaktion nach (a) Langmuir-Hinshelwood und (b) Rideal-Eley[216].

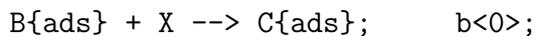
in Verbindung mit den Algorithmen aus Kapitel 6 an Elektronentransfers gekoppelte Oberflächenreaktionen zu simulieren.

Zwei Reaktionstypen, die z.B. im Zusammenhang elektrokatalytischer Prozesse in Brennstoffzellen angenommen werden[5, 8, 10], sind in Abbildung 7.19 skizziert. Im Langmuir-Hinshelwood-Mechanismus (Abbildung 7.19(a)) erfolgt die Reaktion zwischen B_{ads} und einem ebenfalls adsorbierten Reaktionspartner X_{ads} nach einem Geschwindigkeitsgesetz zweiter Ordnung. Eine solche *homogene* Oberflächenreaktion kann auf einfache Weise durch die zusätzlichen Reaktionen



in das System (7.23) eingebracht werden. Die Desorption des Produkts C_{ads} könnte als zusätzliche Reaktion formuliert werden, soll hier allerdings vernachlässigt werden.

Im Falle des Rideal-Eley-Mechanismus (Abbildung 7.19(b)) findet die Reaktion zwischen der gelösten Form X und B_{ads} statt. Die Formulierung einer solchen *heterogenen* Oberflächenreaktion lautet in der Eingabesprache des Compilers beispielsweise³



Im Folgenden soll die Veränderung eines Adsorptions-Präpeaks aus Abbildung 7.18 ($K_2/K_3 = 10^{-5}$) für beide Mechanismen unter den obigen CV-Bedingungen betrachtet werden. Des Weiteren gelten die zusätzlichen Modellparameter $c_X^0 = 1 \cdot 10^{-4} \text{ mol} \cdot \text{l}^{-1}$, $c_{C_{\text{ads}}}^0 = 0 \text{ mol} \cdot \text{cm}^{-2}$ und für die Adsorption von X : $K_4 = 10 \text{ cm}^3 \text{mol}^{-1}$, $k_{f,4} = 10^5 \text{ cm}^3 \text{mol}^{-1} \text{s}^{-1}$. Die verbleibenden Parameter sind in der letzten Zeile in Tabelle 7.3 angegeben.

Abbildung 7.20 zeigt das Potentialfenster des Präpeaks für verschiedene Werte der Geschwindigkeitskonstanten k_f der Oberflächenreaktionen. Das Signal des VETs (Abbil-

³Zur Vereinfachung wird hier die Desorption des Produkts ebenfalls vernachlässigt. Es sei jedoch betont, dass das elektrochemische Verhalten des Systems auch von der Kinetik und Gleichgewichtslage der Desorption abhängt.

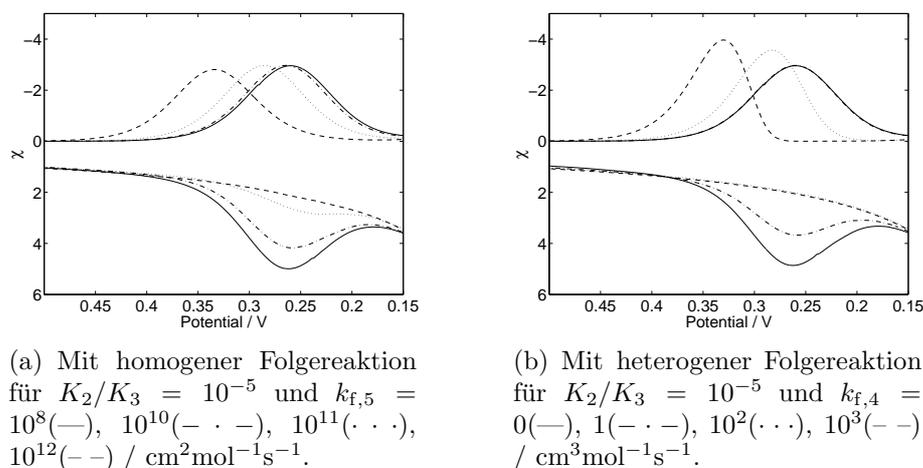


Abbildung 7.20: (*Beispiel 8*) Simulierte CVs (Ros3P-Methode, $V1$). Die Abbildungen zeigen einen Ausschnitt aus dem Potentialfenster der Abbildung 7.18. Weitere Parameter siehe Text und Tabelle 7.3.

Abbildung 7.18) bleibt von der Oberflächenkinetik unbeeinflusst. Mit zunehmender Reaktionsgeschwindigkeit können in beiden Fällen zwei Effekte beobachtet werden: Zum einen verschiebt sich der Reduktionspeak des OET-Signals zu höheren Potentialen hin, zum anderen verschwindet der Oxidationspeak.

Vergleicht man die Form der Reduktionssignale (Abbildungen 7.20(a) und (b)), erkennt man, dass sich bei der homogenen Oberflächenreaktion die Symmetrie der Peakform kaum ändert. Im heterogenen Prozess hingegen erscheint der Peak zunehmend unsymmetrisch.

7.1.9. *Beispiel 9*: Cyclische Voltammetrie in Dünnschichtsystemen mit einer einzelnen Arbeitselektrode

In den bisherigen Beispielen wurden stets *semi-infinite* Diffusionsbedingungen angenommen. In Dünnschichtzellen hingegen wird der Diffusionsraum vor der Arbeitselektrode entweder von einer inerten Wand oder einer zweiten Arbeitselektrode begrenzt (*finite* Diffusion)[51].

Der Gebrauch von Dünnschichtzellen in der Elektroanalytik wird zumeist aufgrund der vorgegebenen Elektrodenanordnung durch hohe Zellwiderstände erschwert.⁴ Aus diesem Grund werden Dünnschichtzellen zur Bestimmung kinetischer Parameter von Elektro-

⁴Die Referenzelektrode kann nicht wie bei herkömmlichen Zellen in einem geringen Abstand zur Arbeitselektrode angebracht werden, sodass häufig ein hoher unkompensierter Widerstand (*iR*-Drop) die experimentellen Ergebnisse beeinträchtigt. Solche Effekte werden hier nicht berücksichtigt.

denreaktionen eher selten verwendet. In der Vergangenheit kamen sie u.a. in Studien über Adsorptionsprozesse oder zur Aufklärung komplexer Reaktionsmechanismen zum Einsatz (siehe Diskussion in[51], Abschnitt 11.7).

Eine größere Bedeutung besitzen Dünnschichtzellen dagegen in der Spektroelektrochemie (SEC) und in Experimenten zur Untersuchung elektrochemisch generierter Chemilumineszenz (Elektrochemilumineszenz, ECL)[51].

Während man sich in den klassischen Experimenten aus den Abschnitten 7.1.1 – 7.1.8 eher für den Stromfluss durch die Arbeitselektrode interessiert, ist in SEC- und ECL-Experimenten die Kenntniss des Konzentrationsverlaufes im *Inneren* der Zelle wichtig. Für die Simulation solcher Systeme können daher nur adaptive Gitter, welche auf einer Fehlerschätzung in den Konzentrationen basieren, in angemessener Rechenzeit zufriedenstellende Resultate liefern. Ein Beispiel, welches den Vorteil des hier eingesetzten adaptiven Verfahrens verdeutlicht, wird anhand eines ECL-Experiments mit zwei sich gegenüberstehenden Arbeitselektroden im nächsten Abschnitt gegeben.

Ein weiterer, mathematischer Aspekt des Modells einer Dünnschichtzelle ist dessen Analogie zu Modellen von elektrochemischen Dünnschicht-Systemen, wie z.B. elektroaktiven Filmen oder chemisch modifizierten Elektroden. Erst kürzlich verwendete Mirčeski[217] beispielsweise in einer theoretischen Studie das Modell einer Dünnschichtzelle zur Simulation des Ladungstransfers elektroaktiver Substanzen, die sich in einem dünnen Polymerfilm auf einer modifizierten Elektrode bewegen.

Das von Mirčeski eingesetzte Modell ist, wie schon das Adsorptionsmodell zuvor (Abschnitt 7.1.8), als Spezialfall bereits in dem in Kapitel 3 hergeleiteten Ansatz enthalten. Darüber hinaus stellt die Integration einer Dünnschichtgeometrie mit einer einzelnen Arbeitselektrode in das numerische Konzept der Finite-Elemente-Methode keine zusätzliche Schwierigkeit dar. Im Vergleich zum Modell der semi-infiniten Diffusion beschreibt nun die Referenzlänge nicht mehr die Ausdehnung der Nernstschen Diffusionsschicht (siehe Gleichung (3.76)), sondern steht für den Abstand zwischen Elektrode und inerter Wand oder für die Dicke des Films auf der Elektrode. Die Vorgabe von homogenen Neumann-Randbedingungen (3.72) am rechten Rand der Modellgeometrie aus Abbildung 3.2(b) liefert somit quasi “umsonst” das Dünnschicht-Modell. Homogene Reaktionen und Oberflächenprozesse können nach wie vor ohne Einschränkung eingebracht werden.

Erste experimentelle Ergebnisse zeigen, dass das Dünnschicht-Modell auch zur Unterstützung von SECM-Experimenten eingesetzt werden kann (Details siehe S.140).

Abbildung 7.21 zeigt simulierte CV-Kurven für den einfachen Elektronentransfer aus *Beispiel 1* für unterschiedliche Schichtdicken. Als Elektrodenoberfläche wurde $A = 0.5 \text{ cm}^2$ verwendet, woraus sich die Volumina der Schichten gemäß $V = A \cdot \delta$ berechnen. Weitere Modellparameter waren $T = 298 \text{ K}$, $c_A^0 = 10^{-3}$, $c_B = 0 \text{ mol} \cdot \text{l}^{-1}$, $D_A = D_B = 10^{-5} \text{ cm}^2 \cdot \text{s}^{-1}$, $E^0 = 0.25 \text{ V}$, $\alpha = 0.5$. Die CV-Anregung wurde von 0 bis 0.5 V bei $v = 2 \cdot 10^{-3} \text{ Vs}^{-1}$ durchgeführt. Die Ergebnisse wurden mit der RothRos3P-Klasse bei einer relativen Fehlertoleranz von 2% und den Werten $\text{ATOL}_\chi = 1$, $\text{ATOL}_A = 10^{-10}$ und $\text{ATOL}_B = 10^{-4}$ erzeugt.

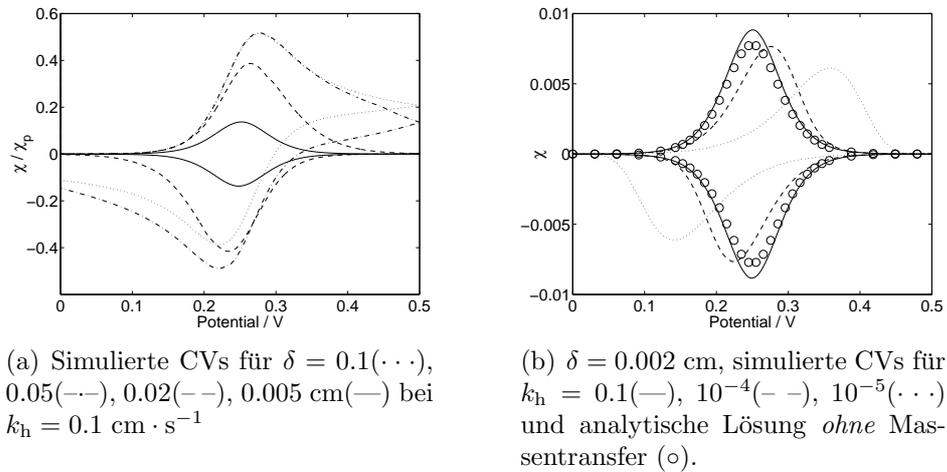


Abbildung 7.21: (*Beispiel 9*) CV-Signale des Modells einer Dünnschichtzelle bei unterschiedlichen Schichtdicken.

Abbildung 7.21(a) verdeutlicht die Veränderung der Signale beim Übergang von semiinfiniten zu finiten Diffusionsbedingungen. Der Einfluss der Diffusion auf die Form der CVs wird mit abnehmender Schichtdicke geringer, wodurch sich schließlich ein symmetrisches Stromsignal ausbildet. Die dazugehörigen Konzentrationsprofile des Edukts sind in Abbildung 7.22 gegeben. Die Konturen zeigen, dass bei einer Schichtdicke von $\delta = 0.1$ cm der Massentransfer vor der Elektrode noch diffusionskontrolliert verläuft. Mit abnehmender Schichtdicke wird der Einfluss der Diffusion geringer. Die Profile zeigen schließlich für einen festen Zeitpunkt eine nahezu konstante Konzentrationsverteilung in der Zelle (Abbildung 7.22(d)).

In Abbildung 7.21(b) sind simulierte CV-Signale und die analytische Lösung[51]

$$\chi_{\text{rev}} = \frac{L_{\text{ref}}}{D_{\text{max}}A} \cdot \frac{FvV}{RT} \cdot \frac{\exp\left(\frac{F}{RT}(E - E^0)\right)}{\left[1 + \exp\left(\frac{F}{RT}(E - E^0)\right)\right]^2} \quad (7.25)$$

eines reversibel verlaufenden Elektronentransfers in einer Dünnschichtzelle der Dicke $\delta = 0.002$ cm ($V = 1 \mu\text{l}$) abgebildet. Die analytische Gleichung (7.25) wird dabei *ohne* Berücksichtigung des Massentransfers in der Schicht hergeleitet. Abbildung 7.21(b) zeigt, dass für $k_h = 0.1 \text{ cm} \cdot \text{s}^{-1}$ der Strom des numerischen Modells leicht oberhalb der analytischen Kurve verläuft. Dies kann auf einen (kleinen) Stromanteil zurückgeführt werden, welcher durch den Massenfluss in der Schicht entsteht. Im analytischen Modell wird dieser Anteil vollständig vernachlässigt.

In Abbildung 7.21(b) sind zudem CV-Signale für langsamere Elektronentransfers abgebildet. Man erkennt das für quasireversible und irreversible Vorgänge typische Auseinanderlaufen der Strompeaks.

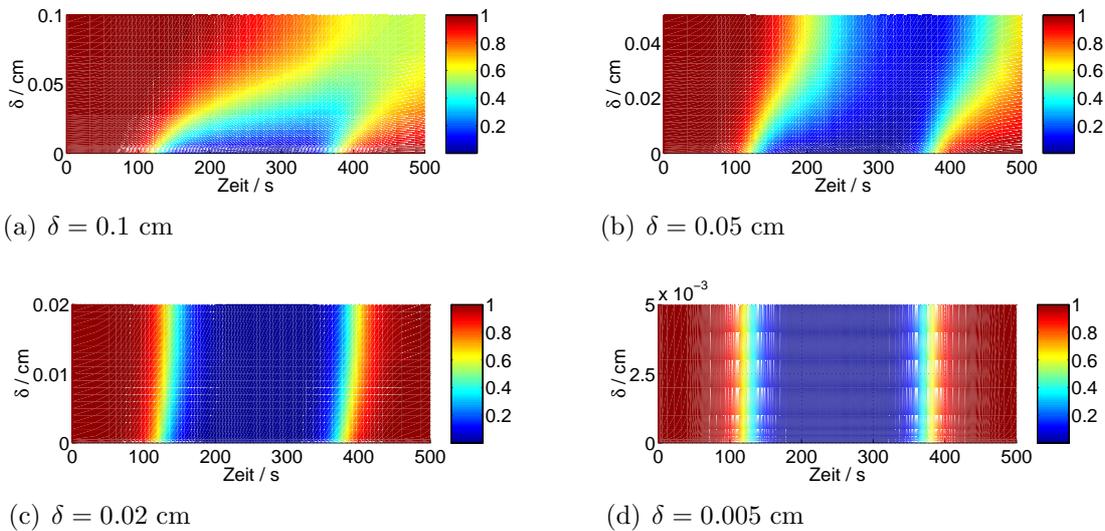


Abbildung 7.22: (*Beispiel 9*) Konzentrationsprofil des Edukts in Dünnschichtzellen mit abnehmender Schichtdicke δ .

Im Rahmen der DFG-Forschergruppe “*Analysis and modeling of diffusion/dispersion-limited reactions in porous media*” werden im Arbeitskreis Speiser elektrochemische Prozesse an Wasser-NAPL⁵-Phasengrenzen mit Hilfe der SECM-Technologie untersucht. In ersten Experimenten wurde dafür eine Mikroelektrode (Durchmesser $d = 200 \mu\text{m}$) von oben an eine Phasengrenze herangeführt, die zwischen einer wässrigen Kaliumhexacyanoferrat- (obere Phase) und einem Dichlormethan-Elektrolyten ohne elektroaktiver Substanz (untere Phase) entsteht. Unter semi-infiniten Diffusionsbedingungen erfolgt der elektrochemisch induzierte Massentransport des Hexacyanoferrat-Ions zur Elektrodenoberfläche hin in einem Diffusionsfeld, in dem aufgrund des geringen Durchmessers der Elektrode bereits laterale Diffusionsanteile an Bedeutung gewinnen (“*edge diffusion*” [218]). Wird die Elektrode jedoch nahe genug an die für das Hexacyanoferrat-Ion undurchlässige Dichlormethan-Phasengrenze gesteuert, kann von einem finiten und näherungsweise *linearen* Diffusionsfeld vor der Elektrode ausgegangen werden. Die laterale Komponente des Massentransports wird vernachlässigt, und an der Phasengrenze beschreiben homogene Neumann-Randbedingungen den Massenfluss. Somit kann das hier verwendete Dünnschicht-Modell (Abbildung 7.21) zumindest qualitative Aussagen über das elektrochemische Verhalten des Zwei-Phasen-Systems liefern.

Abbildung 7.23 zeigt experimentelle (linke Spalte) und simulierte (rechte Spalte) Cyclovoltammogramme einer Elektrodenannäherung.⁶ Ein Vergleich der Kurven zeigt, dass

⁵NAPL steht für *non-aqueous phase liquids*.

⁶Die experimentellen Stromkurven wurden von Patric Beuter (Arbeitskreis Speiser) aufgenommen.

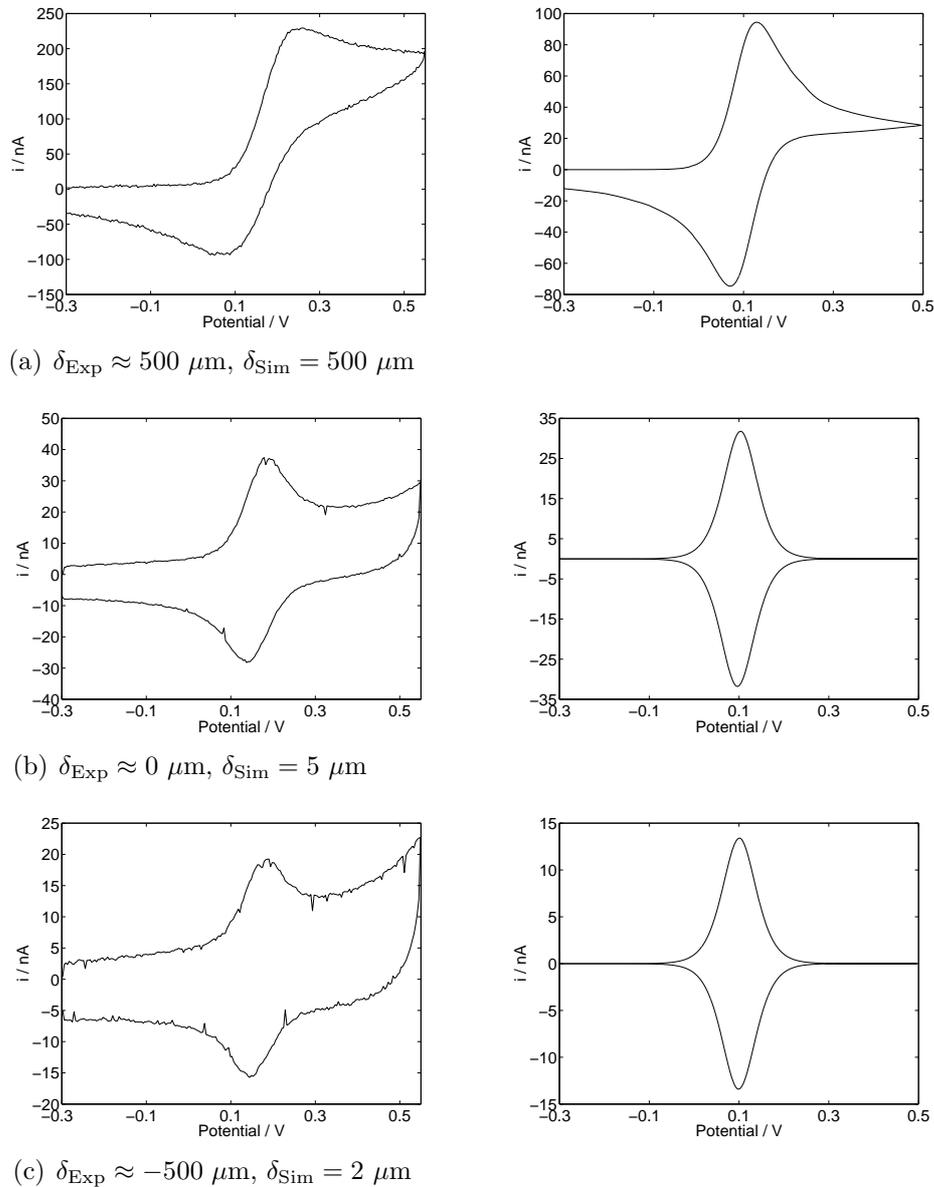


Abbildung 7.23: Experimentelle (links) und simulierte (rechts) CVs ($v = 0.2 \text{ V s}^{-1}$) einer Elektrodenannäherung an eine Wasser (0.1 M NaSO_4 , $1 \text{ mM K}_4[\text{Fe}(\text{CN})_6]$) / CH_2Cl_2 ($0.1 \text{ M NBu}_4\text{PF}_6$)-Phasengrenze. Der Elektrodenabstand zur Phasengrenze wurde im Experiment visuell für $\delta_{\text{Exp}} \approx 0 \mu\text{m}$ festgelegt. Der Abstand des Modells (δ_{Sim}) wurde für $\delta_{\text{Sim}} = 5$ und $2 \mu\text{m}$ anhand des Peakstroms ermittelt. Die experimentellen Daten enthalten noch einen Grundstromanteil, der bei der Peakstromanpassung in den Simulationen berücksichtigt wurde. Die Parameter zur Simulation betragen: $T = 293.15 \text{ K}$, $c^0 = 1 \text{ mM}$, $D = 6 \cdot 10^{-6} \text{ cm}^2 \text{ s}^{-1}$ (für Edukt und Produkt), $E^0 = 0.1 \text{ V}$, $\alpha = 0.5$, $k_h = 0.1 \text{ cm} \cdot \text{s}^{-1}$.

sich bei einem Abstand der Elektrode von $\delta \approx 500 \mu\text{m}$ von der Phasengrenze (Abbildung 7.23(a)) die experimentelle und simulierte Kurven noch deutlich unterscheiden. Der höhere Stromfluss im Experiment ist auf den lateralen Diffusionsanteil (siehe oben) zurückzuführen, der im Modell nicht berücksichtigt wird.

In unmittelbarer Nähe zur Dichlormethan-Phase (Abbildung 7.23(b)) wird die Form der experimentellen Kurve zunehmend symmetrisch und ähnelt dem simulierten Cyclovoltammogramm des Dünnschicht-Modells. Die Simulation zeigt einen vergleichbaren Peakstrom für eine Elektrode-Phasengrenze-Entfernung von $\delta = 5 \mu\text{m}$. Wird die Elektrode, durch die Phasengrenze hindurch, in die Dichlormethan-Phase abgesenkt (Abbildung 7.23(c)), bleibt die Form des Voltammogramms erhalten, während der Peakstrom weiter abnimmt. Die Simulation liefert für $\delta = 2 \mu\text{m}$ einen vergleichbaren Peakstrom, was auf einen verbliebenen dünnen Film der wässrigen Phase vor der Elektrode hindeutet.

Das Dünnschicht-Modell dieser Arbeit ermöglicht somit im vorliegenden Fall eine qualitative Abschätzung der Schichtdicke der wässrigen Phase vor der Elektrode. Eine quantitative Analyse des Systems verlangt jedoch vorzugsweise nach einem höherdimensionalen (2D oder 3D) Modell, in dem der laterale Massentransport enthalten ist.

7.1.10. *Beispiel 10: Chronoamperometrie in einer Dünnschichtzelle mit zwei Arbeitselektroden*

In vielen Elektrochemilumineszenz-Systemen rekombinieren elektrochemisch generierte Radikationen eines Stoffes. Dabei wird ein elektronisch angeregtes Teilchen erzeugt, welches anschließend Licht freisetzt. Zur Untersuchung solcher ECL-Systeme werden häufig DPCA-Experimente an einer einzelnen Arbeitselektrode (siehe Abschnitt 7.1.2) durchgeführt. Andere Experimente hingegen verwenden Dünnschichtzellen mit zwei sich gegenüberliegenden Arbeitselektroden[51].

In solchen Doppelektroden-Dünnschichtzellen werden an den Elektrodenoberflächen beide Radikale räumlich getrennt generiert. Diese diffundieren anschließend in der Zelle aufeinander zu und rekombinieren unter Lichtemission. Der ECL-Prozess kann durch den Mechanismus



beschrieben werden. Der Übergang der angeregten Spezies A^* in den Grundzustand geschieht meistens sehr schnell, sodass stattdessen die beiden Folgereaktionen zusammengefasst werden können



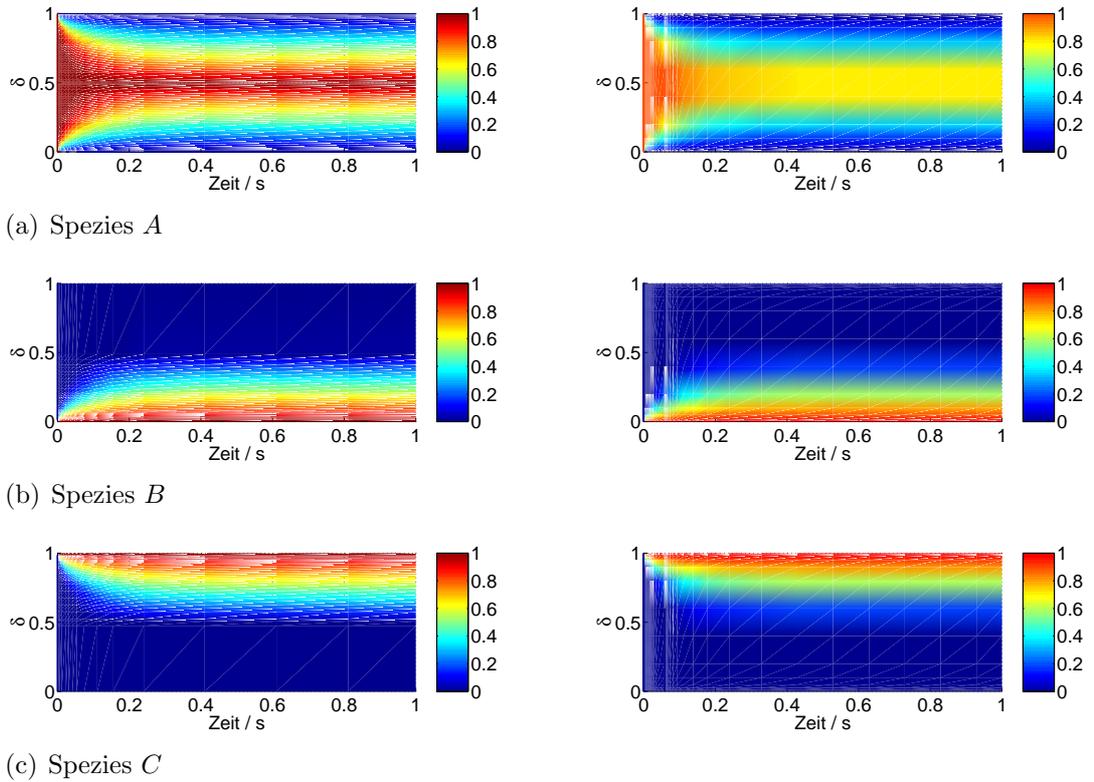
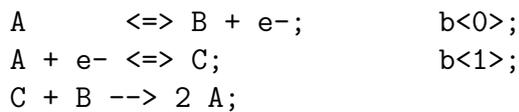


Abbildung 7.24: (*Beispiel 10*) Mit adaptivem (links, 6.3 s) und festem (rechts, 0.39 s) FE-Gitter simulierte Konzentrationsprofile in einer Doppелеlektroden-Dünnschichtzelle.

Mit $B = A^{\bullet+}$ und $C = A^{\bullet-}$ lässt sich der Mechanismus in der Eingabsprache von Ecco gemäß



formulieren, wobei das emittierte Photon ignoriert wird.

Die Rekombination der beiden Radikale erfolgt bei gleicher Diffusionsgeschwindigkeit in der Mitte der Zelle. Dort entsteht ein steiler “Konzentrationswall” des Ausgangsstoffes A, welcher bei der numerischen Berechnung durch eine angemessene Gitteranpassung approximiert werden muss. Ein festes Gitter, welches nur zu den Elektrodenoberflächen hin verfeinert ist, kann den Prozess im Inneren der Zelle nicht zufriedenstellend auflösen. Abbildung 7.24 zeigt die Konzentrationsprofile des ECL-Mechanismus, die mit einem adaptiven (linke Spalte) und einem festen FE-Gitter (rechte Spalte) berechnet wurden. In Abbildung 7.24(a) erkennt man den steilen Konzentrationswall des Ausgangsstoffes in

der Mitte der Zelle, der mit Hilfe des adaptiven Ansatzes (links) im Vergleich zum festen FE-Gitter (rechts)⁷ genauer wiedergegeben wird. Die Konzentrationsprofile an den Rändern (Abbildungen 7.24(b) und (c)) unterscheiden sich hingegen weniger deutlich. Doch auch hier ist das adaptive Gitter, insbesondere zu Beginn des Experiments, dem festen Gitter überlegen.

Eine ähnlich gute Auflösung aller Konzentrationen *zu späteren Zeitpunkten* (Abbildung 7.24, $t > 0.1$ s) konnte mit festen Gittern ab ca. 100 Elementen erreicht werden. Die Rechenzeiten waren dabei immer noch geringer als die des adaptiven Verfahrens. Eine vergleichbar gute Approximation der Konzentrationen *an den ersten Zeitpunkten* der Simulation nach dem Potentialpuls (Abbildung 7.24, $t < 0.1$ s) war durch die festen FE-Gitter innerhalb der Rechenzeit des adaptiven Ansatzes nicht mehr möglich.

Insgesamt kann man somit sagen, dass eine adaptive Gittersteuerung universeller anwendbar ist und im Allgemeinen bei angemessener Rechenzeit die zuverlässigeren Ergebnisse liefert.

Die Modellparameter der Simulation betragen $T = 293.15$ K, $A_0 = A_1 = 6$ mm², $\delta = 2 \cdot 10^{-3}$ cm, $c_A^0 = 10^{-4}$, $c_B^0 = c_C^0 = 0$ mol · l⁻¹, $D_A = D_B = D_C = 10^{-6}$ cm²s⁻¹, $E_0^0 = 0.25$ V, $E_1^0 = -0.25$ V, $\alpha_0 = \alpha_1 = 0.5$, $k_{h,0} = k_{h,1} = 0.1$ cm · s⁻¹, $k_{f,2} = 10^{10}$ l · mol⁻¹s⁻¹. Die Potentialpulse dauerten eine Sekunde und entsprachen an der ersten Elektrode (Γ_0) $E = 0.5$ V und an der zweiten Elektroden (Γ_1) $E = -0.5$ V. Die Ergebnisse wurden mit der Ros3P-Methode bei 2%-iger relativer Fehlertoleranz und $ATOL_\chi = 10^{-10}$, sowie $ATOL_A = ATOL_B = ATOL_C = 10^{-4}$ erzeugt.

7.2. Stromkontrollierte Experimente

In Abschnitt 7.1 wurden potentialkontrollierte Experimente betrachtet. Zur Berechnung der Randbedingungen kann dabei stets das, von der Anregungsfunktion exakt vorgegebene Elektrodenpotential verwendet werden. Die Berechnung des Stroms als Lösungsvariable erfolgt über das Strom-Potential-Residuum mit Hilfe des Elektrodenpotentials und der Konzentrationsverteilung der elektroaktiven Spezies an der Elektrodenoberfläche. Letztere wird durch das Einbringen von potentialabhängigen Randbedingungen der Form (3.30) berechnet.

Im Vergleich dazu wird die mathematische Behandlung von stromkontrollierten Experimenten häufig als weniger kompliziert erachtet, da sich für den speziellen Fall eines einfachen Elektronentransfers die Randbedingungen der Substanzen unmittelbar aus der kontrollierten Gesamtstromdichte ergeben[51]. Doch schon bei einem Reaktionsmechanismus, der zwei Elektronentransferreaktionen enthält, können diese Randbedingungen nicht mehr verwendet werden, denn sie enthalten keine Informationen über die Verteilung der Partialströme der Elektronentransfers auf den Massenfluss der einzelnen elektroaktiven Substanzen. Die potentialabhängigen Massenfluss-Gleichungen (3.30) sowie

⁷Das feste Gitter entspricht dem zu den Rändern hin verfeinerten Ausgangsgitter des adaptiven Verfahrens.

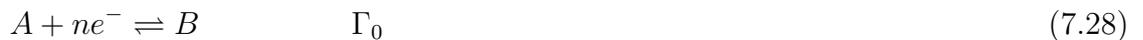
das Strom-Potential-Residuum gelten dagegen weiterhin.

Eine allgemeingültige Strategie zur Simulation stromkontrollierter Experimente ist es daher, aus dem Strom-Potential-Residuum bei vorgegebenen Strom und bekannter Konzentrationsverteilung an der Elektrode das dazugehörige Elektrodenpotential zu ermitteln. Letzteres wird anschließend über die Randgleichungen (3.30) zur Berechnung der neuen Konzentrationswerte herangezogen.

In Abschnitt 6.1 wurde das Elektrodenpotential über das Strom-Potential-Residuum als Lösungsvariable in das elektrochemische Modell eingeführt. Das Strom-Potential-Residuum stellt dabei bezüglich des Elektrodenpotentials eine *nicht-lineare* algebraische Gleichung dar. Diese Gleichung muss in jedem Zeitschritt der Simulation durch das Rosenbrock-Verfahren gelöst werden.

Eine wichtige Konsequenz ist, dass die Randwerte der Konzentrationen nun zusätzlich von der Genauigkeit des Verfahrens abhängen, mit der das Potential zuvor ermittelt werden konnte. Im Vergleich zu den Berechnungen aus Abschnitt 7.1 stellt die numerische Simulation von stromkontrollierten Experimenten somit im Allgemeinen eine anspruchsvollere Aufgabe dar. Dies könnte ein Grund dafür sein, weshalb in der Literatur scheinbar erst in einem Fall stromkontrollierte Simulationen an einem komplizierteren Reaktionssystem durchgeführt wurden[219].

Analytische Lösungen existieren für einige einfachere Reaktionsmechanismen und Anregungsfunktionen jeweils für die Grenzfälle eines reversiblen und irreversiblen Elektronentransfers[59]:



wobei der Potentialverlauf im reversiblen Fall durch die Nernst-Gleichung

$$E(t) = E^0 + \frac{RT}{nF} \ln \frac{c_A(t, 0)}{c_B(t, 0)} \quad (7.29)$$

beschrieben wird. Für einen irreversiblen Prozess gilt

$$E(t) = E^0 + \frac{RT}{\alpha nF} \ln \frac{nFAk_h}{i_0} + \frac{RT}{\alpha nF} \ln \frac{c_A(t, 0)}{f(t)} \quad (7.30)$$

mit einer Anregungsfunktion der allgemeinen Form

$$i(t) = i_0 \cdot f(t) \quad (7.31)$$

Der genaue Verlauf der Randkonzentrationen $c_A(t, 0)$ und $c_B(t, 0)$ wird durch $f(t)$ und mögliche Folgereaktionen beeinflusst[59].⁸

⁸Die meisten der folgenden Simulationsbeispiele wurden von Isabel Morales Moreno (Grupo de Investigación de Electroquímica Teórica y Aplicada de la Universidad de Murcia, Spanien) während eines Aufenthaltes im Arbeitskreis Speiser im Jahre 2005 zusammengestellt und an einem Prototypen des Ros3P-Algorithmus getestet. Sämtliche Ausdrücke für die analytischen Lösungen basieren auf einer von I.M.Moreno verfassten Zusammenstellung der Arbeiten[59, 220, 221]. Sie werden in den folgenden Abschnitten an den entsprechenden Stellen zur Übersicht angegeben.

Stromkontrollierte elektroanalytische Verfahren werden üblicherweise als *chronopotentiometrische* Methoden bezeichnet[51], wobei das Elektrodenpotential als Funktion der Zeit bestimmt wird. Neben der Kurvenform stellt dabei insbesondere die sogenannte *Transitionszeit* die charakteristische Messgröße eines chronopotentiometrischen Experiments dar. Die Transitionszeit bestimmt das Ende eines elektrochemischen Stoffumsatzes (z.B. in (7.29): $c_A(t, 0) \rightarrow 0 \text{ mol} \cdot \text{l}^{-1}$), welches sich im Falle einer Reduktion der Form (7.28) in einem starken Abfall des Elektrodenpotentials bemerkbar macht.

7.2.1. Beispiel 11: Chronopotentiometrie mit konstanter und komplexen Anregungsfunktionen — Einelektronentransfer

Um die Eigenschaften der Rosenbrock-Verfahren für die Simulation stromkontrollierter Modelle zu vergleichen, betrachten wir zunächst den einfachen Elektronentransfer



unter semi-infiniten Diffusionsbedingungen an einer einzelnen Arbeitselektrode. Weiterhin wählen wir eine konstante Anregungsfunktion (klassische Chronopotentiometrie, $|i_0| = 20 \mu\text{A}$).⁹ Mit den Definitionen

$$N_s = \frac{2i_0}{nFAD_A^{1/2}c_A^0}, \quad \mu = \frac{c_B^0}{c_A^0}, \quad \gamma = \left(\frac{D_A}{D_B}\right)^{1/2}, \quad \Omega = \omega t \quad (7.33)$$

folgt für die Randkonzentrationen

$$\begin{aligned} \frac{c_A(t, 0)}{c_{\max}} &= 1 - N_s \left(\frac{1}{\pi}\right)^{1/2} \\ \frac{c_B(t, 0)}{c_{\max}} &= \mu + \gamma N_s \left(\frac{1}{\pi}\right)^{1/2} \end{aligned} \quad (7.34)$$

woraus sich mit den Gleichungen (7.29) bzw. (7.30) die analytischen E - t -Kurven ergeben[59].

Die Dauer der Anregungsfunktion soll der Transitionszeit $t_{\max} = \tau = 0.18 \text{ s}$ entsprechen.¹⁰ Die dazugehörigen Modellparameter betragen $T = 293.15 \text{ K}$, $A = 0.78 \text{ mm}^2$, $c_A^0 = 4 \cdot 10^{-3}$, $c_B^0 = 0 \text{ mol} \cdot \text{l}^{-1}$, $D_A = D_B = 10^{-5} \text{ cm}^2\text{s}^{-1}$, $E^0 = 0.4 \text{ V}$, $\alpha = 0.5$.

Abbildung 7.25 zeigt die Simulationsergebnisse der vier Verfahren für das Testsystem. Die absoluten Fehlertoleranzen aller Variablen betragen in den hier abgebildeten Ergeb-

⁹Reduktionsströme werden in der ModSim-Software nach der IUPAC-Konvention[79] stets mit einem negativen Vorzeichen versehen. Dies ist insbesondere bei der Definition der Anregungsfunktionen zu beachten (hier also $i_0 = -20 \mu\text{A}$).

¹⁰Die Transitionszeit ist im Allgemeinen unbekannt. Sie muss daher vorab in Testsimulationen ermittelt werden.

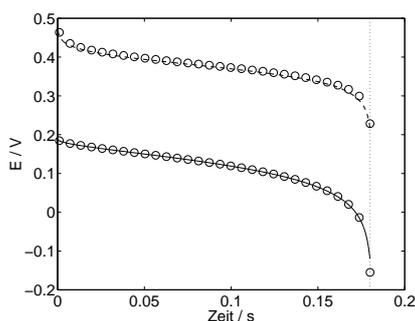
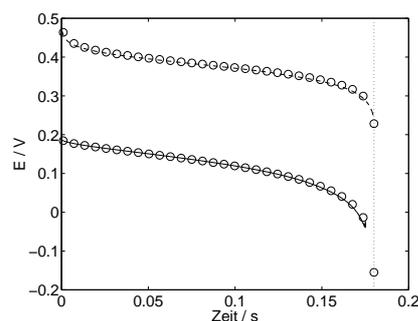
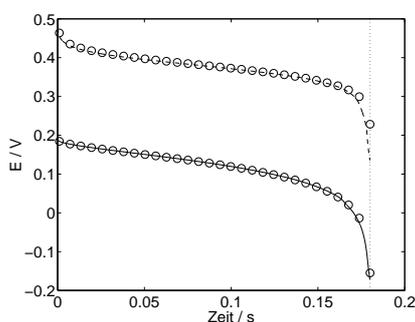
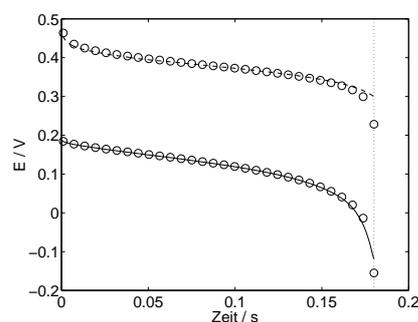
(a) Rowda3: $\text{RTOL}_{\text{rev}} = 0.2\%$ (40 s),
 $\text{RTOL}_{\text{irr}} = 0.5\%$ (27 s)(b) RodasP: $\text{RTOL}_{\text{rev}} = 1\%$ (29 s),
 $\text{RTOL}_{\text{irr}} = 1\%$ (Abbruch)(c) Ros3P: $\text{RTOL}_{\text{rev}} = 0.1\%$ (90 s),
 $\text{RTOL}_{\text{irr}} = 0.1\%$ (52 s)(d) Ros2: $\text{RTOL}_{\text{rev}} = 0.2\%$ (38 s),
 $\text{RTOL}_{\text{irr}} = 0.5\%$ (20 s)

Abbildung 7.25: (*Beispiel 11*) Analytische (\circ) und simulierte CP-Kurven ($k_h = 0.1$ (---), $k_h = 10^{-4} \text{ cm} \cdot \text{s}^{-1}$ (—)) der vier Rosenbrock-Verfahren bei konstantem Strom, $i_0 = -20 \mu\text{A}$. Die Transitionszeit von $\tau = 0.18 \text{ s}$ ist durch die vertikale, gepunktete Linie markiert. Die Simulationsdauer ist jeweils in Klammern angegeben.

nissen stets $\text{ATOL} = 10^{-5}$. Die relativen Fehlertoleranzen der Simulationen sind unter den einzelnen Abbildungen angegeben. Sie beschreiben den kleinsten Wert, bis zu dem eine Konvergenz in den numerischen Lösungskurven beobachtet wurde. Für kleinere RTOL-Werte brechen die Algorithmen entweder die Simulation ab oder liefern von den vorher konvergenten Potentialkurven stark abweichende Ergebnisse. Ein solides Konvergenzverhalten der Verfahren konnte mit Hilfe einer genaueren Anpassung der Fehlerkalibrierung durch die ATOL-Werte: $\text{ATOL}_p = 10^{-10}$, $\text{ATOL}_A = 10^{-10}$ und $\text{ATOL}_B = 1$ erzielt werden (Abbildung 7.26). Der große ATOL_B -Wert wird dabei durch die Anfangswerte $c_B^0(0, x) = 0 \text{ mol} \cdot \text{l}^{-1}$ motiviert, welche eine überwiegend relative Fehlermessung für die Konzentration des Produkts zu Beginn verhindert (siehe dazu auch die Diskussion bezüglich des CV-Stroms in Abschnitt 7.1.4).

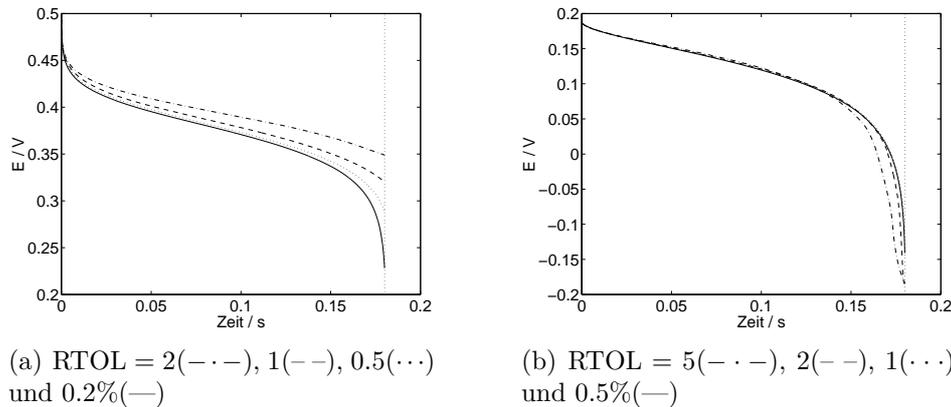


Abbildung 7.26: (*Beispiel 11*) Konvergenzverhalten der Rowda3-Methode im (a) reversiblen und (b) irreversiblen Grenzfall (Abbildung 7.25(a)). ATOL-Werte: $ATOL_p = 10^{-10}$, $ATOL_A = 10^{-10}$, $ATOL_B = 1$.

Es ist bekannt, dass für Rosenbrock-Verfahren eine adaptive Fehlerschätzung auf der Basis des eingebetteten Ansatzes (siehe Abschnitt 6.2.2) für moderate Fehlertoleranzen von 5 – 0.1% zufriedenstellende Ergebnisse liefert[60]. In den hier durchgeführten Simulationen zeigte sich, dass die $E-t$ -Kurven in einer angemessenen Rechenzeit oft nur bis zu einem minimalen RTOL-Wert von 0.2% erzielt werden konnten. Für kleinere Werte brachen die Algorithmen oft wegen zu kleinen Schrittweiten die Simulation ab.

In Abbildung 7.25 ist zu erkennen, dass die simulierten Kurven aller Verfahren für $t < \tau$ mit den analytischen Lösungen übereinstimmen. Am Ende der Kurven $t \approx \tau$ lassen sich für den irreversiblen Prozess (untere Kurven) weiterhin gute Resultate beobachten. Das RodasP-Verfahren bricht jedoch vor Erreichen der Transitionszeit aufgrund zu kleiner Schrittweiten ab (Abbildung 7.25(b)). Vergleicht man für den irreversiblen Fall die verwendeten RTOL-Werte, fällt auf, dass bis auf Ros3P alle Verfahren mit moderaten Toleranzen von 1 – 0.5% sehr gute Ergebnisse erzielen. Die beste Übereinstimmung wird dabei von Rowda3 erreicht (Abbildung 7.25(a)).

Im reversiblen Fall (obere Kurven) liefern Rowda3 und RodasP die besten Resultate. Für Ros3P und Ros2 sind jeweils deutlichere Abweichungen für $t \approx \tau$ zu erkennen. Während Ros3P bei der geringsten Fehlertoleranz von 0.1% die analytische Lösung bereits deutlich unterschätzt, weicht die Ros2-Lösung bei $t = 0.18$ mit immer noch zu großen Potentialwerten ab.

Die Rechenzeiten¹¹ der Verfahren sind für beide Grenzfälle in Klammern unter den Abbildungen angegeben. Beachtet man zudem die Qualität der erzielten Ergebnisse, stellen sich insgesamt Rowda3 und RodasP als die für das stromkontrollierte Modell geeigne-

¹¹siehe Anhang A.4

ten Verfahren heraus. RodasP ist dabei zwar die effizientere Methode, bricht jedoch im irreversiblen Fall vorzeitig ab. In Simulationen mit komplizierten Anregungsfunktionen zeigte sich zudem, dass RodasP insbesondere bei singulären Anfangswerten keine zufriedenstellende Resultate liefern konnte. Rowda3 hingegen ist auch in solchen Fällen den anderen Methoden überlegen.

Anfangssingularitäten treten bei polynomiellen Anregungsfunktionen ($i(t) = i_0 t^u$, $i(t) = i_0 t^u e^{\omega t}$, siehe Tabelle 3.1) auf, für die zu Beginn der Strom den Wert Null annimmt, $i(0) = 0$ A.¹² Weiterhin gilt für die Anfangskonzentration des Produkts meist $c_B^0(0) = 0$ mol · l⁻¹, sodass der Wert des Elektrodenpotentials für $t = 0$ eine Polstelle aufweist (siehe z.B. Gleichungen (7.29) und (7.30)). In manchen Fällen besitzt die Anregungsfunktion dort sogar selbst eine Polstelle (falls der Exponent $u < 0$), sodass der Beginn der numerischen Simulation an einem späteren Zeitpunkt $0 < t \ll 1$ und/oder mit modifizierten Randkonzentrationen $0 < c_B^0(0, 0) \ll 1$ durchgeführt werden muss.

Abbildung 7.27 zeigt die Simulationsergebnisse der Rowda3-Methode für eine polynomielle (a) und eine polynomiell-exponentielle (b) Anregungsfunktion, bei denen Singularitäten für $t=0$ auftreten. Alle anderen Verfahren lieferten, bei längeren Rechenzeiten, höchstens äquivalente Ergebnisse. Im polynomiellen Fall ($i(t) = i_0 t^u$, $i_0 = 100$ μA, $u = 2$) gelten für die Konzentrationen die analytischen Lösungen[59]

$$\begin{aligned} \frac{c_A(t, 0)}{c_{\max}} &= 1 - N_s t^{u+1/2} \frac{\Gamma(u+1)}{2\Gamma(1+(2u+1)/2)} \\ \frac{c_B(t, 0)}{c_{\max}} &= \mu - \gamma N_s t^{u+1/2} \frac{\Gamma(u+1)}{2\Gamma(1+(2u+1)/2)} \end{aligned} \quad (7.35)$$

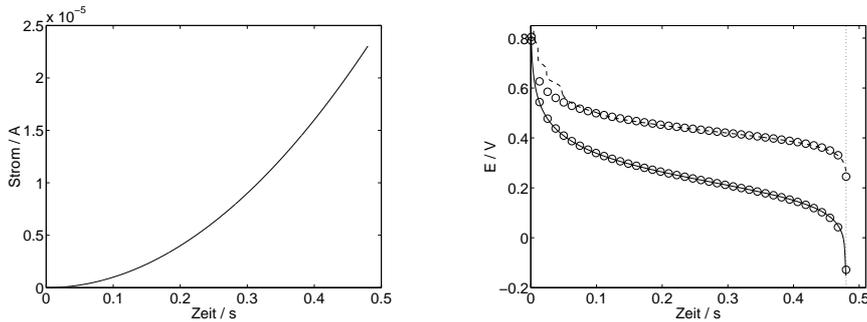
($\Gamma(\cdot)$ beschreibt die Gamma-Funktion). Für die polynomiell-exponentielle Anregung ($i(t) = i_0 t^u e^{\omega t}$, $i_0 = 5$ μA, $u = -0.4$, $\omega = 0.5$ s⁻¹) existieren die Reihenlösungen[59]

$$\begin{aligned} \frac{c_A(t, 0)}{c_{\max}} &= 1 - N_s t^{u+1/2} S(\Omega) \\ \frac{c_B(t, 0)}{c_{\max}} &= \mu + \gamma N_s t^{u+1/2} S(\Omega) \\ S(\Omega) &= \sum_{k=0}^{\infty} \left(\frac{\Gamma(\frac{1}{2} + \frac{6m+6k+4}{6})}{2\Gamma(1 + \frac{6m+6k+4}{6})} \frac{\Omega^k}{k!} \right), \quad m = u - 1/6 \end{aligned} \quad (7.36)$$

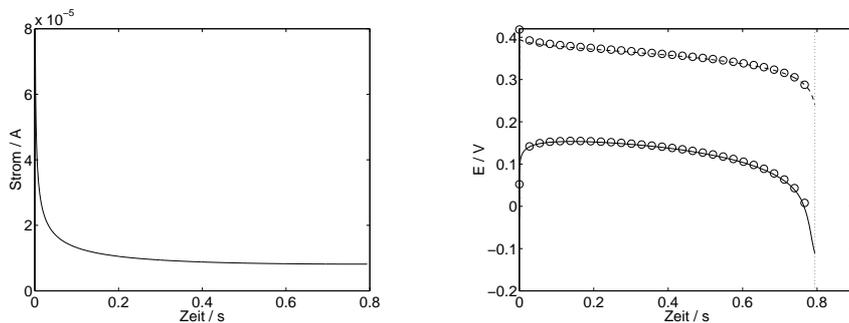
Im polynomiellen Fall (Abbildung 7.27(a)) müssen die Randkonzentrationen des Produkts auf einen kleinen Wert ($c_{\text{mod}} = 10^{-10}$ / dimensionslos) gesetzt werden, um überhaupt Ergebnisse zu erhalten. Andernfalls brachen alle Verfahren bereits zu Beginn die Berechnungen ab. Die Auswirkung der modifizierten Anfangs-Randwerte macht sich insbesondere im reversiblen Grenzfall bemerkbar. Dort fallen sowohl leichte Oszillationen

¹²Der Funktionswert $i(t)$ einer Strom-Zeit-Anregungsfunktion wird in Ampere angegeben. Demnach wird im Funktionsausdruck der Zeitwert t gegebenenfalls (z.B. $i(t) = i_0 t^u$) mit einer Sekunde normiert (vgl. auch Abbildungen 7.27 – 7.29).

7. Anwendungen



(a) Anregungsfunktion $i(t) = 100 \cdot t^2 \mu\text{A}$ (links) und Simulation (rechts) mit $\text{RTOL}_{\text{rev}} = 0.2\%$ (17 s), $\text{RTOL}_{\text{irr}} = 1\%$ (5 s), $c_{\text{mod}} = 10^{-10}$.



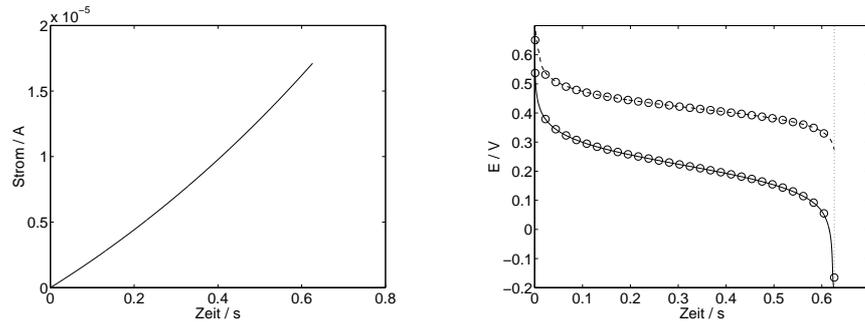
(b) Anregungsfunktion $i(t) = 5 \cdot t^{-0.4} e^{0.5 \cdot t} \mu\text{A}$ (links) und Simulation (rechts) mit $\text{RTOL}_{\text{rev}} = 0.2\%$ (90 s), $\text{RTOL}_{\text{irr}} = 1\%$ (17 s), $c_{\text{mod}} = 10^{-6}$, $t_{\text{init}} = 10^{-6}$.

Abbildung 7.27: (*Beispiel 11*) Analytische (o) und simulierte CP-Kurven ($k_{\text{h}} = 0.1$ (---), $k_{\text{h}} = 10^{-4} \text{ cm} \cdot \text{s}^{-1}$ (—)) von Rowda3 bei Systemen mit Anfangssingularitäten. Die Transitionszeiten sind durch vertikale, gepunktete Linien markiert. Die Simulationsdauer ist jeweils in Klammern angegeben.

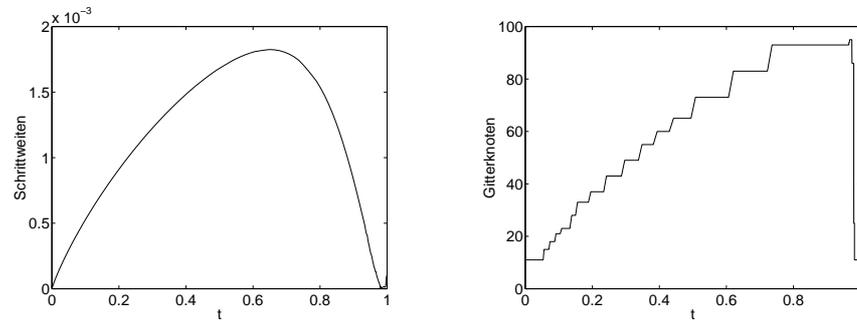
in der numerischen Kurve als auch eine große Abweichung von der analytischen Lösung auf. Die Fähigkeit der Methode, diese doch eher großen Störungen nach relativ kurzer Zeit zu dämpfen, ist jedoch bemerkenswert. Ab $t \approx 0.07 \text{ s}$ stimmt die Kurve mit der analytischen Lösung überein.

Im irreversiblen Grenzfall hingegen liefert Rowda3 bei einer moderaten 1%-igen Fehler-toleranz zu allen Zeiten exzellente Ergebnisse. Die Simulationsdauer beträgt hier nur fünf Sekunden.

Tritt eine zusätzliche Singularität in der Anregungsfunktion auf (Abbildung 7.27(b)), muss neben einem größeren Wert für die Anfangs-Randkonzentrationen ($c_{\text{mod}} = 10^{-6}$) zusätzlich eine von Null verschiedene Anfangszeit von $t_{\text{init}} = 10^{-6}$ (dimensionslos) gewählt werden. Dadurch wird die Polstelle der Anregungsfunktion übersprungen. Bis auf eine



(a) Anregungsfunktion $i(t) = 20 \cdot t^1 e^{0.5 \cdot t} \mu\text{A}$ (links), analytische (\circ) sowie simulierte Kurven (rechts) mit $\text{RTOL}_{\text{rev}} = 0.2\%$ (—, 25 s) und $\text{RTOL}_{\text{irr}} = 0.2\%$ (—, 26 s), $c_{\text{mod}} = 10^{-10}$, $t_{\text{init}} = 10^{-6}$.



(b) Zeitschrittweiten für irreversiblen Fall

(c) Entwicklung der Gitterknoten für irreversiblen Fall

Abbildung 7.28: (*Beispiel 11*) Simulationsergebnisse von Rowda3. Weitere Parameter siehe Text.

anfängliche, leichte Abweichung im reversiblen Fall, stimmen beide simulierten Kurven mit dem analytischen Modell überein.

In den Abbildungen 7.28 und 7.29 sind weitere Simulationsergebnisse diverser Anregungsfunktionen zusammengefasst. Abbildungen 7.28(b) und (c) zeigen exemplarisch die Veränderungen der Zeitschrittweiten und der Gitterknoten während einer Simulation. Man erkennt, dass an den Rändern des Zeitfensters sehr kleine Zeitschrittweiten gewählt werden müssen, um den Verlauf der Lösung an den Polstellen berechnen zu können.

Das FE-Gitter wird bis zum Ende der Simulation stetig verfeinert (Abbildung 7.28(c)). In der Nähe der Transitionszeit erfolgt eine schlagartige Vergrößerung auf das Ausgangsgitter zurück. Am Ende wird die Kontrolle der Zeitschrittweiten überwiegend von einem großen Fehlerwert für das Elektrodenpotential dominiert, was zu sehr kleinen Zeitschrittweiten führt. Die Konzentrationsprofile verändern sich hingegen nur noch geringfügig, sodass insgesamt eine gröbere Ortsdiskretisierung ausreicht.

Die analytischen Randkonzentrationen lauten für exponentielle (Abbildung 7.29(b))[59]

$$\begin{aligned}\frac{c_A(t, 0)}{c_{\max}} &= 1 - N_s \frac{1}{2\sqrt{\omega}} e^{\Omega} \operatorname{erf}(\Omega^{1/2}) \\ \frac{c_B(t, 0)}{c_{\max}} &= \mu + \gamma N_s \frac{1}{2\sqrt{\omega}} e^{\Omega} \operatorname{erf}(\Omega^{1/2})\end{aligned}\quad (7.37)$$

und sinusoidale (Abbildung 7.29(c)) Anregungsfunktionen[220]

$$\begin{aligned}\frac{c_A(t, 0)}{c_{\max}} &= 1 - N_s t^{1/2} G_s(\Omega) \\ \frac{c_B(t, 0)}{c_{\max}} &= \mu + \gamma N_s t^{1/2} G_s(\Omega)\end{aligned}\quad (7.38)$$

$$G_s(\Omega) = \sum_{k=0}^{\infty} \frac{(-1)^k \Omega^{2k+1}}{(2k+1)! p_{4k+3}}, \quad p_n = \frac{2\Gamma(1+n/2)}{(1+n)/2}$$

Insgesamt zeigen die Resultate in diesem Abschnitt, dass der Algorithmus im Falle eines einfachen Elektronentransfers für die getesteten Anregungsfunktionen zuverlässige Resultate liefert. Allerdings muss vorab in einer Konvergenzanalyse der numerischen Kurven eine geeignete Fehlerskalierung durch die Wahl der ATOL-Werte ermittelt werden. Zu große Fehlertoleranzen führen stets zu einer Fehlerfortpflanzung, welche sich insbesondere in einer Abweichung am Ende der Kurven bemerkbar macht. Die Transitionszeiten der Experimente können jedoch durch die Wahl moderat gewählter RTOL-Werte in angemessener Zeit erreicht werden.

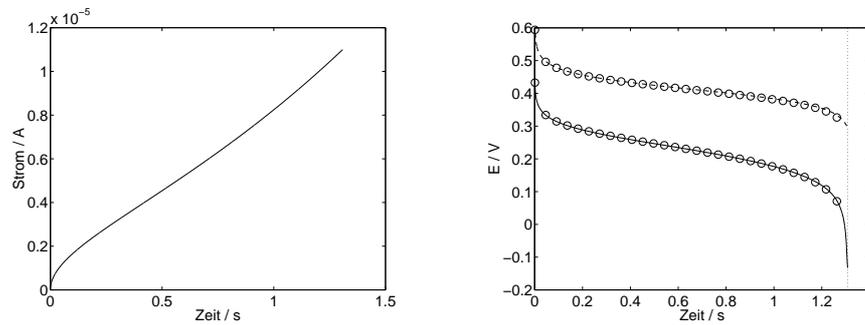
7.2.2. *Beispiel 12*: Chronopotentiometrie mit konstanter Anregungsfunktion — Einelektronentransfer mit Folgereaktion erster Ordnung

In diesem Abschnitt wird der VET aus *Beispiel 11* um eine reversibel verlaufende Folgereaktion erster Ordnung erweitert

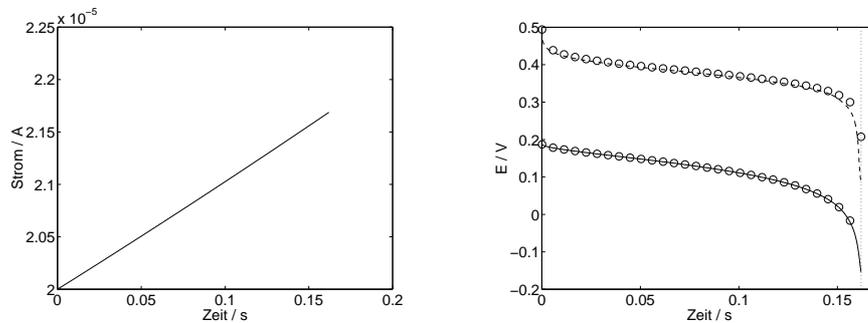


Bei konstanter Anregung folgt die analytische E - t -Kurve aus den Gleichungen[59]

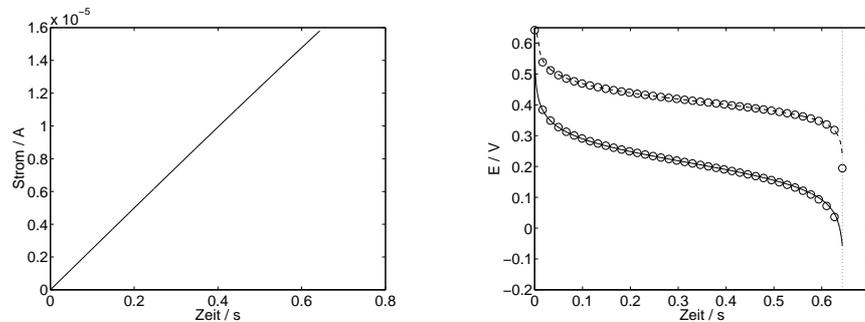
$$\begin{aligned}\frac{c_A(t, 0)}{c_{\max}} &= 1 - N_s \left(\frac{1}{\pi}\right)^{1/2} \\ \frac{c_B(t, 0)}{c_{\max}} &= \frac{1}{1 + \frac{1}{K_1}} \left(\frac{\mu}{K_1} + \gamma N_s t^{1/2} \left(\frac{1}{K_1 \sqrt{\pi}} + \frac{\operatorname{erf}(\sqrt{\pi})}{2\sqrt{\chi}} \right) \right)\end{aligned}\quad (7.40)$$



(a) Anregungsfunktion $i(t) = 5 \cdot t^{0.5} e^{0.5 \cdot t} \mu\text{A}$ (links) und Simulation (rechts) mit $\text{RTOL}_{\text{rev}} = 0.2\%$ (28 s), $\text{RTOL}_{\text{irr}} = 0.5\%$ (23 s), $c_{\text{mod}} = 10^{-10}$, $t_{\text{init}} = 10^{-10}$.



(b) Anregungsfunktion $i(t) = 20 \cdot e^{0.5 \cdot t} \mu\text{A}$ (links) und Simulation (rechts) mit $\text{RTOL}_{\text{rev}} = 0.2\%$ (50 s), $\text{RTOL}_{\text{irr}} = 0.5\%$ (13 s).



(c) Anregungsfunktion $i(t) = 50 \cdot \sin(0.5 \cdot t) \mu\text{A}$ (links) und Simulation (rechts) mit $\text{RTOL}_{\text{rev}} = 0.2\%$ (35 s), $\text{RTOL}_{\text{irr}} = 1\%$ (5 s), $c_{\text{mod}} = 10^{-10}$.

Abbildung 7.29: (*Beispiel 11*) Analytische (\circ) und simulierte CP-Kurven ($k_{\text{h}} = 0.1$ (---), $k_{\text{h}} = 10^{-4} \text{ cm} \cdot \text{s}^{-1}$ (—)) von Rowda3 bei verschiedenen Anregungsfunktionen. Die Transitionszeiten sind durch vertikale, gepunktete Linien markiert. Die Simulationsdauer ist jeweils in Klammern angegeben.

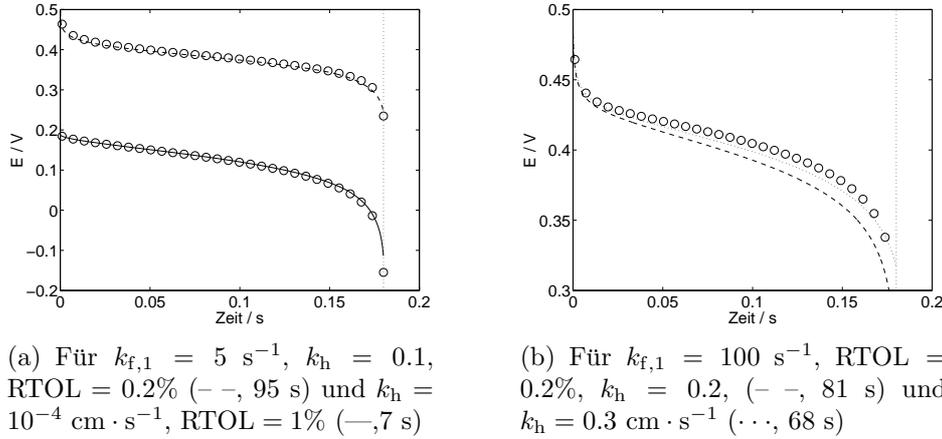


Abbildung 7.30: (*Beispiel 12*) Analytische (o) und simulierte CP-Kurven von Rowda3 bei einer Anregung von $i(t) = i_0 = 20 \mu\text{A}$. Die Transitionszeit ist durch die vertikale, gepunktete Linie markiert.

wobei die Definitionen

$$N_s = \frac{2i_0}{nFAD_A^{1/2}c_A^0}, \quad \mu = \frac{c_B^0 + c_C^0}{c_A^0}, \quad \gamma = \left(\frac{D_A}{D}\right)^{1/2}, \quad \chi = (k_{f,1} + k_{b,1})t \quad (7.41)$$

gelten ($D = D_B = D_C$).

Zusätzlich zu den Modellparametern aus *Beispiel 12*, wählen wir zur Simulation die Anfangskonzentration $c_C^0 = 0 \text{ mol} \cdot \text{l}^{-1}$. Die Gleichgewichtskonstante der Folgereaktion erhält den Wert $K_1 = 5$.

Mit RTOL-Werten von 5 bis 0.2% konnten konvergente E - t -Kurven für $\text{ATOL}_p = 10^{-10}$, $\text{ATOL}_A = 10^{-10}$, $\text{ATOL}_B = 1$ und $\text{ATOL}_C = 1$ erzielt werden.

Abbildung 7.30(a) zeigt die Simulationsergebnisse der Rowda3-Methode bei einer Geschwindigkeitskonstanten der Folgereaktion von $k_{f,1} = 5 \text{ s}^{-1}$. Wie schon in den Simulationen aus *Beispiel 11*, kann eine gute Übereinstimmung zwischen analytischer und numerischer Lösung festgestellt werden. Aus den Gleichungen (7.30), (7.34) und (7.40) geht hervor, dass die Folgereaktion keinen Einfluss auf den Potentialverlauf eines irreversiblen VETs haben kann.

Bei einer Geschwindigkeitskonstanten von $k_{f,1} = 5 \text{ s}^{-1}$ ist auch noch kein Einfluss auf die Lage und Form der reversiblen CP-Kurve zu beobachten. Bei höherer Geschwindigkeitskonstante (Abbildung 7.30(b)) reicht die in den Simulationen gewählte heterogene Geschwindigkeitskonstante von $k_{h,0} = 0.1 \text{ cm/s}$ nicht mehr aus, um den reversiblen Grenzfall anzunähern. Das Resultat kann durch einen Wert von $k_{h,0} = 0.3 \text{ cm/s}$ und einer genaueren Approximation der Produktkonzentrationen ($\text{ATOL}_B = 10^{-4}$ und $\text{ATOL}_C = 10^{-4}$) leicht verbessert werden. Insgesamt muss allerdings festgehalten wer-

den, dass mit den Butler-Volmer-Randgleichungen des numerischen Modells aus Abschnitt 6.1 und den hier eingesetzten Rosenbrock-Verfahren die Simulation einer schnellen Folgereaktion im Anschluss an einen vollständig reversibel verlaufenden VET nicht mit befriedigender Genauigkeit möglich ist.

Für höhere Werte von $k_{h,0}$ konnten im Rahmen der zulässigen Genauigkeit $\geq 0.1\%$ des Algorithmus keine zufriedenstellenden Approximationen erreicht werden. Es sei allerdings betont, dass für quasi-reversibel bis irreversibel verlaufende VETs ($0.1 \leq k_h \leq 10^{-4} \text{ cm} \cdot \text{s}^{-1}$) der Algorithmus verlässliche Resultate liefert.

Eine Verbesserung der Ergebnisse könnte möglicherweise durch die Wahl eines anderen numerischen Algorithmus erreicht werden. Im Falle stringenter Fehlertoleranzen wird in der Literatur z.B. die Richardson-Extrapolation als Alternative zu dem hier eingesetzten eingebetteten Ansatz vorgeschlagen[60](S.32).

Des Weiteren könnte ein Wechsel der Randwerte des Modells, zumindest für die Spezialfälle eines einfachen Elektronentransfers und/oder für vollständig reversible Elektronentransferkinetik (Nernst-Gleichung), die Berechnungen erleichtern. Letztere Modifikationen stehen jedoch im Gegensatz zur Allgemeinheit des in dieser Arbeit gewählten Modellierungskonzepts.

8. Zusammenfassung

Die Arbeit befasste sich mit der objekt-orientierten Strukturierung und der numerischen Simulation von Reaktions-Diffusions-Prozessen in elektrochemischen Zellen.

In Teil I wurde ein allgemeines Konzept zur computerunterstützten Modellierung und Simulation dynamischer Elektrodenprozesse vorgestellt und in Form eines Kinetik-Compilers realisiert.

In Teil II wurde ein voll-adaptives Finite-Elemente-Verfahren an die, von dem Compiler erzeugten Modellgleichungen gekoppelt. Die Gültigkeit der Modellgleichungen sowie die Zuverlässigkeit und allgemeine Anwendbarkeit des numerischen Algorithmus wurden in Testsimulationen überprüft.¹

8.1. Teil I — Modellierung

Im Modellierungsteil dieser Arbeit wurden in der Elektroanalytik relevante Elektrodenprozesse durch ein allgemeines Modell wiedergegeben. Die kinetischen Modellgleichungen wurden mit Hilfe eines objekt-orientierten Designs strukturiert und in der C++-Softwarebibliothek Ecco (ElectroChemical COmpiler) zur Modellierung elektrochemischer Reaktionsmechanismen abgebildet. Ecco erweitert und verbessert dabei frühere Ansätze von elektrochemischen Reaktions-Compilern.

Der Compiler erzeugt aus einem nahezu beliebigen elektrochemischen Reaktionsmechanismus die Reaktions- und Randterme in Form von *Objekten*, welche zur numerischen Simulation von reaktiven Transportprozessen in elektrochemischen Zellen entworfen wurden. Die Zellen können eine oder mehrere Arbeitselektroden enthalten.

Das kinetische Modell beinhaltet Geschwindigkeitsgesetze für heterogene Elektronentransfers, homogene Reaktionen, Adsorptionen, Oberflächenelektronentransfers sowie Oberflächenreaktionen.

Die Kinetik der Elektronentransfers und der Adsorptionsprozesse folgt dabei den üblichen physikalisch-chemischen Beziehungen. Neben Potenzgesetzen kann die Kinetik für homogene und Oberflächenreaktionen durch sogenannte *generische Gesetze* frei definiert werden.

Die Eingabesprache des Kinetik-Compiler unterstützt eine intuitive chemische Formulierung der Reaktionsmechanismen. Ihre Grammatik wurde *nicht* auf der Basis des zugrundeliegenden mathematischen Modells entworfen, sondern folgt der üblichen chemischen

¹Die Software-Komponenten beider Teile bilden die Grundlage eines Anwenderprogramms zur Simulation elektroanalytischer Experimente (ModSim[30]).

Notation. Komplexe Geschwindigkeitsgesetze können optional in Form von algebraischen Ausdrücken definiert werden.

Die Übersetzung des Mechanismus durch den Kinetik-Compiler ist vollständig von der anschließenden Simulation getrennt. Die Effizienz der numerischen Simulation wird somit nicht von der Geschwindigkeit der Übersetzung beeinflusst.

In einer Modellanalyse wurden die Geschwindigkeitsgesetze als die kleinsten *Bausteine* des Gesamtmodells identifiziert und später in Form des *Composite*-Strukturmusters in einer *Law*-Klassenhierarchie abgebildet. Diese *Composite*-Struktur ermöglicht es dem Kinetik-Compiler die *Law*-Bausteine auf elegante Weise zu den Reaktions- und Randtermen des Modells zusammenzusetzen. Darüber hinaus kann die *Law*-Klassenhierarchie durch neue kinetische Modelle mittels zusätzlicher Unterklassen leicht erweitert werden. Alle Reaktions- und Randterme erfüllen eine gemeinsame, abstrakte Klassenschnittstelle. Im Falle einer zukünftigen Erweiterung des kinetischen Modells bleibt ein an diese Schnittstelle gekoppeltes numerisches Verfahren somit unbeeinflusst. Dies zeigte sich bereits während der Weiterentwicklung des Compilers im Rahmen dieser Arbeit.

Während der gesamten Entwicklung des Compilers wurden keine einschränkenden Annahmen über das später zur Simulation eingesetzte numerische Verfahren getroffen. Ecco kann somit an ein beliebiges numerisches Verfahren gebunden werden.

Für das objekt-orientierte Design kamen weitere bewährte Strukturmuster zum Einsatz: Das *Fassade*-Muster entkoppelt alle Klassen des Übersetzungsalgorithmus vom Klienten und ermöglicht somit eine Veränderung des Compiler-Inneren *ohne* Auswirkungen nach Außen. Der Übersetzungsalgorithmus wurde zudem durch das *Kommando*-Muster strukturiert, sodass neue algorithmische Bestandteile auf elegante Weise hinzugefügt werden können. Die Kommunikation zwischen *Law*- und den dazugehörigen Reaktionsparameter-Objekten wird durch eine in dieser Arbeit entworfene generische Form des *Beobachter*-Musters gesteuert. Komplexe Beziehungen zwischen abhängigen Reaktionsparametern werden über ein Zusammenspiel von *Mediator*- und *Beobachter*-Objekten aufgelöst.

Neben diesen eher technischen Aspekten der Implementierung ermöglicht erst die Bereitstellung der Ecco-Bibliothek als *Open-Source-Software* eine realistische Wiederverwendung und Erweiterung der Software durch andere Wissenschaftler.

Es ist somit zum ersten Mal gelungen, ein allgemeines und vollständig *dynamisches* Modellierungskonzept elektrochemischer Elektrodenprozesse in wiederverwendbarer, erweiterbarer und freier Weise der elektrochemischen Gemeinschaft zugänglich zu machen. Darüber hinaus liefern insbesondere die Erweiterungen der Modelle in Bezug auf *dynamische* Adsorptionsprozesse und daran gekoppelte Oberflächenreaktionen in dieser allgemeinen Form bisher noch nicht bekannte Möglichkeiten zur Simulation heterogener Prozesse an Elektrodenoberflächen.

In Verbindung mit einem leistungsstarken numerischen Verfahren können die Modelle von Ecco somit zur Aufklärung der Art, der Thermodynamik und insbesondere der Dynamik verschiedenster elektrochemischer Phänomene dienen.

8.2. Teil II — Simulation

Im Simulationsteil dieser Arbeit wurden voll-adaptive Algorithmen auf der Basis eines adaptiven Multilevel-Finite-Elemente-Verfahrens für eindimensionale, elektrochemische Reaktions-Diffusions-Modelle objekt-orientiert strukturiert und in Form einer C++-Klassenbibliothek implementiert.

Für den Stofftransport in der Zelle wurden *lineare* Diffusionsbedingungen vorausgesetzt. Die Algorithmen können somit zur Simulation *planarer* Elektrodengeometrien verwendet werden. Darüber hinaus sind semi-infinite und finite Diffusionsmodelle mit einer oder zwei sich gegenüberliegenden Arbeitselektroden zugänglich. An den Elektroden können potential- oder stromkontrollierte Experimente simuliert werden. Migrations- und Konvektionsprozesse werden nicht berücksichtigt.

Durch die Kopplung der Algorithmen an die Reaktions- und Randterme des Kinetik-Compilers aus Teil I sind sämtliche in der Ecco-Bibliothek enthaltene Elektrodenprozesse zugänglich. Die Anbindung der Verfahren an die *Excitation Function Template Library* ermöglicht zudem den Einsatz verschiedenster Anregungsfunktionen, sowohl für potentialkontrollierte als auch für stromkontrollierte Experimente.

Die hier verwendeten numerischen Algorithmen beschreiben Abkömmlinge der *Rothe*-Methode, in denen zunächst die Zeitdiskretisierung von *Rosenbrock*-Verfahren durchgeführt wird. Die Kontrolle der Zeitschrittweiten erfolgt über ein eingebettetes Verfahren. Die durch die Zeitdiskretisierung gebildeten stationären Probleme werden durch einen adaptiven Multilevel-Finite-Elemente-Ansatz behandelt. Ein *hierarchischer* Fehlerschätzer steuert die Adaption des FE-Gitters. Die Verfahren wurden von Lang[60] ausführlich analysiert und zur Simulation diverser nicht-elektrochemischer Modelle eingesetzt. Für elektrochemische Modelle konnten ihre Eigenschaften in der vorliegenden Arbeit zum ersten Mal untersucht werden.

Die Observable des elektrochemischen Experiments (Strom bzw. Potential) wurde in das numerische Modell als zusätzliche Lösungsvariable durch das Strom-Potential-Residuum eingebracht (*Variante 1*). Das Residuum beschreibt dabei die Differenz aus Strom und den kinetischen, potentialabhängigen Termen der Elektronentransferreaktionen an einer Elektrode. Für potentialkontrollierte Experimente besteht zudem die Alternative, den Strom über den klassischen Ansatz einer Approximation der Oberflächengradienten zu ermitteln (*Variante 2*).

Die Betrachtung der Observablen als Lösung einer zusätzlichen, algebraischen Randgleichung, führt im Allgemeinen zu einem System partieller Differentialgleichungen im Inneren der Zelle, gekoppelt mit differentiell-algebraischen Systemen an den Elektrodenoberflächen.

Im *potentialkontrollierten* Fall wird dadurch zum ersten Mal in der elektrochemischen Literatur die Berechnung des Stroms als Lösung eines allgemeinen, differentiell-algebraischen Randsystems mit Hilfe des Strom-Potential-Residuums durchgeführt. Somit konnte ein adaptives Verfahren geschaffen werden, welches auf natürliche Weise eine zusätzliche Fehlerkontrolle im Strom vornimmt. Dies ist mit der klassischen *Variante 2*

nicht möglich. Die Adaption der Zeitschrittweiten und des Ortsgitters auf Basis einer Schätzung des Fehlers in den Konzentrationen gelöster *und* adsorbierter Teilchen bleibt vollständig erhalten.

Im Rahmen dieser Arbeit wurden umfangreiche Testsimulationen anhand einiger Modellprobleme durchgeführt. Letztere wurden dabei nach zwei Prinzipien ausgewählt. Zum einen sollten sie die Vielfalt und Gültigkeit der Modelle demonstrieren, die von dem Kinetik-Compiler der Ecco-Bibliothek und der *Excitation Function Template Library* angeboten werden (siehe Teil I). Zum anderen wurden durch das Auftreten nicht-linearer, un stetiger und singulärer Rand- und Anfangswerte sowie durch steile, sich in der Diffusionsschicht bewegende Reaktionsfronten erhebliche Herausforderungen an das numerische Verfahren gestellt. Die Zuverlässigkeit und allgemeine Anwendbarkeit der Algorithmen konnte somit überprüft werden.

Zur Simulation kamen jeweils die Rosenbrock-Verfahren Ros3P, Rowda3, RodasP und Ros2 zum Einsatz. Die Simulationsergebnisse wurden, falls verfügbar, mit analytischen Lösungen verglichen.

Insgesamt zeigten die Simulationen, dass für die *potentialkontrollierten* Simulationen die Algorithmen in allen Fällen bei moderaten Fehlertoleranzen von 1 – 2% zuverlässige Resultate lieferten. Hier stellte sich insbesondere das Rosenbrock-Schema Ros3P als die bevorzugte Methode heraus. Die Berechnung des Stroms nach *Variante 1* kann somit als eine neue, verlässliche Alternative zu den bisherigen adaptiven Ansätzen in der Elektrochemie betrachtet werden.

Speziell konnte gezeigt werden, wie sich die CV-Signale eines Oberflächenelektronentransfers unter dem Einfluss zweier Oberflächenreaktionen mit unterschiedlicher Reaktionskinetik verändern. Weiterführende Simulationen könnten, in Verbindung mit experimentellen, elektroanalytischen Studien, in diesem Zusammenhang wichtige Aussagen über Reaktionsabläufe heterogener Katalysen liefern. Eine solche Möglichkeit besteht mit den bisher in der Literatur bekannten Simulationsansätzen nicht.

Im Gegensatz zum potentialkontrollierten Fall ist die Simulationen allgemeiner *stromkontrollierter* Modelle durch den, in Bezug auf das Elektrodenpotential nicht-linearen Charakter des Strom-Potential-Residuums erschwert. Zudem treten zu Beginn stromkontrollierter Experimente häufig Singularitäten auf, sodass zusätzliche Modifikationen nötig waren, um die Modelle mit den hier verwendeten numerischen Verfahren lösen zu können.

Insgesamt zeigte sich, dass die Rowda3-Methode für stromkontrollierte Modelle den anderen Methoden überlegen ist. Im Falle eines einfachen Elektronentransfers lieferte Rowda3 für alle getesteten Anregungsfunktionen zuverlässige Ergebnisse. Ein konvergentes Verhalten des Algorithmus mit abnehmender relativer Fehlertoleranz konnte erst durch die gezielte Wahl der absoluten Fehlertoleranzen beobachtet werden. Im Vergleich zum potentialkontrollierten Fall konnten zufriedenstellende Ergebnisse erst bei niedrigeren Fehlertoleranzen von 1 – 0.2% und deutlich höheren Rechenzeiten erzielt werden.

Alle stromkontrollierten Simulationen wurden mit analytischen Lösungen in den Grenz-

fällen eines irreversibel oder reversibel verlaufenen Elektronentransfers verglichen. Es stellte sich schließlich heraus, dass im Falle einer schnellen Folgereaktion erster Ordnung die analytische Lösung des reversiblen Grenzfalles im Rahmen der für eingebettete Verfahren typischen moderaten Genauigkeiten bis 0.1% nicht mehr angenähert werden konnte. Es sei jedoch betont, dass für quasi-reversible bis irreversible Elektronentransferkinetik das Rowda3-Verfahren bisher stets verlässliche Ergebnisse lieferte.

Für die Implementierung der Algorithmen kam das *Strategie*-Muster in Verbindung mit C++-Klassentemplates zum Einsatz. Somit ist eine Erweiterung durch andere Rosenbrock-Verfahren ohne größeren Aufwand möglich.

A. Anhang

A.1. Erläuterung zu Gleichung (3.19)

Der vollständige Beweis der Beziehungen (3.18) und (3.19)

$$K^s = \frac{k_f}{k_b}, \quad s = \frac{\beta_i - \alpha_i}{\nu_i} > 0, \quad \forall i \quad (\text{A.1})$$

für den Gleichgewichtszustand einer Reaktion

$$0 \rightleftharpoons \sum_i \nu_i A_i \quad (\text{A.2})$$

wird von Blum und Luus[77] beschrieben und soll an dieser Stelle nur ansatzweise dargestellt werden.

Im Gleichgewichtszustand gilt für die Gleichgewichtskonstante das Massenwirkungsgesetz.

$$K = \prod_i c_i^{\nu_i} \quad (\text{A.3})$$

Für den Quotienten der Geschwindigkeitskonstanten folgt die Beziehung

$$\frac{k_f}{k_b} = \prod_i c_i^{\beta_i - \alpha_i} \quad (\text{A.4})$$

Weiterhin wird angenommen, dass K und k_f/k_b alleine von der Temperatur abhängen, sodass eine funktionale Beziehung $k_f/k_b = f(K)$ besteht. Mit (A.3) und (A.4) gilt

$$f\left(\prod_i c_i^{\nu_i}\right) = \prod_i c_i^{\beta_i - \alpha_i} \quad (\text{A.5})$$

Wir wählen nun einen beliebigen Index j und setzen in (A.5) $c_i = 1$ für alle $i \neq j$. Man erhält

$$f(c_j^{\nu_j}) = c_j^{\beta_j - \alpha_j} \quad (\text{A.6})$$

Mit $c_j = z^{1/\nu_j}$ und der Definition $s := (\beta_j - \alpha_j)/\nu_j$ folgt daraus

$$f(z) = z^{(\beta_j - \alpha_j)/\nu_j} = z^s \quad (\text{A.7})$$

Da j beliebig gewählt wurde, gilt $s = (\beta_j - \alpha_j)/\nu_j$ für alle j . Gleichung (A.7) liefert somit die Bedingung $f(K) = K^s$. Die Bedingung $s > 0$ folgt aus einem eher technischen Beweis und basiert auf der Betrachtung des Einflusses der Reaktionslaufzahl auf die rechte Seite von Gleichung (A.4) (Details siehe[77]).

A.2. Ein generisches Beobachter-Muster

In objekt-orientierten Softwareentwürfen hat sich in der Vergangenheit das *Beobachter*-Muster zur Kontrolle von komplexen Aktualisierungsprozessen bewährt[27]. Im Rahmen dieser Arbeit wird es in unterschiedlichen Zusammenhängen eingesetzt. So dient das Muster innerhalb der Ecco- und Adapter-Bibliotheken zur Aktualisierung von Parameterwerten (siehe Abbildungen 4.10, 4.13 und 5.1). Zudem wird es zur graphischen Darstellung der Simulationsergebnisse sowie zur Kontrolle der Kommunikation zwischen sämtlichen Dialogen der graphischen Anwenderoberfläche (GUI) des ModSim-Programms verwendet.

Der Aktualisierungsmechanismus beruht dabei auf der Verwendung von sogenannten Aspekt-Klassen[27], welche kontextabhängige Informationen transportieren und deren Schnittstelle sich mit den Anwendungsbereichen verändert. Unter Verwendung der in[27] beschriebenen, nicht-generischen Implementierung des Musters, wäre eine Vervielfältigung des Quellcodes nicht zu vermeiden. In jedem Anwendungsbereich (z.B. in jeder der obigen Bibliotheken) müssten aufgrund einer spezifischen Aspekt-Schnittstelle die Subjekt- und Observer-Klassen erneut implementiert werden. Eine Lösung bietet hier die Verwendung eines generischen Designs.

Abbildung A.1 zeigt die Struktur des generischen Beobachter-Musters, in dem der Template-Parameter A für eine nahezu beliebige Aspekt-Schnittstelle steht. Letztere muss als einzige Vorgabe eine Funktion 'what()' besitzen, die das Aktualisierungsereignis als string zurückgibt (siehe S.165). Die Verwendung der Klassen erfolgt wie in[27], wobei die Aspektschnittstelle als Template-Parameter angegeben wird (siehe z.B. die Beschreibung zu Abbildung 4.13).

Die Aktualisierungsereignisse und die dazugehörigen Beobachter eines Subjekts werden mithilfe einer STL-map[131] verwaltet. Die map erhält in der 'logIn()' -Methode das Ereignis (Schlüssel) und trägt den Beobachter in die dazugehörige Liste (Wert) ein. Die 'inform()' -Methode greift dann über den durch den Aspekt definierten Ereignisschlüssel ('what() -Methode') auf die Beobachterliste zu und ruft die 'update()' -Methode auf allen darin enthaltenen Beobachter-Objekten auf (siehe Pseudo-Code in Abbildung A.1). Die Unterklassen der Observer-Schnittstelle können somit die transportierten Informationen des Aspektobjekts abfragen.

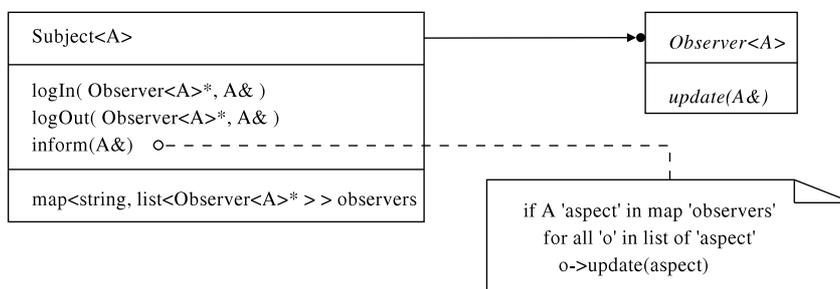


Abbildung A.1: Klassendiagramm des generischen Beobachter-Musters.

Die vollständige Implementierung des Musters wird im Folgenden angegeben. Es sei erwähnt, dass durch die Angabe eines Defaultwertes für die Aspekt-Policy das Muster auch ohne die Verwendung von Aspekten eingesetzt werden kann.

Die Implementierung der Observer-Klasse erfolgt in der Datei 'observer.hpp':

```
//! (Default) policy for observer aspects
class Aspect
{
public:
    Aspect(){}
    ~Aspect(){}

    std::string what() const
    {
        return "Aspect";
    }
};

//! generic observer class
template<class A = Aspect>
class Observer
{
public:
    Observer(){}
    virtual ~Observer(){}

    virtual void update( const A& aspect = A() ) = 0;
};
```

Die Subjekt-Klasse ist in der Datei 'subject.hpp' implementiert:

```
template<class A = Aspect>
class Subject
{
public:
    virtual ~Subject(){}

    virtual void logIn( Observer<A>* observer, const A& aspect = A() );
    virtual void logOut( Observer<A>* observer, const A& aspect = A() );
    virtual void inform( const A& aspect = A() );
};
```

```
protected:
    Subject(){}

private:
    std::map<std::string, std::list<Observer<A>* > > _observers;
    typedef typename std::list<Observer<A>* >::iterator LI;

};

template<class A>
void
Subject<A>::logIn( Observer<A>* observer, const A& aspect )
{
    std::string what = aspect.what();
    if ( _observers.count(what) == 0 )
    {
        _observers.insert( std::make_pair( what, std::list<Observer<A>* >() ) );
        _observers.find(what)->second.push_back(observer);
    }
    else
    {
        _observers.find(what)->second.push_back(observer);
    }
}

template<class A>
void
Subject<A>::logOut( Observer<A>* observer, const A& aspect )
{
    std::string what = aspect.what();
    if ( _observers.count(what) == 0 )
    {
        // do nothing!
    }
    else
    {
        _observers.find(what)->second.remove(observer);
    }
}

template<class A>
void
Subject<A>::inform( const A& aspect )
```

```
{
  std::string what = aspect.what();
  if ( _observers.count(what) == 0 )
  {
    // do nothing!
  }
  else
  {
    std::list<Observer<A>* > observers = _observers.find(what)->second;
    LI p = observers.begin();
    for(; p != observers.end(); ++p)
    {
      (*p)->update( aspect );
    }
  }
}
```

A.3. Die *definition*-Klasse

Die folgenden Zeilen stellen einen Auszug aus der geschachtelten *definition*-Klasse aus Abbildung 4.6 dar. Sie entsprechen gleichzeitig der nahezu vollständigen Grammatikdefinition der Eingabesprache von Ecco (vgl. Abbildung 4.4).

```
//-----
// Grammar definition
//-----

template<typename ScannerT>
Parser::definition<ScannerT>::definition(Parser const& self)
{

  //-----
  // Rules
  //-----

  electron
    = leaf_node_d[ str_p("e-") ];

  rev
    = leaf_node_d[ str_p("<=>") ];

  irrev
    = leaf_node_d[ str_p("-->") ];
```

```
equil
  = leaf_node_d[ str_p("=") ];

langmuir
  = leaf_node_d[ str_p("Langmuir") ];

henry
  = leaf_node_d[ str_p("Henry") ];

temkin
  = leaf_node_d[ str_p("Temkin") ];

frumkin
  = leaf_node_d[ str_p("Frumkin") ];

bv
  = leaf_node_d[ str_p("Butler-Volmer") ];

marcus
  = leaf_node_d[ str_p("Marcus") ];

nernst
  = leaf_node_d[ str_p("Nernst") ];

number
  = leaf_node_d[ real_p ];

integer
  = leaf_node_d[ lexeme_d[ !ch_p('-') >> +digit_p ] ];

coefficient
  = leaf_node_d
  [
    lexeme_d[ +digit_p >> blank_p ]
    - ( lexeme_d[ +digit_p ] >> ( ch_p('+') | '-' | '<' | ';' ) )
  ];

ion
  = leaf_node_d
  [
    lexeme_d
    [
      +( alnum_p | '(' | '[' )
```

```

    >> *( alnum_p | ']' | ') ' | '_ ' | '- ' )
    >> ch_p('^') >> ch_p('(')
        >> +digit_p >> ( ch_p('+') | '- ' )
        >> ch_p(')')
    ]
];

neutrum
= leaf_node_d
[
  lexeme_d
  [
    +( alnum_p | '(' | '[' ) >>
    *( alnum_p | ']' | ') ' | '_ ' | '- ' )
  ]
];

adsIon
= leaf_node_d
[
  lexeme_d
  [
    +( alnum_p | '(' | '[' ) >>
    *( alnum_p | ']' | ') ' | '_ ' | '- ' )
    >> ch_p('^') >> ch_p('(')
        >> +digit_p >> ( ch_p('+') | '- ' )
        >> ch_p(')') >> str_p("{ads}")
  ]
];

adsNeutrum
= leaf_node_d
[
  lexeme_d
  [
    +( alnum_p | '(' | '[' ) >>
    *( alnum_p | ']' | ') ' | '_ ' | '- ' ) >> str_p("{ads}")
  ]
];

options
= infix_node_d[ boundaryOperator >> !( ch_p(',') >> laws ) >> ch_p(';') ]
| infix_node_d[ laws >> !( ch_p(',') >> boundaryOperator ) >> ch_p(';') ];

```

```
laws
  = adsorptionLaw
  | electrodeLaw
  | homogeneousLaw;

adsorptionLaw
  = langmuir
  | henry
  | temkin
  | frumkin;

electrodeLaw
  = bv
  | marcus
  | nernst;

boundaryOperator
  = inner_node_d[ str_p("b<") >> integer >> ch_p('>') ];

homogeneousLaw
  = powerLaw
  | genericRate;

powerLaw
  = infix_node_d[ powerExpression >> *( ch_p(',') >> powerExpression ) ];

powerExpression
  = infix_node_d[ powerOperator >> ch_p('=') >> powers ];

powers
  = number
  | numberTuple;

numberTuple
  = discard_node_d[ch_p('(')]
  >> infix_node_d[ number >> ch_p(',') >> number >> ch_p(')') ];

powerOperator
  = inner_node_d[ str_p("p<") >> symbol >> ch_p('>') ];

symbol
  = leaf_node_d
  [
    lexeme_d
```

```

[
  +(alnum_p | '[' | ']' | '('
    | ')' | '{' | '}'
    | '^' | '+' | '-'
    | '_' | ',' )
]
];

//-----
// grammar for generic rate laws
//-----

genericRate
  = discard_node_d[ ch_p('r') >> ch_p('=') ] >> expression;

//-----
// grammar for mathematical expressions
//-----

term
  = factor
    >> *( (ch_p('*') >> factor)
          | (ch_p('/') >> factor)
          );

expression
  = term
    >> *( (ch_p('+') >> term)
          | (ch_p('-') >> term)
          );

factor
  = number
    | concentrationOperator
    | constant
    | forwardRateConstant
    | backwardRateConstant
    | equilibriumConstant
    | function
    | ch_p('(') >> expression >> discard_node_d[ch_p(')')]
    | (ch_p('-') >> factor);

function

```

```
= functionTable1 >> inner_node_d[ '(' >> expression >> ')' ]
| infix_node_d[ functionTable2 >> '(' >> expression
                >> ',' >> expression >> ')' ]];

concentrationOperator
= inner_node_d[ str_p("c<") >> symbol >> ch_p('>') ];

constant
= leaf_node_d[ lexeme_d[ (alpha_p - keywords) >> *digit_p ] ];

forwardRateConstant
= leaf_node_d[ lexeme_d[ str_p("kf") >> +digit_p ] ];

backwardRateConstant
= leaf_node_d[ lexeme_d[ str_p("kb") >> +digit_p ] ];

equilibriumConstant
= leaf_node_d[ lexeme_d[ ch_p('K') >> +digit_p ] ];

//-----
// grammar for overall reaction network
//-----

reactionNetwork
= reactionBlock >> *( reactionBlock );

reactionBlock
= discard_last_node_d
  [
    reaction
    >> echem_guard( expect_semi( ch_p(';') )) [ErrorHandler() ]
  ]
>> !(options);

reaction
= educts >> reactionArrows >> products;

reactionArrows
= rev
| irrev
| equil;

educts
= infix_node_d[ reactionAddend >> *( ch_p('+') >> reactionAddend ) ];
```

```
products
  = infix_node_d[ reactionAddend >> *( ch_p('++') >> reactionAddend ) ];

reactionAddend
  = !(coefficient) >> species;

species
  = electron
  | adsIon
  | adsNeutrum
  | ion
  | neutrum;

}
```

A.4. Technische Anmerkungen

Die Entwicklung der Software-Bausteine wurde unter den SuSE-Linux 8.1 – 9.3 Betriebssystemen (SuSE Linux, Nürnberg, Deutschland[222]) mit dem g++ Compiler (Versionen 3.2 – 3.3) der GNU Compiler Collection[223] durchgeführt.

Die Rechenzeiten in Kapitel 7 wurden mit der STL-clock()-Funktion[131] unter SuSE Linux 9.3 (Pentium M, 1.3 GHz) gemessen.

Die im Rahmen dieser Arbeit entworfene Software ist lizenziert unter der GPL (GNU General Public Licence[224]) und kann frei bezogen werden[30].

Die Software basiert auf den Open-Source-Bibliotheken: GiNaC[133], GMM++[197], Qt[225], Quantity[137], Spirit[124], STL[131], tinyxml[226], tree.hh[202] und VTK[227].

Literaturverzeichnis

- [1] D. Britz, *Digital Simulation in Electrochemistry*, Band 666, *Lecture Notes in Physics*, Springer, Berlin, **2005**.
- [2] B. Speiser, in A.J. Bard, I. Rubinstein (Hrsg.), *Electroanal. Chem.*, Band 19, S. 1 – 108, Marcel Dekker, New York, **1996**.
- [3] S.W. Feldberg, in J. Leddy, V. Birss, P. Vanýsek (Hrsg.), *Historical Perspectives on the Evolution of Electrochemical Tools*, Band 2002-29, S. 189 – 214, Electrochem. Soc., New Jersey, **2004**.
- [4] L.K. Bieniasz, D. Britz, *Polish J. Chem.* **2004**, *78*, 1195 – 1219.
- [5] M.T.M Koper, A.P.J. Jansen, R.A. van Santen, J.J. Lukkien, P.A.J. Hilbers, *J. Chem. Phys.* **1998**, *109*(14), 6051 – 6062.
- [6] W.B. Gu, C.Y. Wang, B.Y. Liaw, *J. Electrochem. Soc.* **1998**, *145*(10), 3418 – 3427.
- [7] A.F. Slaterbeck, M.L. Stegemiller, C.J. Seliskar, T.H. Ridgway, W.R. Heineman, *Anal. Chem.* **2000**, *72*, 5567 – 5575.
- [8] I.-M. Hsing, X. Wang, Y.-J. Leng, *J. Electrochem. Soc.* **2002**, *149*(5), A615 – A621.
- [9] C.-Y. Wang, *Chem. Rev.* **2004**, *104*, 4727 – 4766.
- [10] T. Iwasita, *Handbook of Fuel Cells – Fundamentals, Technology and Applications*, Band 2, *Electrocatalysis*, Kapitel 41, John Wiley & Sons, **2003**.
- [11] J. Strutwolf, D.E. Williams, *Electroanalysis* **2005**, *17*(2), 169 – 177.
- [12] T. Ito, K. Maruyama, O. Niwa, K. Suzuki, in C. Bruckner-Lea, P. Vanýsek, G. Hunter, M. Egashira, N. Miura, F. Mizutani (Hrsg.), *Chemical Sensors VI: Chemical and Biological Sensors and Analytical Methods*, S. 289 – 299, Electrochem. Soc., New Jersey, **2004**.
- [13] T. Ito, K. Maruyama, K. Sobue, S. Ohya, O. Niwa, K. Suzuki, *Electroanalysis* **2004**, *16*(24), 2035 – 2041.
- [14] F. Novak, B. Speiser, E. Lindner, Z.-L. Lu, H.A. Mayer, *Angew. Chem.* **2004**, *116*, 2059 – 2062; *Angew. Chem. Int. Ed.* **2004**, *43*, 2025 – 2028.

- [15] F. Novak, B. Speiser, H.A.Y. Mohammad, H.A. Mayer, *Electrochim. Acta* **2004**, *49*, 3841 – 3853.
- [16] B. Speiser, in A.J. Bard, M. Stratmann, H. Schäfer (Hrsg.), *Encyclopedia of Electrochemistry, Vol. 8 Organic Electrochemistry*, Kapitel 1, S. 1 – 23, Wiley-VCH, **2004**.
- [17] B. Speiser, *Curr. Org. Chem.* **1999**, *3*, 171 – 191.
- [18] M.V. Mirkin, B.R. Horrocks, *Anal. Chim. Acta* **2000**, *406*, 119 – 146.
- [19] G. Nagy, L. Nagy, *Fresenius J. Anal. Chem.* **2000**, *366*, 735 – 744.
- [20] B. Limoges, J.-M. Savéant, D. Yazidi, *J. Am. Chem. Soc.* **2003**, *125*(30), 9192 – 9203.
- [21] C.P. Andrieux, B. Limoges, D. Marchal, J.-M. Savéant, *Anal. Chem.* **2006**, *78*, 3138 – 3143.
- [22] <http://www.lemp7.cnrs.fr>, Juni 2006.
- [23] B. Speiser, *Comput. Chem.* **1990**, *14*, 127 – 140.
- [24] A.W. Bott, S.W. Feldberg, M. Rudolph, *Curr. Sep.* **1995**, *13*, 108 – 112.
- [25] <http://www.digielch.de>, Juni 2006.
- [26] L.K. Bieniasz, *Comp. Chem.* **1997**, *21*, 1 – 12.
- [27] E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design Patterns. Elements of Reusable Object-Oriented Software*, Addison-Wesley, Boston, **1995**.
- [28] S.L. Graham, *Informatics. 10 years back. 10 years ahead*, S. 195 – 208, Springer, **2001**.
- [29] L.K. Bieniasz, in B.E. Conway, R.E. White (Hrsg.), *Mod. Asp. Electrochem.*, Band 35, S. 135 – 195, Kluwer Academic/Plenum Publishers, New York, **2002**.
- [30] <http://echempp.sourceforge.net>, Juli 2006.
- [31] <http://sourceforge.net/projects/echempp>, Juli 2006.
- [32] B. Oestereich, *Objektorientierte Softwareentwicklung*, Oldenbourg, München, 5. Auflage, **2001**.
- [33] D. Britz, *Digital Simulation in Electrochemistry*, Springer, Berlin, 2. Auflage, **1988**.

- [34] M. Rudolph, in I. Rubinstein (Hrsg.), *Physical Electrochemistry. Principles, Methods, and Applications*, Monographs in Electroanalytical Chemistry and Electrochemistry, S. 81 – 129, Marcel Dekker, New York, **1995**.
- [35] P.N. Bartlett, S.L. Taylor, *J. Electroanal. Chem.* **1998**, *453*, 49 – 60.
- [36] K. Harriman, D.J. Gavaghan, P. Houston, E. Süli, *Electrochem. Commun.* **2000**, *2*, 157 – 162.
- [37] K. Harriman, D.J. Gavaghan, E. Süli, *Electrochem. Comm.* **2003**, *5*, 519 – 529.
- [38] S.C.B. Abercrombie, G. Denuault, *Electrochem. Comm.* **2003**, *5*, 647 – 656.
- [39] C. Amatore, I. Svir, *J. Electroanal. Chem.* **2003**, *557*, 75 – 90.
- [40] O. Sklyar, G. Wittstock, *J. Phys. Chem. B* **2002**, *106*, 7499 – 7508.
- [41] T. Nann, J. Heinze, *Electrochim. Acta* **2003**, *48*, 3975 – 3980.
- [42] J.A. Alden, R.G. Compton, *J. Phys. Chem. B* **1997**, *101*, 8941 – 8954.
- [43] N.P.C. Stevens, A.C. Fisher, *J. Phys. Chem. B* **1997**, *101*, 8259 – 8263.
- [44] K.A. Gooch, A.C. Fisher, *J. Phys. Chem. B* **2002**, *106*, 10668 – 10673.
- [45] I.E. Henley, A.C. Fisher, *J. Phys. Chem. B* **2003**, *107*, 6579 – 6585.
- [46] J. Zhang, N. Li, *Electroanalysis* **2004**, *16*, 650 – 655.
- [47] M. Thompson, O.V. Klymenko, R.G. Compton, *J. Electroanal. Chem.* **2005**, *576*, 333 – 338.
- [48] I. Henley, A. Fisher, *Electroanalysis* **2005**, *17*, 255 – 262.
- [49] O.V. Klymenko, D.J. Gavaghan, K.E. Harriman, R.G. Compton, *J. Electroanal. Chem.* **2002**, *531*, 25 – 31.
- [50] P. Salaun, J. Josserand, J. Morandini, H.H. Girault, J. Buffle, *J. Electroanal. Chem.* **2004**, *566*, 147 – 158.
- [51] A.J. Bard, L.R. Faulkner, *Electrochemical Methods. Fundamentals and Applications*, Wiley, New York, 2. Auflage, **2001**.
- [52] K. Ludwig, L. Rajendran, B. Speiser, *J. Electroanal. Chem.* **2004**, *568*, 203 – 214. Erratum:[228].
- [53] J.M. Savéant, *Electrochim. Acta* **1967**, *12*, 999 – 1030.

- [54] S. Buchmann, H.A. Mayer, B. Speiser, M. Seiler, M.P. Feth, H. Bertagnolli, S. Steinbrecher, E. Plies, *Electrochim. Acta* **2003**, *48*, 2725 – 2737. Erratum:[229].
- [55] A. Molina, J. Gonzalez, *Langmuir* **2003**, *19*, 406 – 415.
- [56] J. Gonzalez, A. Molina, *Langmuir* **2001**, *17*, 5520 – 5526.
- [57] W.R. Heineman, P.T. Kissinger, in P.T. Kissinger, W.R. Heineman (Hrsg.), *Laboratory Techniques in Electroanalytical Chemistry*, S. 51 – 125, Marcel Dekker, New York, 2. Auflage, **1996**.
- [58] B. Speiser, in A.J. Bard, M. Stratmann, P. Unwin (Hrsg.), *Encyclopedia of Electrochemistry, Vol. 3 Instrumentation and Electroanalytical Chemistry*, Kapitel 2.1, S. 81 – 104, Wiley-VCH, **2003**.
- [59] A. Molina, M. López-Tenés, C. Serna, *J. Electroanal. Chem.* **1993**, *346*, 53 – 71.
- [60] J. Lang (hrsg. von M. Griebel, D.E. Keyes, R.M. Nieminen, D. Roose, T. Schlick), *Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems. Theory, Algorithm and Applications*, Band 16, *Lecture Notes in Computational Science and Engineering*, Springer, Berlin, **2000**.
- [61] A. Hastings, K. Cuddington, K.F. Davies, C.J. Dugaw, S. Elmendorf, A. Freestone, S. Harrison, M. Holland, J. Lambrinos, U. Malvadkar, B.A. Melbourne, K. Moore, C. Taylor, D. Thomson, *Ecology Letters* **2005**, *8*, 91 – 101.
- [62] A.B. Medvinsky, S.V. Petrovskii, I.A. Tikhonova, H. Malchow, B.-L. Li, *SIAM Review* **2002**, *44*(3), 311 – 370.
- [63] K.H. Huebner, D.L. Dewhirst, D.E. Smith, T.G. Byrom, *The Finite Element Method for Engineers*, Wiley, New York, 4. Auflage, **2001**.
- [64] P. Knabner, L. Angermann, *Numerik partieller Differentialgleichungen: eine anwendungsorientierte Einführung*, Springer, **2000**.
- [65] F. Kleinschmidt, *Anomalous Diffusion in Anisotropic Media*, Dissertation, Albert-Ludwigs-Universität, Freiburg, **2005**.
- [66] J. Ross, M.O. Vlad, *Annu. Rev. Phys. Chem.* **1999**, *50*, 51 – 78.
- [67] C. Wagner, *J. Phys. Chem. B* **2004**, *108*, 2425 – 2431.
- [68] I. Fishtik, C.A. Callaghan, R. Datta, *J. Phys. Chem. B* **2004**, *108*, 5671 – 5682.
- [69] I. Fishtik, C.A. Callaghan, R. Datta, *J. Phys. Chem. B* **2004**, *108*, 5683 – 5697.

-
- [70] P. Kirkegaard, E. Bjergbakke, Technical Report Riso-R-1085(EN), Riso National Laboratory, Roskilde, Denmark, **2000**.
- [71] I.B. Svir, O.V. Klymenko, M.S. Platz, *Comp. Chem.* **2002**, *26*, 379 – 386.
- [72] L.K. Bieniasz, *J. Electroanal. Chem.* **1996**, *406*, 33 – 43.
- [73] L.K. Bieniasz, *J. Electroanal. Chem.* **1996**, *406*, 45 – 52.
- [74] W. Luo, S.W. Feldberg, M. Rudolph, *J. Electroanal. Chem.* **1994**, *368*, 109 – 113.
- [75] R.W. Missen, C.A. Mims, B.A. Saville, *Introduction to Chemical Reaction Engineering and Kinetics*, Willey, New York, **1999**.
- [76] L.P. Hammett, *Physikalische Organische Chemie*, Verlag Chemie, Weinheim/Bergstr., **1970**.
- [77] E.H. Blum, R. Luus, *Chem. Eng. Sci.* **1964**, *19*, 322 – 323.
- [78] E. Laviron, in A.J. Bard (Hrsg.), *Electroanalytical Chemistry*, Band 12, S. 53 – 157, Marcel Dekker, New York, **1982**.
- [79] IUPAC, *Pure Appl. Chem.* **1976**, *45*, 131 – 134.
- [80] R.A. Marcus, *Angew. Chem.* **1993**, *105*, 1161 – 1172; *Angew. Chem. Int. Ed. Engl.* **1993**, *32*, 1111 – 1121.
- [81] R.A. Marcus, *Pure Appl. Chem.* **1997**, *69*, 13 – 29.
- [82] R.A. Marcus, *J. Electroanal. Chem.* **1997**, *438*, 251 – 259.
- [83] R.A. Marcus, *J. Chem. Phys.* **1956**, *24*(5), 966 – 978.
- [84] R.A. Marcus, *J. Chem. Phys.* **1957**, *26*(4), 867 – 871.
- [85] R.A. Marcus, *J. Chem. Phys.* **1957**, *26*(4), 872 – 877.
- [86] R.A. Marcus, *Can. J. Chem.* **1959**, *37*, 155 – 163.
- [87] R.A. Marcus, *J. Phys. Chem.* **1963**, *67*, 853 – 857.
- [88] R.A. Marcus, *J. Chem. Phys.* **1965**, *43*(2), 679 – 701.
- [89] R.A. Marcus, *J. Electroanal. Chem.* **2000**, *483*, 2 – 6.
- [90] K. Volke, *Wissensch. Fortschr.* **1988**, *38*, 327 – 330.
- [91] K. Volke, *Tenside Surf. Det.* **1991**, *28*, 52 – 56.
-

- [92] J.M.L. Fonseca, M.E.S. de Vincente, *Electroanalysis* **2003**, *15*(21), 1682 – 1686.
- [93] R.H. Wopschall, I. Shain, *Anal. Chem.* **1967**, *39*(13), 1514 – 1527.
- [94] R.H. Wopschall, I. Shain, *Anal. Chem.* **1967**, *39*(13), 1535 – 1542.
- [95] E. Laviron, *J. Electroanal. Chem.* **1982**, *140*, 247 – 268.
- [96] A. Leverenz, B. Speiser, *J. Electroanal. Chem.* **1991**, *318*, 69 – 89.
- [97] S. Martinet, J. Bouteillon, J.P. Caire, *J. Appl. Electrochem.* **1998**, *28*, 819 – 825.
- [98] S.W. Feldberg, *Comput. Chem. Instrum.* **1972**, *2*, 185 – 215.
- [99] E. Laviron, L. Roullier, *J. Electroanal. Chem.* **1980**, *115*, 65 – 74.
- [100] C. Schulz, B. Speiser, *J. Electroanal. Chem.* **1993**, *354*, 255 – 271.
- [101] M. Lovrić, *Anal. Bioanal. Chem.* **2002**, *373*, 781 – 786.
- [102] D. Krulic, N. Fatouros, *J. Electroanal. Chem.* **1995**, *395*, 273 – 278.
- [103] V. Mirčeski, *J. Electroanal. Chem.* **2001**, *508*, 138 – 149.
- [104] V. Mirčeski, M. Lovrić, *J. Electroanal. Chem.* **2004**, *565*, 191 – 202.
- [105] J. Cecilia, J. Galceran, J. Salvador, J. Puy, F. Mas, *Int. J. Quant. Chem.* **1994**, *51*, 357 – 367.
- [106] E. Laviron, *J. Electroanal. Chem.* **1995**, *391*, 187 – 197.
- [107] P. Delahay, D.M. Mohilner, *J. Am. Chem. Soc.* **1962**, *84*, 4247 – 4252.
- [108] P. Nikitas, A. Papoutsis, *Electrochim. Acta* **1988**, *33*, 683 – 692.
- [109] M.H. Hulbert, I. Shain, *Anal. Chem.* **1970**, *42*, 162 – 171.
- [110] E.E. Engelman, D.H. Evans, *J. Electroanal. Chem.* **1992**, *331*, 739 – 749.
- [111] Y.-C. Liao, E.I. Franses, O.A. Basaran, *J. Coll. Inter. Sci.* **2003**, *258*, 310 – 321.
- [112] F.G. Chevallier, O.V. Klymenko, L. Jiang, T.G.J. Jones, R.G. Compton, *J. Electroanal. Chem.* **2005**, *574*, 217 – 237.
- [113] R. Gulaboski, V. Mirčeski, M. Lovrić, I. Bogeski, *Electrochem. Commun.* **2005**, *7*, 515 – 522.

- [114] H. Anders, *Objektorientierte Formulierung von Anfangs- und Randwertbedingungen bei elektrochemischen Simulationen*, Wissenschaftliche Arbeit, Eberhard-Karls-Universität Tübingen, **2005**.
- [115] B. Speiser, *Zur Bestimmung von Reaktionsparametern aus elektroanalytischen Experimenten – Entwicklung und Anwendung von Simulationstechniken und chemometrischen Methoden*, Habilitationsschrift der Fakultät für Chemie und Pharmazie der Eberhards-Karls-Universität Tübingen, **1989**.
- [116] L.K. Bieniasz, *J. Electroanal. Chem.* **1993**, *360*, 119 – 138.
- [117] N.P.C. Stevens, S.J. Hickey, A.C. Fisher, *An. Chim.* **1997**, *93*(4), 225 – 232.
- [118] T. Nann, J. Heinze, *Electrochem. Comm.* **1999**, *1*, 289 – 294.
- [119] K. Ludwig, B. Speiser, *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 2051 – 2060.
- [120] L.K. Bieniasz, *Comp. Chem.* **1996**, *20*, 403 – 418.
- [121] <http://www.drhuang.com>, Juli 2006.
- [122] K.G. Denbigh, *The Principles of Chemical Equilibrium*, Cambridge University Press, New York, **1981**.
- [123] A.V. Aho, R. Sethi, J.D. Ullman, *Compilers. Principles, Techniques, and Tools*, Addison-Wesley, Reading, **1988**.
- [124] <http://spirit.sourceforge.net>, Juli 2006.
- [125] <http://dinosaur.compilertools.net>, April 2004.
- [126] <http://www.gnu.org/software/flex>, April 2004.
- [127] <http://www.gnu.org/software/bison/bison.html>, April 2004.
- [128] <http://www.antlr.org>, April 2004.
- [129] ISO/IEC 14977:1996(E), <http://www.iso.ch>.
- [130] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, W. Lorensen, *Object-Oriented Modeling and Design*, Prentice Hall, Engelwood Cliffs, NJ, **1991**.
- [131] B. Stroustrup, *The C++ Programming Language*, Addison-Wesley, Reading, **1997**.
- [132] C. Bauer, A. Frink, R. Kreckel, *J. Symbolic Comput.* **2002**, *33*, 1 – 12.
- [133] <http://www.ginac.de>, Juni 2006.

- [134] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Fannery, *Numerical Recipes in C++: The Art of Scientific Computing*, Cambridge University Press, 2. Auflage, **2002**.
- [135] F. Opekar, P. Beran, *J. Electroanal. Chem.* **1976**, *69*, 1 – 105.
- [136] Z.-X. Deng, X.-Q. Lin, Z.-H. Tong, *Acta Chim. Sinica* **2002**, *60*(8), 1415 – 1418.
- [137] <http://sourceforge.net/projects/quantity>, Juni 2006.
- [138] W. Heineken, *Adaptive Verfahren zur numerischen Berechnung von Reaktions-Diffusions-Systemen*, Dissertation, Otto-von-Guerike-Universität Magdeburg, **2005**.
- [139] P. Deuffhard, F. Bornemann (hrsg. von J.E. Marsden, L. Sirovich, M. Golubitsky, S.S. Antman), *Scientific Computing with Ordinary Differential Equations*, Band 42, *Texts in Applied Mathematics*, Springer, New York, **2002**.
- [140] L.K. Bieniasz, *J. Electroanal. Chem.* **1994**, *374*, 1 – 22.
- [141] L.K. Bieniasz, *J. Electroanal. Chem.* **1994**, *374*, 23 – 35.
- [142] L.K. Bieniasz, *J. Electroanal. Chem.* **1994**, *379*, 71 – 87.
- [143] L.K. Bieniasz, *J. Electroanal. Chem.* **2000**, *481*, 115 – 133.
- [144] L.K. Bieniasz, *J. Electroanal. Chem.* **2000**, *481*, 134 – 151.
- [145] L.K. Bieniasz, C. Bureau, *J. Electroanal. Chem.* **2000**, *481*, 152 – 167.
- [146] L.K. Bieniasz, *Electrochem. Commun.* **2001**, *3*, 149 – 153.
- [147] L.K. Bieniasz, *J. Electroanal. Chem.* **2002**, *527*, 1 – 10.
- [148] L.K. Bieniasz, *J. Electroanal. Chem.* **2002**, *527*, 11 – 20.
- [149] L.K. Bieniasz, *J. Electroanal. Chem.* **2002**, *527*, 21 – 32.
- [150] L.K. Bieniasz, *J. Electroanal. Chem.* **2002**, *529*, 51 – 58.
- [151] L.K. Bieniasz, *J. Electroanal. Chem.* **2004**, *565*, 251 – 271.
- [152] L.K. Bieniasz, *J. Electroanal. Chem.* **2004**, *565*, 273 – 285.
- [153] H.R. Schwarz, *Methode der Finiten Elemente*, Teubner, Stuttgart, **1991**.
- [154] D. Braess, *Finite Elemente - Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie*, Springer, Berlin, **1992**.

- [155] J.N. Reddy, D.K. Gartling, *The Finite Element Method in Heat Transfer and Fluid Dynamics*, CRC Press, Boca Raton, Florida, **1994**.
- [156] T.J.R. Hughes, *The Finite Element Method, Linear Static and Dynamic Finite Element Analysis*, Dover Publications, Mineola, New York, **2000**.
- [157] I. Babuska, W.C. Rheinboldt, *SIAM J. Numer. Anal.* **1978**, *15*(4), 736 – 754.
- [158] I. Babuska, A. Miller, *Int. J. Numer. Meth. Eng.* **1984**, *20*, 1085 – 1109.
- [159] I. Babuska, A. Miller, *Int. J. Numer. Meth. Eng.* **1984**, *20*, 1111 – 1129.
- [160] I. Babuska, A. Miller, *Int. J. Numer. Meth. Eng.* **1984**, *20*, 2311 – 2324.
- [161] R.E. Bank, A. Weiser, *Math. Comp.* **1985**, *44*(107), 283 – 301.
- [162] W. Gui, I. Babuska, *Numer. Math.* **1986**, *49*, 577 – 612.
- [163] W. Gui, I. Babuska, *Numer. Math.* **1986**, *49*, 613 – 657.
- [164] W. Gui, I. Babuska, *Numer. Math.* **1986**, *49*, 659 – 683.
- [165] F.A. Bornemann, *Imp. Comp. Sci. Eng.* **1990**, *2*, 279 – 317.
- [166] P. Leinen, *Ein schneller adaptiver Löser für Elliptische Randwertprobleme auf Seriell- und Parallelrechnern*, Dissertation, Universität Dortmund, **1990**.
- [167] K. Eriksson, C. Johnson, *SIAM J. Numer. Anal.* **1991**, *28*(1), 43 – 77.
- [168] J. Lang, A. Walter, *Imp. Comp. Sci. Eng.* **1992**, *4*, 269 – 314.
- [169] M. Ainsworth, J.T. Oden, *Numer. Math.* **1993**, *65*, 23 – 50.
- [170] R.E. Bank, R.K. Smith, *SIAM J. Numer. Anal.* **1993**, *30*(4), 921 – 935.
- [171] P. Leinen, *Computing* **1995**, *55*, 325 – 354.
- [172] K. Eriksson, C. Johnson, *SIAM J. Numer. Anal.* **1995**, *32*(6), 1729 – 1749.
- [173] R.E. Bank, *Acta Numer.* **1996**, 1 – 43.
- [174] R. Verfürth, *Numer. Math.* **1998**, *78*, 479 – 493.
- [175] P.K. Moore, *SIAM J. Sci. Comp.* **2000**, *21*(4), 1567 – 1586.
- [176] M. Ainsworth, J.T. Oden, *A Posteriori Error Estimation in Finite Element Analysis*, Pure and Applied Mathematics, Wiley, New York, **2000**.

- [177] K. Harriman, D.J. Gavaghan, P. Houston, E. Süli, *Electrochem. Commun.* **2000**, *2*, 150 – 156.
- [178] K. Harriman, D.J. Gavaghan, E. Süli, *J. Electroanal. Chem.* **2004**, *569*, 35 – 46.
- [179] K. Harriman, D.J. Gavaghan, E. Süli, *J. Electroanal. Chem.* **2004**, *573*, 169 – 174.
- [180] K. Gillow, D.J. Gavaghan, E. Süli, *J. Electroanal. Chem.* **2006**, *587*, 1 – 17.
- [181] K. Gillow, D.J. Gavaghan, E. Süli, *J. Electroanal. Chem.* **2006**, *587*, 18 – 24.
- [182] P. Deuffhard, P. Leinen, H. Yserentant, *IMPACT Comp. Sci. Eng.* **1989**, *1*, 3 – 35.
- [183] L.K. Bieniasz, D. Britz, *J. Electroanal. Chem.* **2001**, *503*, 141 – 152.
- [184] K. Ludwig, B. Speiser, *J. Electroanal. Chem.* **2006**, *588*, 74 – 87.
- [185] A. Sandu, J.G. Verwer, J.G. Blom, E.J. Spee, G.R. Carmichael, Technical Report NM-R9614, Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands, **1996**.
- [186] A. Sandu, J.G. Verwer, J.G. Blom, E.J. Spee, G.R. Carmichael, F.A. Potra, *Atmosph. Environm.* **1997**, *31*, 3459 – 3472.
- [187] M. Roche, *Numer. Math.* **1988**, *52*, 45 – 63.
- [188] C. Lubich, M. Roche, *Computing* **1990**, *43*, 325 – 342.
- [189] J. Lang, J.G. Verwer, Technical Report MAS-R0013, Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands, **2000**.
- [190] J. Lang, J. Verwer, *BIT* **2001**, *41*(4), 731 – 738.
- [191] K. Gustafsson, *ACM Trans. Math. Software* **1994**, *20*(4), 496 – 517.
- [192] J. Lang, *Appl. Numer. Math.* **1998**, *26*, 105 – 116.
- [193] H. Yserentant, *Numer. Math.* **1986**, *49*, 379 – 412.
- [194] J. Lang, B. Erdmann, *Numerical Heat Transfer, Part A* **2002**, *42*, 107 – 119.
- [195] J. Lang, W. Cao, W. Huang, R.D. Russell, *Appl. Numer. Math.* **2003**, *46*, 75 – 94.
- [196] I.N. Bronstein, K.A. Semendjajew, G. Musiol, H. Mühlig, *Taschenbuch der Mathematik*, Harri Deutsch, 5. Auflage, **2001**.
- [197] http://www-gmm.insa-toulouse.fr/getfem/gmm_intro, Juni 2006.

- [198] <http://crd.lbl.gov/~xiaoye/SuperLU>, Juni 2005.
- [199] G. Fischer, *Lineare Algebra*, Vieweg Verlag, 11. Auflage, **1997**.
- [200] K.E. Brenan, S.L. Campbell, L.R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, Elsevier, New York, **1989**.
- [201] J. Rang, L. Angermann, Mathematik-Bericht 2005/3, Institut für Mathematik, Technische Universität Clausthal, **2005**.
- [202] <http://www.aei.mpg.de/~peekas/tree>, Juni 2006.
- [203] F.G. Cottrell, *Z. Phys. Chem.* **1902**, *42*, 385.
- [204] L.K. Bieniasz, D. Britz, *Acta Chem. Scand.* **1993**, *47*, 757 – 767.
- [205] B. Speiser, *Acta Chem. Scand.* **1993**, *47*, 1238 – 1240.
- [206] L.K. Bieniasz, D. Britz, *Acta Chem. Scand.* **1994**, *48*, 609 – 610.
- [207] R.S. Nicholson, I. Shain, *Anal. Chem.* **1964**, *36*, 706 – 723.
- [208] E. Steckhan, *Angew. Chem.* **1986**, *98*, 681 – 699; *Angew. Chem. Int. Ed. Engl.* **1986**, *25*, 693.
- [209] J.M. Savéant, E. Vianello, *Adv. in Polarography* **1960**, *2*, 367 – 374.
- [210] J.M. Savéant, E. Vianello, *Electrochim. Acta* **1965**, *10*, 905 – 920.
- [211] M.D. Ryan, *J. Electrochem. Soc.* **1978**, *125*, 547 – 555.
- [212] K. Hinkelmann, J. Heinze, *Ber. Bunsenges. Phys. Chem.* **1987**, *91*, 243 – 249.
- [213] S. Daniele, G. Bontempelli, F. Magno, M. Fiorani, *Ann. d. Chim.* **1988**, *78*, 363 – 379.
- [214] Z. Rongfeng, D.H. Evans, *J. Electroanal. Chem.* **1995**, *385*, 201 – 207.
- [215] K. Ludwig, M.G. Quintanilla, Bernd Speiser, Andreas Stauß, *J. Electroanal. Chem.* **2002**, *531*, 9 – 18.
- [216] I. Chorkendorff, J.W. Niemantsverdriet, *Concepts of Modern Catalysis and Kinetics*, Wiley, Weinheim, **2003**.
- [217] V. Mirčeski, *J. Phys. Chem. B* **2004**, *108*, 13719 – 13725.
- [218] J.F. Cassidy, S. Pons, A.S. Hinman, B. Speiser, *Can. J. Chem.* **1984**, *62*, 716 – 720.

- [219] B. Speiser, *J. Electroanal. Chem.* **1996**, *413*, 67 – 79.
- [220] F. Martinez-Ortiz, A. Molina, C. Serna, *J. Electroanal. Chem.* **1991**, *308*, 97 – 112.
- [221] M.-L. Alcaraz, Á. Molina, M. López-Tenés, *Electrochim. Acta* **1997**, *42*, 1351 – 1359.
- [222] <http://www.novell.com/linux>, Juni 2006.
- [223] <http://www.gnu.org>, Juni 2006.
- [224] <http://opensource.org/licenses/gpl-license.php>, Juni 2006.
- [225] <http://www.trolltech.com>, Juni 2006.
- [226] <http://www.grinninglizard.com/tinyxml>, Juni 2006.
- [227] <http://public.kitware.com/VTK/>, Juni 2006.
- [228] K. Ludwig, L. Rajendran, B. Speiser, *J. Electroanal. Chem.* **2004**, *571*, 119.
- [229] S. Buchmann, H.A. Mayer, B. Speiser, M. Seiler, M.P. Feth, H. Bertagnolli, S. Steinbrecher, E. Plies, *Electrochim. Acta* **2003**, *48*, 4319 – 4320.
- [230] <http://www.iupac.org>, Juli 2006.

Abbildungsverzeichnis

2.1. Skizze zur Zielsetzung	5
3.1. Elektrodenprozesse	9
3.2. Modellgeometrien	11
3.3. Anregungsfunktionen	12
3.4. Reaktions- und Speziesindizierung	16
4.1. Ecco's Eingabesprache: Grundstruktur	43
4.2. Ecco's Eingabesprache: Beispiele	45
4.3. Ecco's Eingabesprache: Komplexe Kinetik	47
4.4. Ecco's Eingabesprache: Grammatik	54
4.5. Ecco's Eingabesprache: Parserbaum	55
4.6. Fassade-Muster	56
4.7. Befehl-Muster	57
4.8. Interaktionsdiagramm des Syntheseteils	58
4.9. Proxy-Muster	60
4.10. Reaktionsnetzwerk-Klassen	61
4.11. Kompositum-Muster: Law-Klassenhierarchie	62
4.12. Flussdiagramm zum <i>CurrentResponse</i> -Objekt	66
4.13. Beobachter-Muster für TSRs	71
5.1. Adapter-Bibliothek	76
6.1. Finite Elemente (1D)	90
6.2. Vergrößerungsfehler	95
6.3. Übersicht Multilevel-Algorithmus	99
6.4. Strategie-Muster: Rothe-Rosenbrock-Klassen	102
6.5. MFEMSolver-Klassen-Template	103
6.6. Interaktionsdiagramm des MFEM-Algorithmus	104
6.7. Kopplung an Ecco	105
7.1. Chronoamperometrische Anregungsfunktionen	109
7.2. Cottrell-Simulation: Ströme	110
7.3. Cottrell-Simulation: Methoden-Charakteristik	111
7.4. Cottrell-Simulation: Konzentrationsprofil	113

7.5. CA-Simulation: Quasi-reversibler Fall	115
7.6. DPCA-Simulation: Ströme	117
7.7. DPCA-Simulation: Methoden-Charakteristik	118
7.8. MPCA-Simulation: Ströme und Gitterevolution	119
7.9. Reinert-Berg-System	121
7.10. CV-Simulation: Anregungsfunktion und Fehlertoleranzen für einfachen ET	122
7.11. CV-Simulation: Einfacher ET	123
7.12. CV-Simulation: Methoden-Charakteristik	124
7.13. LSV-Simulation: Katalytische Folgereaktion	126
7.14. LSV-Simulation: Methoden-Charakteristik	127
7.15. CV-Simulation: EE-Disp	129
7.16. CV-Simulation: EE-Disp Konzentrationsprofile	130
7.17. CV-Simulation: Elektroaktive Adsorptionsschicht	131
7.18. CV-Simulation: Adsorptionssystem	134
7.19. Oberflächenreaktionen	136
7.20. CV-Simulation: Adsorptionssystem mit Oberflächenreaktionen	137
7.21. CV-Simulation: Dünnschichtzelle	139
7.22. CV-Simulation: Konzentrationsprofile in Dünnschichtzellen	140
7.23. SECM-Experiment und Simulation	141
7.24. CA-Simulation: Doppel-Elektroden-Dünnschichtzelle	143
7.25. CP-Simulation: Konstante Anregung	147
7.26. CP-Simulation: Konvergenz bei konstanter Anregung	148
7.27. CP-Simulation: Anfangssingularitäten	150
7.28. CP-Simulation: Methoden-Charakteristik	151
7.29. CP-Simulation: Diverse Anregungen	153
7.30. CP-Simulation: Folgereaktion erster Ordnung	154
A.1. Generisches Beosbacher-Muster	164

Tabellenverzeichnis

3.1. Anregungsfunktionen	13
3.2. Normalisierungen	31
4.1. Ecco's Eingabesprache: Komponenten	44
4.2. Ecco's Eingabesprache: Kinetische Optionen	46
4.3. Ecco's Eingabesprache: Komponenten eines generischen Gesetzes	48
4.4. Befehlsklassen der Ecco-Bibliothek	59
6.1. Rosenbrock-Methoden	101
7.1. Cottrell-Simulation: Fehlertoleranzen	112
7.2. DPCA-Simulation: Fehlertoleranzen	116
7.3. CV-Simulation: Modellparameter für Adsorptionssystem	135

Abkürzungsverzeichnis

ASCII	American Standard Code for Information Interchange
CA	Chronoamperometrie
CP	Chronopotentiometrie
CRDEC	Cyclic Reciprocal Derivative Exponential Chronopotentiometry
CRDPC	Cyclic Reciprocal Derivative Power Chronopotentiometry
CV	Cyclische Voltammetrie / Cyclovoltammogramm
DAE	Differentiell-Algebraische Gleichung (Differential-Algebraic Equation)
DFG	Deutsche Forschungsgemeinschaft
DPCA	Doppel-Puls-Chronoamperometrie
EBNF	Extended Backus-Naur Form[129]
ECL	Elektrochemilumineszenz
ET	Heterogener Elektronentransfer
FE	Finite-Elemente
GG	Gleichgewicht
GoF	Gang of Four (vgl. Fußnote S.55)
ID	Identifikationsnummer
IUPAC	International Union of Pure and Applied Chemistry[230]
LGS	Lineare Gleichungssystem
LSV	Linear Sweep Voltammetry
MPCA	Multi-Puls-Chronoamperometrie
NAPL	Non-Aqueous Phase Liquids

ODE	Gewöhnliche Differentialgleichung (Ordinary Differential Equation)
OET	Oberflächenelektronentransfer (ET eines adsorbierten Teilchens)
OMT	Object Modeling Technique[130]
OO	Objekt-orientiert
OOA	Objekt-orientierte Analyse
OOD	Objekt-orientiertes Design
OSS	Open-Source-Software
PDAE	Partielle Differentiell-Algebraische Gleichung (Partial Differential-Algebraic Equation)
PDE	Partielle Differentialgleichung (Partial Differential Equation)
PSE	Problem Solving Environment
SEC	Spektroelektrochemie
SECM	Scanning Electrochemical Microscopy
SI	Surface Index
STL	Standard Template Library[131]
SWV	Square Wave Voltammetry
TSR	Thermodynamisch überflüssige Reaktion (Thermodynamically Superfluous Reaction[74])
VET	Volumenelektronentransfer (ET eines gelösten Teilchens)

Meine akademischen Lehrer an der Eberhard-Karls-Universität Tübingen waren:

K. Albert, D. Christen, G. Gauglitz, M. Hanack, G. Häfeling, N. Kuhn, E. Lindner, M.E. Maier, H.A. Mayer, H.-J. Meyer, U. Nagel, H. Oberhammer, Ch. Ochsenfeld, B. Speiser, J. Strähle, K.-P. Zeller, T. Ziegler (Chemie)

G. Betsch, K. Engel, W. Grölz, C. Hering, W. Kaup, P. Leinen, F. Loose, C. Lubich, M. Oliver, P. Schmid, M. Voit, H. Yserentant (Mathematik)

H.-G. Klinzing (Erziehungswissenschaft)

Lebenslauf

Persönliche Daten

Name: Kai Ludwig
Geburtstag und -ort: 12.03.1976, Reutlingen
Staatsangehörigkeit: deutsch
Familienstand: ledig

Werdegang

1982 – 1986 Achalmschule in Eningen u.A.
1986 – 1995 Isolde-Kurz-Gymnasium in Reutlingen
10/1995 – 10/1996 Zivildienst im Kreiskrankenhaus Reutlingen
10/1996 – 11/2002 Eberhard-Karls-Universität Tübingen
Studium der Mathematik und der Chemie
(Abschluss: Staatsexamen)
12/2000 – 05/2001 Wissenschaftliche Arbeit im Fach Chemie
(Betreuer: Prof. Dr. B. Speiser)
12/2002 – 7/2006 Doktorarbeit mit dem Titel:
“Objekt-orientierte Modellierung und Simulation dynamischer Elektrodenprozesse in der molekularen Elektrochemie. Eine Anwendung der adaptiven Multilevel-Finite-Elemente-Methode”
Institut für Organische Chemie, Tübingen
(Betreuer: Prof. Dr. B. Speiser)