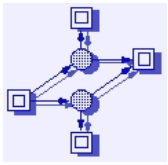


Lehrstuhl Datenstrukturen und
Softwarezuverlässigkeit

Diplomarbeit

Brandenburgische Technische
Universität Cottbus

Ein Werkzeug zum Entwerfen und Simulieren kontinuierlicher Petrinetze

Daniel Scheibler

daniel.scheibler@informatik.tu-cottbus.de

15. Februar 2006

Betreuer / Gutachter : Prof. Dr.-Ing. Monika Heiner
Prof. Dr.-Ing. habil. Hartmut König

Eidesstattliche Erklärung

Die vorliegende Diplomarbeit wurde von mir selbständig angefertigt. Die verwendeten Hilfsmittel und Quellen sind im Literaturverzeichnis vollständig aufgeführt.
Eingetragene Warenzeichen und Copyrights werden anerkannt, auch wenn sie nicht explizit gekennzeichnet sind.

Cottbus, den 15. Februar 2006

Daniel Scheibler

Danksagung

An dieser Stelle möchte ich mich bei allen Personen für die vielseitige Unterstützung während meiner Diplomarbeitsphase bedanken. Einen besonderen Dank verdienen die folgenden Personen:

Frau Prof. Heiner, die mich während der Diplomarbeit betreut hat und immer gute Hinweise und Denkanstöße parat hatte.

Ronny Richter und Alexey A. Tovchigrechko, die mir oftmals praktische Hilfestellung bei der Ausarbeitung dieser Arbeit und beim Entwickeln des Werkzeuges gaben.

Gerd und Gabriele Scheibler, dass sie mich auf all meinen Wegen unterstützt und begleitet haben.

Falk Hamann, Kristin Hoyer, Skander Morgenthaler, Miriam Padberg, Lars Schönberg, Kristin Winkler und Joerg Zeltner möchte ich für die freundliche Unterstützung beim Korrekturlesen danken.

Inhaltsverzeichnis

Aufgabenstellung **1**

1 Einleitung **3**
1.1 Motivation . . . 3
1.2 Ziele der Arbeit . . . 3
1.3 Vorgehen . . . 4
1.4 Aufbau der Diplomarbeit . . . 4

2 Kontinuierliche Petrinetze **5**
2.1 Einordnung . . . 5
2.2 Definition . . . 6
2.3 Erweiterungen . . . 7
2.3.1 Lese- und Inhibitorkanten . . . 8
2.3.2 Kontinuierliche Parameter . . . 8
2.3.3 Fixierte kontinuierliche Plätze . . . 9
2.3.4 Umkehrbare kontinuierliche Transitionen . . . 10
2.4 Zusammenfassende Definition . . . 11

3 Mathematische Grundlagen der Simulation **13**
3.1 Differentialgleichungen . . . 13
3.2 Grundalgorithmus . . . 14
3.3 Explizite m-stufige Runge-Kutta-Verfahren . . . 15
3.3.1 Explizite RK-Verfahren mit fester Schrittweite . . . 16
3.3.2 Explizite RK-Verfahren mit Schrittweitensteuerung . . . 18
3.4 Rosenbrock-Verfahren . . . 20

4 Implementierung **25**
4.1 Analyse bestehender Programmpakete . . . 25
4.1.1 Gepasi / Copasi . . . 25
4.1.2 Visual Object Net / Genomic Object Net / Cell Illustrator . . . 26
4.1.3 ModelKinetix ModelMaker . . . 27
4.2 Entwurfsentscheidungen für Snoopy . . . 29
4.2.1 Darstellung der kontinuierlichen Petrinetze . . . 29
4.2.2 Verwendung von Gleichungsmustern . . . 29
4.2.3 Übertragung kont. Petrinetze in gew. Differentialgleichungen . . . 30

4.2.4 Konfigurationsmöglichkeiten der Simulationen . . . 30
4.2.5 Ergebnisdarstellung . . . 31
4.3 Validierung . . . 31

5 Beispielanwendungen 33
5.1 Künstlich konstruierte Beispiele . . . 33
5.1.1 Lineare Beispiele - Vorbetrachtungen . . . 34
5.1.2 Lineares Beispiel - BLR . . . 37
5.1.3 Lineares Beispiel - ¬BLR . . . 37
5.1.4 Lineares Beispiel - B¬LR . . . 38
5.1.5 Lineares Beispiel - ¬B¬LR . . . 39
5.1.6 Lineares Beispiel - BL¬R . . . 40
5.1.7 Lineares Beispiel - ¬BL¬R . . . 40
5.1.8 Lineares Beispiel - B¬L¬R . . . 41
5.1.9 Lineares Beispiel - ¬B¬L¬R . . . 42
5.2 Praktische Beispiele aus der aktuellen Forschung . . . 42
5.2.1 RKIP-Beispiel . . . 43
5.2.2 MAPK-Kaskaden-Beispiel von Kholodenko . . . 44
5.2.3 Leber-Beispiel von Knoblauch/Reich . . . 44
5.2.4 Kartoffel-Beispiel von Junker . . . 44
5.2.5 Hefe-Beispiel von Klipp/Kofahl . . . 47

6 Erkenntnisse 49

7 Zusammenfassung und Ausblick 53
7.1 Zusammenfassung . . . 53
7.2 Ausblick und weitergehende Aufgaben . . . 54

A Implementierungsstruktur 55
A.1 Netzklasse Continuous Petri Net . . . 55
A.1.1 Kontinuierliche Petrinetzknoten . . . 56
A.1.2 Kontinuierliche Petrinetzkanten . . . 57
A.1.3 Export in SBML . . . 57
A.1.4 Export in LaTeX . . . 58
A.2 Simulation . . . 58
A.3 Auswertung der Simulationsergebnisse . . . 60

B Petrinetze und Simulationsergebnisse 61
B.1 RKIP-Beispiel . . . 62
B.2 MAPK-Kaskaden-Beispiel von Kholodenko . . . 69
B.3 Leber-Beispiel von Reich . . . 73
B.4 Kartoffel-Beispiel von Junker . . . 82
B.5 Hefe-Beispiel von Klipp/Kofahl . . . 90

Literaturverzeichnis 109

Abbildungsverzeichnis

2.1 Kontinuierliche Netzelemente 7
2.2 Ein Beispiel zur Konstruktion der gewöhnlichen Differentialgleichungen 8
2.3 Kontinuierliche Lese- und Inhibitorkante und ein Beispiel 9
2.4 Kontinuierliche Parameter, Parameterkanten und ein Beispiel 10
2.5 Fixierter kontinuierlicher Platz und zwei Beispiele 10
2.6 Umkehrbare kontinuierliche Transition 11

4.1 Gepasi 26
4.2 Copasi 27
4.3 Genomic Object Net 28
4.4 ModelKinetix ModelMaker 28
4.5 Snoopy: Darstellung der kontinuierlichen Petrinetzelemente 29

5.1 Lineare Beispiele: Erlaubte Strukturen 35
5.2 Lineares Beispiel: Verbotene Struktur 35
5.3 Lineare Beispiele: Kurvenverlaufsdiagramme 36
5.4 Lineares Beispiel: BLR - kontinuierliches Petrinetz, INA-Ergebnisvektor 37
5.5 Lineares Beispiel: ¬BLR - kontinuierliches Petrinetz, INA-Ergebnisvektor 38
5.6 Lineares Beispiel: B¬LR - kontinuierliches Petrinetz, INA-Ergebnisvektor 39
5.7 Lineares Beispiel: ¬B¬LR - kontinuierliches Petrinetz, INA-Ergebnisvektor . . . 40
5.8 Lineares Beispiel: ¬BL¬R - kontinuierliches Petrinetz, INA-Ergebnisvektor . . . 41
5.9 Lineares Beispiel: B¬L¬R - kontinuierliches Petrinetz, INA-Ergebnisvektor . . . 42
5.10 Lineares Beispiel: ¬B¬L¬R - kontinuierliches Petrinetz, INA-Ergebnisvektor . . 43
5.11 RKIP: Hauptstruktur 44
5.12 RKIP: Parameter 44
5.13 MAPK-Kaskade: Parameter 44
5.14 MAPK-Kaskade: Hauptstruktur 45
5.15 Leber: Hauptstruktur 45
5.16 Leber: Parameter 45
5.17 Kartoffel: Hauptstruktur 46
5.18 Kartoffel: Parameter 46
5.19 Hefe: Parameter 47
5.20 Hefe: Hauptstruktur 48

6.1 Umkehrbare Transition und zwei Transitionen 50
6.2 Beispiel für Oszillation und Gleichgewichtszustand 50

A.1 Snoopy: Eigenschaftendialog für kontinuierliche Transitionen 56
A.2 Snoopy: Eigenschaftendialoge 57
A.3 Snoopy: Auswahldialog für Export in LaTeX 59
A.4 Snoopy: Simulation 59

B.1 Legende 62
B.2 RKIP: Legende 63
B.3 RKIP: Absoluter Fehler bei Zeitschritt 50 66
B.4 RKIP: Kurvenverlaufsdiagramme 67
B.5 RKIP: Kurvenverlaufsdiagramme (Forts.) 68
B.6 MAPK-Kaskade: Absoluter Fehler bei Zeitschritt 5000 69
B.7 MAPK-Kaskade: Kurvenverlaufsdiagramme 71
B.8 Leber: Absoluter Fehler bei Zeitschritt 15 76
B.9 Leber: Kurvenverlaufsdiagramme 77
B.10 Leber: Kurvenverlaufsdiagramme (Forts.) 78
B.11 Leber: Unterstrukturen 79
B.12 Kartoffel: Absoluter Fehler bei Zeitschritt 5500 85
B.13 Kartoffel: Kurvenverlaufsdiagramme 86
B.14 Kartoffel: Kurvenverlaufsdiagramme (Forts.) 87
B.15 Hefe: Absoluter Fehler bei Zeitschritt 200 95
B.16 Hefe: Absoluter Fehler bei Zeitschritt 200 (Forts.) 96
B.17 Hefe: Absoluter Fehler bei Zeitschritt 200 (Forts.) 97
B.18 Hefe: Kurvenverlaufsdiagramme 98
B.19 Hefe: Kurvenverlaufsdiagramme (Forts.) 99
B.20 Hefe: Kurvenverlaufsdiagramme (Forts.) 100
B.21 Hefe: Kurvenverlaufsdiagramme (Forts.) 101
B.22 Hefe: Kurvenverlaufsdiagramme (Forts.) 102
B.23 Hefe: Unterstrukturen 103
B.24 Hefe: Unterstrukturen (Forts.) 104

Tabellenverzeichnis

B.1 RKIP: Berechnungszeiten 63
B.2 RKIP: Absoluter Fehler bei Zeitschritt 50 64
B.3 RKIP: Relativer Fehler bei Zeitschritt 50 65
B.4 MAPK-Kaskade: Berechnungszeiten 69
B.5 MAPK-Kaskade: Absoluter Fehler bei Zeitschritt 5000 70
B.6 MAPK-Kaskade: Relativer Fehler bei Zeitschritt 5000 72
B.7 Leber: Berechnungszeiten 73
B.8 Leber: Absoluter Fehler bei Zeitschritt 15 74
B.9 Leber: Relativer Fehler bei Zeitschritt 15 75
B.10 Kartoffel: Berechnungszeiten 82
B.11 Kartoffel: Absoluter Fehler bei Zeitschritt 5500 83
B.12 Kartoffel: Relativer Fehler bei Zeitschritt 5500 84
B.13 Hefe: Berechnungszeiten 90
B.14 Hefe: Absoluter Fehler bei Zeitschritt 200 90
B.15 Hefe: Absoluter Fehler bei Zeitschritt 200 (Forts.) 91
B.16 Hefe: Absoluter Fehler bei Zeitschritt 200 (Forts.) 92
B.17 Hefe: Relativer Fehler bei Zeitschritt 200 93
B.18 Hefe: Relativer Fehler bei Zeitschritt 200 (Forts.) 94

Aufgabenstellung

Das Softwarepaket Snoopy zum Erstellen und Animieren von diskreten Petrinetzen ist um eine Netzklasse ”Continuous Petri Net” zu erweitern. Mit dieser Erweiterung sollen kontinuierliche Petrinetze, wie in einschlägiger Literatur beschrieben, erstellt und simuliert werden können.

Für die Simulation sind mehrere Differentialgleichungslöser zu implementieren. Die resultierenden Werte der einzelnen Plätze des Petrinetzes sollen graphisch ansprechend und gut auswertbar dargestellt werden. Weiterhin ist ein Export in die Zwischensprache SBML, sowie für gängige mathematische Programme, wie zum Beispiel Matlab, vorzusehen. SBML ist ein in der Biochemie gängiges Austauschformat für Analyseprogramme, wie zum Beispiel Gepasi/Copasi.

Beim Erstellen der kontinuierlichen Petrinetze sollen die Flussgleichungen einer Konsistenzprüfung unterzogen werden. Diese soll die verwendeten Platz- respektive Variablennamen und die Wiedergabe dieser Relationen durch Kanten des Petrinetzes überprüfen. Das hat besondere Bedeutung bei der Verwendung von Testkanten, die sonst in den Flussgleichungen nur schlecht identifizierbar beziehungsweise schwer überschaubar sind.

Bei der Implementierung ist darauf zu achten, dass eine mögliche Erweiterung in hybride Petrinetze einfach umzusetzen ist.

Kapitel 1

Einleitung

Das Ziel der hier vorliegenden Diplomarbeit ist es, ein Werkzeug zum Entwurf und zur Simulation kontinuierlicher Petrinetze zu entwickeln. Für die Simulation der gegebenen kontinuierlichen Petrinetze wird auf die numerische Lösung gewöhnlicher Differentialgleichungen zurückgegriffen.

1.1 Motivation

In der Biochemie wird viel mit gewöhnlichen Differentialgleichungen zur Beschreibung von Prozessen aller Art, wie zum Beispiel Stoffwechselprozesse innerhalb eines Organismus oder eines Organes, gearbeitet. Diese Differentialgleichungssysteme können dabei schnell sehr groß und für mit dem Problem nicht vertraute Personen schwer verständlich werden. Um die Verständlichkeit zu erhöhen, ist es ein Ansatz, die Differentialgleichungen in Form eines kontinuierlichen Petrinetzes darzustellen. Dadurch wird die Struktur des Problems deutlicher und es kann einfacher zu einer Diskussion darüber kommen.

Ergänzend zum Verständnisgewinn, der sicherlich auch mit Hilfe anderer Visualisierungsmethoden erreichbar ist, soll die Arbeit Ansätze zur Übertragung von Eigenschaften diskreter Petrinetze auf kontinuierliche Petrinetze andeuten und ihre Übertragbarkeit auf Probleme mit veränderlichen kontinuierlichen Flüssen aufzeigen.

1.2 Ziele der Arbeit

Zielstellung der vorliegenden Arbeit ist die Entwicklung eines Werkzeuges zum Entwurf und zur Simulation von kontinuierlichen Petrinetzen. Das Werkzeug soll auf das von Markus Fieber in [Fie04] entwickelte Werkzeug **Snoopy** aufbauen. Snoopy ist plattformunabhängig in der Programmiersprache *C++* mit dem Fenstersystem *wxWidgets* entwickelt worden. Weitergehende Informationen zum Aufbau und zu den Möglichkeiten des Werkzeuges können der Diplomarbeit von Markus Fieber entnommen werden. Des Weiteren soll die Anwendung des Werkzeuges zur Simulation und Analyse von biochemischen Beispielanwendungen zur exemplarischen Verdeutlichung der gebotenen Möglichkeiten aufgezeigt werden.

1.3 Vorgehen

Zum Erreichen der Zielstellungen dieser Arbeit werden zuerst die kontinuierlichen Petrinetze eingeführt, danach wird auf die Besonderheiten der Simulation kontinuierlicher Petrinetze eingegangen. Weiterhin werden existierende Werkzeuge zur Simulation kontinuierlicher Aufgabenstellungen untersucht und Entscheidungen für das zu entwerfende Werkzeug getroffen. Mit dem entwickelten Werkzeug werden mehrere Beispielanwendungen, für die größtenteils Vergleichsmodelle für eines der bereits vorhandenen Werkzeuge existieren, erstellt und simuliert. Die funktionelle Prüfung erfolgt dann an Hand der Vergleichsdaten aus dem unter Biochemikern anerkannten Werkzeug **Copasi**.

1.4 Aufbau der Diplomarbeit

Nach diesem einleitenden Kapitel werden die kontinuierlichen Petrinetze definiert und eingeordnet. Zusätzlich werde ich im Kapitel 2 eine kurze Einführung in Differentialgleichungen geben. Im Kapitel 3 wird auf die mathematischen Grundlagen der Simulation eingegangen und es werden einige Schwierigkeiten bei der Simulation aufgezeigt. Danach wird sich der konkreten Implementierung gewidmet. Zu diesem Zweck wurden zuerst einige existierende Programmpakete analysiert und ihre Stärken und Schwächen aufgezeigt. Darauf aufbauend folgen die getroffenen Entwurfsentscheidungen für die Simulation im Werkzeug Snoopy. Zum Nachweis der Praxistauglichkeit des entwickelten Werkzeuges werden im Kapitel 5 einige Beispielanwendungen vorgestellt und mit Ergebnissen des in der biochemischen Forschung anerkannten Werkzeuges Gepasi verglichen. Dazu werden die modellierten Beispielanwendungen mittels des standardisierten Austauschformates SBML aus Snoopy exportiert und in Gepasi importiert. Aus der Simulation und Analyse der Beispielanwendungen werden erste Erkenntnisse zu Zusammenhängen zwischen dem beobachtbaren Verhalten und den Struktureigenschaften des modellierten Petrinetzes formuliert und in Kapitel 6 zur Diskussion gestellt. Dabei wird es sich größtenteils um Vermutungen handeln, die in dieser Arbeit nicht bewiesen werden können. Abschließend wird sowohl eine Zusammenfassung als auch ein Ausblick angegeben.

Kapitel 2

Kontinuierliche Petrinetze

Kontinuierliche Petrinetze stellen eine Erweiterung der nach Carl Adam Petri benannten diskreten Petrinetze dar. In diesem Kapitel werden die Petrinetze speziell in ihrer kontinuierlichen Form eingeordnet und danach wird eine Definition der verwendeten Netzklasse angegeben.

2.1 Einordnung

Die Einordnung der verwendeten kontinuierlichen Petrinetze bezieht sich im wesentlichen auf das Buch [DA05] von David und Alla, sowie auf die Diplomarbeit von Kristina Lonitz [Lon05]. Historisch werden die Petrinetze auf die Dissertation von Carl Adam Petri [Pet62] mit dem Titel „Kommunikation mit Automaten“ aus dem Jahre 1962 zurückgeführt. In dieser Arbeit wird ein mathematisches Modell zur Beschreibung von Relationen zwischen Bedingungen und Ereignissen beschrieben.

Mit Hilfe von diskreten Petrinetzen können einige Verhaltenstypen wie Parallelität, Nebenläufigkeit, Synchronisation und geteilte Ressourcen einfach modelliert und dargestellt werden. Gleichzeitig können aufgrund der genauen formalen Definition viele Rückschlüsse auf Eigenschaften der modellierten Netze gezogen und bewiesen werden. Bekannte und aussagekräftige Eigenschaften sind zum Beispiel

- Beschränktheit
- Rücksetzbarkeit
- Lebendigkeit
- Erreichbarkeit
- T-Invarianten
- P-Invarianten

Für genauere Informationen und Definitionen der genannten Eigenschaften sei auf [Sta90], [Rei91] und [Bau96] verwiesen.

Die Ausdrucksstärke der diskreten Petrinetze mit ihrer Beschränkung auf einen diskreten Zustandsraum wurden durch die Erweiterung zu kontinuierlichen Petrinetzen von David und Alla

1987 [DA87] wesentlich vergrößert. Sie führen Plätze mit positiven reellwertigen Marken und eine kontinuierliche Feuerregel mit einer variablen Feuergeschwindigkeit ein. Ihre Erweiterungen, bis hin zu hybriden Petrinetzen, werden in [DA92] und [DA05] formal definiert und tiefgreifend diskutiert.

Für weitergehende Informationen zu Petrinetzen und existierenden Erweiterungen ist folgende Literatur empfehlenswert: [PNW], [GC], [Sta90], [Rei91], [Bau96], [DA05], [SR05] und [Lon05].

2.2 Definition

In diesem Abschnitt werden die kontinuierlichen Petrinetze definiert und danach um einige nützliche Eigenschaften erweitert. Diese erweiterten Eigenschaften dienen dazu, die Modelle übersichtlicher zu machen und die Simulation parametrisierbar zu gestalten.

Definition 2.2.1 (Kontinuierliches Petrinetz mit variablen Geschwindigkeiten, informal)
Ein kontinuierliches Petrinetz mit variablen Geschwindigkeiten ist ein Petrinetz mit reellwertigen Plätzen und einer kontinuierlichen Feuerregel. Die Schaltgeschwindigkeit berechnet sich aus einer Geschwindigkeitsfunktion, die jeder Transition zugeordnet wird und von den Werten der Plätze im Vorbereich der Transition abhängig ist.

Die Formalisierung dieser Definition befindet sich in Definition 2.2.2, wobei $\mathbb{R}$ die reellen Zahlen und $\mathrm{dom}(f)$ der Definitionsbereich einer Funktion f bezeichnen. Ferner wird folgende Notation für den Vorbereich und den Nachbereich eines Netzknotens vereinbart:

- ${}^\bullet x := \{y | (y,x) \in F\}$ ist die Menge von Vorknoten eines Knoten x
- $x^\bullet := \{y | (x,y) \in F\}$ ist die Menge von Nachknoten eines Knoten x

Definition 2.2.2 (Kontinuierliches Petrinetz mit variablen Geschwindigkeiten, formal) *Ein kontinuierliches Petrinetz mit variablen Geschwindigkeiten ist ein 6-Tupel $R = \langle P, T, F, g, v, m_0 \rangle$ mit:*

- $P = \{p_1, p_2, \ldots, p_n\}$ *ist eine endliche, nicht leere Menge von kontinuierlichen Plätzen*
- $T = \{t_1, t_2, \ldots, t_m\}$ *ist eine endliche, nicht leere Menge von kontinuierlichen Transitionen*
- $P \cap T = \emptyset$*, die Mengen P und T sind disjunkt*
- $F \subseteq (T \times P) \cup (P \times T)$ *ist eine Flussrelation, genannt kontinuierliche Kanten*
- $g : F \to \mathbb{R}^+ \setminus \{0\}$ *ist eine Kantengewichtungsfunktion*
- $m_0 : P \to \mathbb{R}^+$ *ist eine Funktion, genannt Startmarkierung*
- $H := \bigcup_{t \in T} \left\{ h | h : \mathbb{R}^{|{}^\bullet t|} \to \mathbb{R} \right\}$ *ist die Menge aller Geschwindigkeitsfunktionen*
- $v : T \to H$*, mit* $\mathrm{dom}(v(t)) = {}^\bullet t$*, ist eine Funktion, die jeder kontinuierlichen Transition eine Geschwindigkeitsfunktion zuordnet*

Die kontinuierlichen Netzelemente werden üblicherweise abweichend von der Darstellung diskreter Netzelemente dargestellt. In der Literatur sind dazu mehrere verschiedene Vorschläge zu finden. Dazu zählen Kreise bzw. Rechtecke mit einer Doppellinie oder mit einer breiteren grauen Linie als Abgrenzung. Im entwickelten Werkzeug werden mehrere Alternativen angeboten, die breitere graue Linie wird als Standardeinstellung verwendet. In der Abbildung 2.1 sind die bisher definierten Netzelemente kontinuierlicher Platz als Kreis und kontinuierliche Transition als Rechteck, jeweils mit breitem grauem Rand dargestellt. Dieses Design wird in allen folgenden Abbildungen kontinuierlicher Petrinetze beibehalten.

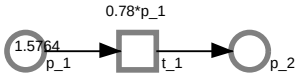


Abbildung 2.1: Kontinuierliche Netzelemente

Zur Vorbereitung des Markenspiels sind einige Vorüberlegungen zu tätigen. Zuerst müssen die Beziehungen zwischen den Plätzen und Transitionen in konkrete mathematisch auswertbare Konstrukte umgeformt werden. Diese Herausforderung wurde dadurch gelöst, dass zu Beginn der Simulation für jeden Platz eine gewöhnliche Differentialgleichung konstruiert wird. Diese gewöhnliche Differentialgleichung spiegelt die Zusammenhänge der direkten Umgebung des betreffenden Platzes wieder.

Eine gewöhnliche Differentialgleichung zur Markierungsänderung eines kontinuierlichen Platzes ist folgendermaßen definiert.

Definition 2.2.3 (Konstruktion der gewöhnlichen Differentialgleichungen) *Sei R ein kontinuierliches Petrinetz nach Definition 2.2.2, $p \in P$ ein kontinuierlicher Platz und $t \in T$ eine kontinuierliche Transition. Dann wird*

$$\frac{dm(p)}{dt} = \sum_{t \in {}^\bullet p} g(t,p)\, v(t) - \sum_{t \in p^\bullet} g(p,t)\, v(t)\,.$$

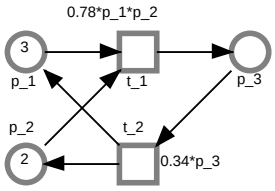
als Markierungsänderung eines kontinuierlichen Platzes bezeichnet.

Alle nach dieser Definition erstellten Flussgleichungen für die Plätze eines kontinuierlichen Petrinetzes bilden gemeinsam ein gewöhnliches, im allgemeinen nicht lineares Differentialgleichungssystem, welches mit numerisch-mathematischen Methoden behandelt werden kann. Dies kann einfach an Hand eines Beispieles verdeutlicht werden. Abbildung 2.2 zeigt ein einfaches kontinuierliches Petrinetz mit den darauf folgenden gewöhnlichen Differentialgleichungen.

Die zur Berechnung der gewöhnlichen Differentialgleichungen, notwendigen mathematischen Grundlagen werden im Kapitel 3 einführend vorgestellt. Dabei wird besonders auf die beiden Klassen der verwendeten Algorithmen eingegangen.

2.3 Erweiterungen

Ergänzend zu den in Abschnitt 2.2 definierten kontinuierlichen Petrinetzen werden in diesem Abschnitt einige Erweiterungen eingeführt. Für jede neue Eigenschaft wird kurz erläutert, warum



$$\begin{aligned}
\frac{dp_1}{dt} &= g(t_2,p_1)\,v(t_2) - g(p_1,t_1)\,v(t_1) &= 0.34 * p_3 - 0.78 * p_1 * p_2 \\
\frac{dp_2}{dt} &= g(t_2,p_2)\,v(t_2) - g(p_2,t_1)\,v(t_1) &= 0.34 * p_3 - 0.78 * p_1 * p_2 \\
\frac{dp_3}{dt} &= g(t_1,p_3)\,v(t_1) - g(p_3,t_2)\,v(t_2) &= 0.78 * p_1 * p_2 - 0.34 * p_3
\end{aligned}$$

Abbildung 2.2: Ein Beispiel zur Konstruktion der gewöhnlichen Differentialgleichungen

selbige integriert wurde und wie sie sich auf die Simulation der Petrinetze auswirkt. Diese Erweiterungen sind größtenteils der effizienteren Arbeit mit dem Werkzeug geschuldet, teilweise greifen sie auch relativ tief in die Simulation ein.

2.3.1 Lese- und Inhibitorkanten

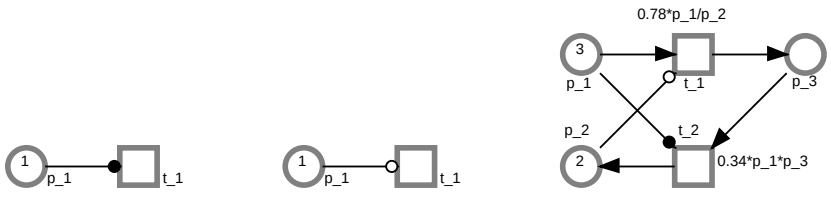
In Ergänzung zur klassischen Bedeutung von Kanten in Petrinetzen wurden zwei weitere Kantentypen eingeführt. Beide Kantenarten dienen nur zur Verbindung von Plätzen nach Transitionen, zwischen denen kein Markenfluss stattfindet, die Markierungen der Plätze aber Einfluss auf die Berechnung der Flussgeschwindigkeit der betreffenden Transition nehmen. Semantisch an bekannte Erweiterungen diskreter Petrinetze anlehnend, werden dazu zwei Kantentypen mit verschiedenen Bedeutungen verwendet. Zum einen ist dies die **Lesekante** und zum anderen die **Inhibitorkante**.

Im Unterschied zu den diskreten Petrinetzen bewirkt die kontinuierliche Lesekante nicht nur einen reinen Markierungstest, sondern erhöht die Flussgeschwindigkeit mit steigender Markierung des gelesenen kontinuierlichen Platzes.

Analog verringert die kontinuierliche Inhibitorkante den Markenfluss umgekehrt proportional zur Markierung des inhibierenden kontinuierlichen Platzes. Diese beiden zusätzlichen Kantenarten werden im Petrinetz-Modell mittels einer Linie mit einem gefüllten bzw. einem ungefüllten Kreis am Ende dargestellt (siehe Abbildung 2.3 (a) und (b)). Ein einfaches Beispiel zur Anwendung der beiden Kantenarten ist in Abbildung 2.3 (c) dargestellt und erzeugt die darauf folgenden Differentialgleichungen.

2.3.2 Kontinuierliche Parameter

Ein neues Konzept zur einfachen Parametrisierung der zugrunde liegenden gewöhnlichen Differentialgleichungen stellen die **kontinuierlichen Parameter** dar. Sie sind eine einfache Auslagerung von beliebigen, aber festen Werten, welche innerhalb des Simulationsprozesses nicht verändert werden. Alternativ können die Werte auch direkt in der Geschwindigkeitsfunktion der betreffenden Transition festgelegt werden. Zur Fehlervermeidung und leichteren Änderbarkeit von Parametern wurden diese in eigenständige Netzelemente ausgelagert. Diese kontinuierlichen



(a) (b) (c)

$$\begin{aligned}
\frac{dp_1}{dt} &= -g(p_1,t_1)\,v(t_1) &&= -\frac{0.78 * p_1}{p_2}\\
\frac{dp_2}{dt} &= g(t_2,p_2)\,v(t_2) &&= 0.34 * p_1 * p_3\\
\frac{dp_3}{dt} &= g(t_1,p_3)\,v(t_1) - g(p_3,t_2)\,v(t_2) &&= \frac{0.78 * p_1}{p_2} - 0.34 * p_1 * p_3
\end{aligned}$$

Abbildung 2.3: Kontinuierliche Lese- und Inhibitorkante und ein Beispiel

Parameter werden als global gültig bewertet und sind für jede Transition ohne Kante zwischen einem Parameter und dieser Transition sichtbar. Vergleichbar sind die kontinuierlichen Parameter mit der mehrfachen Verwendung von Konstanten anstelle von Direktwerten in Programmiersprachen. In kontinuierlichen Petrinetzen werden kontinuierliche Parameter als Ellipse, wie in Abbildung 2.4 (a), dargestellt.

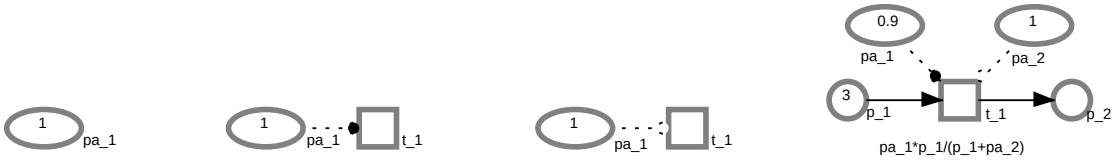
Zur besseren Modellierung der Zugehörigkeit bestimmter kontinuierlicher Parameter zu einer Transition werden zwei zusätzliche Kantentypen bereitgestellt. In Anlehnung an die kontinuierliche Lesekante wird eine **kontinuierliche Parameterlesekante** mit äquivalenter Semantik, bezogen auf kontinuierliche Parameter, sowie, analog zur kontinuierlichen Inhibitorkante, eine **kontinuierliche Parameterinhibitorkante** angeboten. Zur Unterscheidung von den in Abschnitt 2.3.1 genannten Kantentypen werden die kontinuierlichen Parameterkanten mittels gestrichelter Linien gezeichnet. Die Kreise am Linienende sind, wie in Abbildung 2.4 (b) und (c), ebenfalls gefüllt bzw. ungefüllt dargestellt. Semantisch soll mit den beiden Kantentypen eine äquivalente Einflussnahme signalisiert werden.

Kontinuierliche Parameter müssen, wie alle Netzelemente, nicht eindeutig benannt werden. Aufgrund dieser Nicht-Eindeutigkeit von Parameternamen werden Parameter, welche mit Hilfe einer Kante an eine Transition gebunden sind, den ungebundenen mit gleichem Namen gegenüber bevorzugt. Eine eindeutige Verwendung von Parametern ist über die Nutzung der eindeutigen Parameter-ID möglich. Alternativ können vom Benutzer eindeutige Parameternamen verwendet werden.

Ein einfaches Beispiel zur Anwendung der beiden Kantenarten ist in Abbildung 2.4 (d) dargestellt und erzeugt die genannten Differentialgleichungen.

2.3.3 Fixierte kontinuierliche Plätze

Bei den untersuchten biochemischen Anwendungen handelt es sich teilweise um Ausschnitte aus komplexen Stoffwechselprozessen. Bei diesen Prozessen werden Eingangs- und Ausgangsstoffe parallel zum Ablauf des untersuchten Ausschnittes erzeugt und konsumiert. Zur Modellierung dieser Produktion und des Verbrauchs werden zwei Varianten verwendet. Die erste wird durch



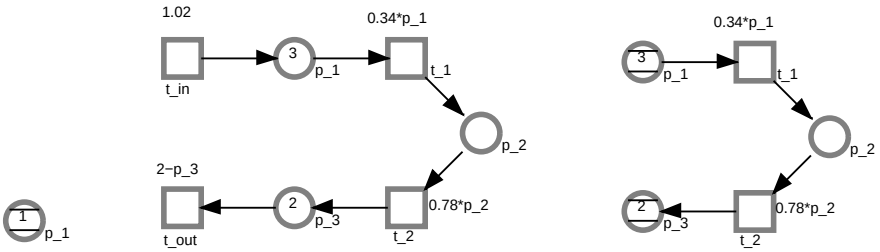
(a) (b) (c) (d)

$$\frac{dp_1}{dt} = -g(p_1,t_1)\,v(t_1) = -\frac{pa_1 * p_1}{p_1 + pa_2}$$
$$\frac{dp_2}{dt} = g(t_1,p_2)\,v(t_1) = \frac{pa_1 * p_1}{p_1 + pa_2}$$

Abbildung 2.4: Kontinuierliche Parameter, Parameterkanten und ein Beispiel

Randtransitionen realisiert, welche dafür sorgen, dass immer genügend oder idealerweise konstant viele Eingangsstoffe auf den betreffenden Plätzen zur Verfügung stehen. Diese Variante ist in einigen Anwendungen schwer zu realisieren, da die entsprechenden Erzeugungsprozesse sehr komplex sind oder noch nicht ausreichend erforscht wurden. Beide Varianten werden in den Abbildungen 2.5 (b) und (c) gegenübergestellt. Dabei stellt mittlere Abbildung die Variante mit Randtransitionen und rechte Abbildung die Variante mit fixierten Plätzen dar.

Um dieses Problem zu umgehen, wurden die kontinuierlichen Plätze um die Eigenschaft **Fixiertheit** erweitert. Diese Eigenschaft sorgt dafür, dass die Markierung der betreffenden Plätze unverändert bleibt. Dadurch wird ein konstantes Level an Markierung erzwungen, auch wenn sich die Markierung entsprechend den Schaltgeschwindigkeiten eigentlich verändern müsste. Wie in Abbildung 2.5 (a) exemplarisch gezeigt, werden die fixierten kontinuierlichen Plätze mit Hilfe von zwei dünnen waagerechten schwarzen Linien innerhalb der Umrandung dargestellt.



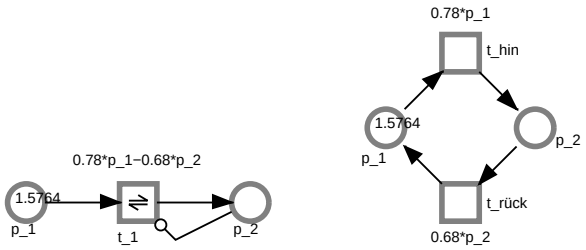
(a) (b) (c)

Abbildung 2.5: Fixierter kontinuierlicher Platz und zwei Beispiele

2.3.4 Umkehrbare kontinuierliche Transitionen

Die Analyse der untersuchten Anwendungen ergab ferner, dass es sinnvoll ist, umkehrbare Reaktionen einfach modellierbar zu machen. Dies wurde dadurch realisiert, dass Transitionen durch die Eigenschaft **Umkehrbarkeit** markiert werden können. Diese Eigenschaft hat keinen direkten Einfluss auf die Simulation des Modells, sondern muss explizit durch geeignete Wahl der zugrunde liegenden Differentialgleichung umgesetzt werden. Untersuchungen der Struktur von entspre-

chenden Differentialgleichungen haben ergeben, dass umkehrbare Transitionen in den meisten Fällen durch Differentialgleichungen der Form $v_{\text{hin}} - v_{\text{rück}}$ modelliert werden. Damit lassen sich diese Reaktionen, wegen Definition 2.2.3, im kontinuierlichen Petrinetz auch durch zwei Transitionen, jeweils eine für die Hin- und die Rückrichtung, modellieren. Dies ist in der Anwendungsdomäne der biochemischen Modellierung aber nicht immer üblich und auch nicht gewünscht, weil normalerweise einzelne chemische Reaktionen als eine Transition modelliert werden. Beide Modellierungsvarianten sind in Abbildung 2.6 dargestellt und können als äquivalent angesehen werden. Dabei sei besonders auf die Notwendigkeit der Inhibitorkante vom Platz p_2 zur Transition t_1 in Abbildung 2.6 (a) hingewiesen, da die Markierung von p_2 in die Geschwindigkeitsfunktion von t_1 eingeht.



(a) (b)

Abbildung 2.6: Umkehrbare kontinuierliche Transition

In der Simulation äußert sich diese Eigenschaft normalerweise dadurch, dass die Schaltgeschwindigkeit zeitweise einen negativen Wert annimmt. Damit wird eine stärkere Rückreaktion symbolisiert. Zur Visualisierung dieser Eigenschaft im Modell wird innerhalb des rechteckigen Transitionssymbols ein Doppelpfeil gezeichnet.

2.4 Zusammenfassende Definition

Ergänzend zur bereits in Abschnitt 2.2 gegebenen Definition kontinuierlicher Petrinetze werden **erweiterte kontinuierliche Petrinetze** eingeführt. Diese erweiterten kontinuierlichen Petrinetze beinhalten zusätzlich zu den Standardnetzelementen, kontinuierlicher Platz und kontinuierliche Transition, die im Abschnitt 2.3 aufgeführten Erweiterungen.

Für die Definition der erweiterten kontinuierlichen Petrinetze sei zusätzlich die erweiterte Menge $\hat{\bullet}t$ von Vorplätzen einer Transition $t \in T$ vereinbart, mit:

$$\hat{\bullet}t := \{p \in P | (p,t) \in (F \times I \times L)\}.$$

Definition 2.4.1 (Erweitertes kontinuierliches Petrinetz) *Ein erweitertes kontinuierliches Petrinetz mit variablen Geschwindigkeiten ist ein 9-Tupel* $S = \langle R, Q, q, L, I, L_Q, I_Q, fix, rev \rangle$ *mit:*

- *R ist ein kontinuierliches Petrinetz;*
- $Q = \{q_1, q_2, \ldots, q_k\}$ *ist eine endliche Menge von Parametern*
- $P \cap T = Q \cap P = Q \cap T = \emptyset$*, die Mengen P, T und Q sind paarweise disjunkt*

- $q : Q \to \mathbb{R}^+$ *ist eine Funktion, genannt Parametermarkierung*
- $L \subseteq P \times T$ *ist eine Relation, genannt kontinuierliche Lesekanten*
- $I \subseteq P \times T$ *ist eine Relation, genannt kontinuierliche Inhibitorkanten*
- $L_Q \subseteq Q \times T$ *ist eine Relation, genannt kontinuierliche Parameterlesekanten*
- $I_Q \subseteq Q \times T$ *ist eine Relation, genannt kontinuierliche Parameterinhibitorkanten*
- $fix : P \to \{0,1\}$ *ist eine Funktion, genannt Fixierung*
- $rev : T \to \{0,1\}$ *ist eine Funktion, genannt Umkehrbarkeit*
- $H_q := \bigcup_{t \in T} \left\{ h_q | h_q : \mathbb{R}^{|\hat{\bullet}t|} \to \mathbb{R} \right\}$ *ist die Menge aller parametrisierten Geschwindigkeitsfunktionen*
- $v_q : T \to H_q$, *mit* $\mathrm{dom}\left(v_q(t)\right) = \hat{\bullet}t$, *ist eine Funktion, die einer kontinuierlichen Transition eine parametrisierte Geschwindigkeitsfunktion zuordnet*

Die Parametrisierung der Geschwindigkeitsfunktion wirkt sich auch auf die Konstruktion der Differentialgleichungen aus und führt zu einer erweiterten Definition 2.4.2:

Definition 2.4.2 (Konstruktion der parametrisierten Differentialgleichungen) *Sei S ein erweitertes kontinuierliches Petrinetz nach Definition 2.4.1, $p \in P$ ein kontinuierlicher Platz, $t \in T$ eine kontinuierliche Transition und $q \in Q^k$, mit $k = |Q|$, ein beliebiger fester Parametersatz. Dann wird*

$$\frac{dm(p)}{dt} = \sum_{t \in {}^\bullet p} g(t,p)\, v_q(t) - \sum_{t \in p^\bullet} g(p,t)\, v_q(t)\,.$$

als parametrisierte Markierungsänderung eines kontinuierlichen Platzes bezeichnet.

In den folgenden Kapiteln dieser Arbeit wird von kontinuierlichen Petrinetzen gesprochen, gemeint sind damit aber die hier definierten erweiterten kontinuierlichen Petrinetze.

Kapitel 3

Mathematische Grundlagen der Simulation

Für die Simulation gewöhnlicher Differentialgleichungen wurden in den letzten 50 Jahren zahlreiche numerische Verfahren entwickelt, die alle ihre Vor- und Nachteile haben. Leider gibt es kein Verfahren, welches für alle Aufgabenstellungen optimal hinsichtlich Zeit- und Genauigkeitsansprüchen arbeitet. Dies macht es erforderlich, dass eine Verfahrensauswahl getroffen werden muss.

In diesem Kapitel werden zwei Klassen von Lösungsverfahren zur Simulation gewöhnlicher Differentialgleichungen vorgestellt und dabei ihre Stärken und Schwächen hervorgehoben. Dies sind einige Varianten der expliziten m-stufigen Runge-Kutta-Verfahren und Rosenbrock-Verfahren mit verschiedenen Koeffizienten. Alle vorgestellten Varianten beider Verfahren sind im Werkzeug für die Simulation verfügbar.

Einführende Literatur zu gewöhnlichen Differentialgleichungen und Verfahren zur Lösung selbiger sind vielfältig zu finden. Besonders zu empfehlen sind die mathematischen Lehrbücher [Wei84], [WS79], [Aul04] und [SW95]. Zur weitergehenden Recherche numerischer Lösungsverfahren kann prinzipiell jedes Lehrbuch zur numerischen Mathematik empfohlen werden. Gute Hilfestellungen für die Ausarbeitung dieser Diplomarbeit waren die Bücher [SW95], [Gri72] und [Gri77], das Vorlesungsskript [Han98], sowie die Webseite [Mor].

3.1 Differentialgleichungen

Zuerst sollen die bei den Differentialgleichungen üblichen Begriffe erläutert werden. Dazu gehören die Begriffe der impliziten und expliziten gewöhnlichen Differentialgleichungen und der Ordnung einer Funktion. Außerdem wird die Eigenschaft der steifen Differentialgleichungen eingeführt.

Definition 3.1.1 (Implizite gewöhnliche Differentialgleichung) *Eine gewöhnliche Differentialgleichung wird als* ***implizit*** *bezeichnet, wenn sie die Form*

$$F\left(x, y(x), y'(x), \ldots, y^{(n)}(x)\right) = 0$$

hat und $\left(x, y(x), y'(x), \ldots, y^{(n)}(x)\right)$ *zum Definitionsgebiet G von F gehört.*

Definition 3.1.2 (Ordnung) *Eine gewöhnliche Differentialgleichung hat die Ordnung n, wenn F von $y^{(n)}$ und keiner höheren Ableitung von y abhängt.*

Definition 3.1.3 (Explizite gewöhnliche Differentialgleichung) *Eine gewöhnliche Differentialgleichung n-ter Ordnung wird als **explizit** bezeichnet, wenn sie sich in der Form*

$$y^n(x) = f\left(x, y(x), y'(x), \ldots, y^{(n-1)}(x)\right)$$

mit einer Funktion f von $n+1$ Variablen darstellen lässt. ([WS79], Kapitel IV. Definition 1.1, S. 247)

Weitergehend lassen sich gewöhnliche Differentialgleichungen nach ihrem numerischen Lösungsverhalten charakterisieren. Dies führt zu einer Unterscheidung zwischen steifen und nichtsteifen gewöhnlichen Differentialgleichungen. Da in der Literatur keine einheitliche Definition zu finden ist, wurde für diese Diplomarbeit die Definition von Strehmel und Weiner aus dem Lehrbuch [SW95] gewählt, die hier kurz wiedergegeben wird:

„[..] Im Jahre 1952 beobachteten Curtiss und Hirschfelder bei der numerischen Behandlung von Differentialgleichungen chemischer Reaktionen, daß explizite Runge-Kutta-Verfahren für gewisse Probleme versagen. Sie führten den Begriff der **Steifheit** für solche chemische Reaktionen ein, bei denen sich die schnell reagierenden Komponenten in sehr kurzer Zeit einem Gleichgewichtszustand nähern, wenn die langsam veränderlichen Komponenten eingefroren, d.h. „steif“ gemacht werden.[..]“([SW95]; Abschnitt (5.2.1); S. 201)

„[..] kann man als wesentliche Eigenschaften steifer Systeme ansehen:

- Es existieren (für entsprechende Anfangswerte) langsam veränderliche Lösungen.
- Lösungen in der Umgebung dieser „glatten“ Lösungen nähern sich ihnen sehr schnell an.[..]“([SW95]; Abschnitt (5.2.1); S. 202)

Definition 3.1.4 *Wir nennen ein Anfangswertproblem steif, wenn eine logarithmische Matrixnorm existiert, so dass für $t_0 \leq t \leq t_e$ und $v \in \mathbb{R}$ aus einer Umgebung der exakten Lösung $y(t)$ für die Jacobi-Matrix f_y gilt:*

$$(t_e - t_0)\sup\mu\left[f_y(t,v)\right] = \mu_0 \ll (t_e - t_0)\sup\| f_y(t,v) \|.$$

([SW95]; Definition (5.2.1); S.202)

Strehmel und Weiner geben in Abschnitt 5.2.2 ihres Buches [SW95] drei typische Beispiele für das Auftreten steifer Systeme und verweisen auf Aiken [Aik85] für eine ausführlichere Übersicht.

3.2 Grundalgorithmus

Allen Algorithmen zur numerischen Lösung gewöhnlicher, nicht linearer Differentialgleichungssysteme liegt das selbe Prinzip zugrunde. Dabei wird aus den Informationen der bekannten Punkte der Lösung, auch Stützstellen genannt, versucht weitere Punkte zu bestimmen. Bei den einzelnen Verfahren werden unterschiedlich viele Informationen, wie zum Beispiel Lage des Punktes und

Ableitungen am Punkt, hinzugezogen. Außerdem unterscheiden sich die Verfahren in der Bestimmung der optimalen Schrittweite, mit der die nächsten Punkte bestimmt werden.

Allgemein kann der folgende Grundalgorithmus allen Verfahren zugerechnet werden. Die Schwierigkeit besteht letztendlich darin, die Berechnung von ϕ und die Wahl von h ideal zu gestalten. Dabei wird von den meisten Lösungsverfahren der Algorithmus zum Bestimmen von ϕ fest vorgegeben und die Schrittweite dynamisch bestimmt.

```
1  Gegeben:
2  t0, y0, a
3  Anfangsschrittweite h ≤ a
4  Intervall [t0, t0 + a]
5  Inkrementfunktion φ
6
7  i := 0
8  t := t0 - h
9  while t + h < t0 + a do
10    ti := t + h
11    berechne yi+1 := yi + hφ (ti, yi, h, f)
12    t := ti
13    i := i + 1
14 endwhile
15
16 h := t0 + a - t
17 yi+1 := yi + hφ (ti, yi, h, f)
```

Wie bereits einleitend erwähnt, werden in den folgenden Abschnitten 3.3 und 3.4 die beiden Verfahrensklassen der expliziten m-stufigen Runge-Kutta-Verfahren und der Rosenberg-Verfahren kurz vorgestellt und die im Werkzeug Snoopy implementierten Varianten mathematisch definiert. Bei diesen Varianten handelt es sich größtenteils um Parametrisierungen des jeweiligen Verfahrens mittels Koeffizienten.

3.3 Explizite m-stufige Runge-Kutta-Verfahren

Die expliziten m-stufigen Runge-Kutta-Verfahren sind Lösungsverfahren, die für nicht-steife Probleme besonders geeignet sind. Sie gehören zur Klasse der Einschrittverfahren. Diese Verfahren sind algorithmisch durch folgende Gleichung definiert:

$$y_{j+1} = y_j + h_j \sum_{i=1}^{m} \gamma_i k_i \left(t_j, y_j\right)$$

mit

$$\begin{aligned} k_1 &= f\left(t_j, y_j\right) \\ k_i\left(t_j, y_j\right) &= f\left(t_j + \alpha_i h_j, y_j + h_j \sum_{l=1}^{i-1} \beta_{i,l} k_l\left(t_j, y_j\right)\right) \text{ mit } \forall i \in \{2, \ldots, m\} \end{aligned}$$

Eine grundlegende Bedingung für die Wahl der Parameter α_i, β_i, γ_i stellen folgende zwei Lemmata dar.

Lemma 3.3.1 *Ein explizites Runge-Kutta-Verfahren ist mit jedem Anfangswertproblem konsistent genau dann wenn*

$$\sum_{i=1}^{m} \gamma_i = 1$$

gilt. ([SW95]; Lemma 2.4.1; S. 43)

Lemma 3.3.2 *Ferner werden die Werte für α_i im Allgemeinen durch die Knotenbedingung*

$$\alpha_i = \sum_{j=1}^{i-1} \beta_i$$

festgelegt. ([SW95]; Knotenbedingung (2.4.4); S. 43)

Die Parameter α_i, β_i, γ_i werden häufig in sogenannten Butcher-Diagrammen dargestellt:

$$\begin{array}{c|cccccc}
0 & & & & & & \\
\alpha_2 & \beta_{2,1} & & & & & \\
\vdots & \vdots & \ddots & & & & \\
\alpha_r & \beta_{r,1} & \dots & \beta_{r,r-1} & & & \\
\vdots & \vdots & & & \ddots & & \\
\alpha_m & \beta_{m,1} & \dots & \dots & \dots & \beta_{m,m-1} & \\
\hline
 & \gamma_1 & \dots & \dots & \dots & \gamma_{m-1} & \gamma_m
\end{array}$$

Ein großes Problem der expliziten Runge-Kutta-Verfahren stellt die Bestimmung der optimalen Schrittweite dar. Ein gängiges, aber nicht optimales Verfahren besteht in der Vorbestimmung einer Schrittweite und der Berechnung aller Werte unter Verwendung dieser festen Schrittweite $h_j = h \ (\forall j : 0 \leq j \leq n)$. Verfahren mit fester Schrittweite werden in Abschnitt 3.3.1 vorgestellt. Alternativ gibt es Verfahren zur dynamischen Bestimmung der Schrittweite für jeden einzelnen Berechnungsschritt. Das Grundprinzip dieser dynamischen Verfahren wird in Abschnitt 3.3.2 erklärt.

3.3.1 Explizite m-stufige Runge-Kutta-Verfahren mit fester Schrittweite

Zur Charakterisierung der expliziten Runge-Kutta-Verfahren mit fester Schrittweite reicht der in Abschnitt 3.2 eingeführte Grundalgorithmus, die in Abschnitt 3.3 definierte Funktion und die im Folgenden aufgeführten Koeffizienten aus.

Verfahren von Euler

Das Verfahren von Euler stellt das einfachste Runge-Kutta-Verfahren dar. Es besitzt die Ordnung 1 und der Berechnung wird folgendes Butcher-Diagramm zugrundegelegt:

$$\begin{array}{c|c}
0 & \\
\hline
 & 1
\end{array}$$

Dieses entspricht der Formel:

$$y_{j+1} = y_j + h_j f\left(t_j, y_j\right).$$

Algorithmisch interpretiert ergibt diese Formel, dass nur ein Schritt mit gegebener Schrittweite in Richtung des aktuellen Anstiegs, der am Punkt y_j besteht, gemacht wird.

Modifiziertes Verfahren von Euler

Ein einfaches Runge-Kutta-Verfahren der Ordnung 2 stellt das modifizierte Euler-Verfahren mit kommendem Butcher-Diagramm dar.

0		
$\frac{1}{2}$	$\frac{1}{2}$	
	0	1

Dieses entspricht der Formel:

$$y_{j+1} = y_j + h_j f\left(t_j + \frac{1}{2}h_j, y_j + \frac{1}{2}h_j f\left(t_j, y_j\right)\right).$$

Klassisches Runge-Kutta-Verfahren 4. Ordnung

Ein bekanntes Runge-Kutta-Verfahren der Ordnung 4 ist das klassische Runge-Kutta-Verfahren mit dem folgenden Butcher-Diagramm:

0				
$\frac{1}{2}$	$\frac{1}{2}$			
$\frac{1}{2}$	0	$\frac{1}{2}$		
1	0	0	1	
	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{6}$

Dieses entspricht der Formel:

$$y_{j+1} = y_j + h_j \left[\frac{1}{6}k_1\left(t_j, y_j\right) + \frac{1}{3}k_2\left(t_j, y_j\right) + \frac{1}{3}k_3\left(t_j, y_j\right) + \frac{1}{6}k_4\left(t_j, y_j\right)\right]$$

mit

$$\begin{aligned}
k_1\left(t_j, y_j\right) &= f\left(t_j, y_j\right) \\
k_2\left(t_j, y_j\right) &= f\left(t_j + \frac{1}{2}h_j, y_j + \frac{1}{2}h_j k_1\left(t_j, y_j\right)\right) \\
k_3\left(t_j, y_j\right) &= f\left(t_j + \frac{1}{2}h_j, y_j + \frac{1}{2}h_j k_2\left(t_j, y_j\right)\right) \\
k_4\left(t_j, y_j\right) &= f\left(t_j + h_j, y_j + h_j k_3\left(t_j, y_j\right)\right).
\end{aligned}$$

Runge-Kutta-Verfahren 4. Ordnung von Kuntzmann

Weiterhin wird die optimale Formel von Kuntzmann in Lehrbüchern gern zitiert. Diese hat das Butcher-Diagramm:

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{2}{5} & \frac{2}{5} & & & \\ \frac{3}{5} & -\frac{3}{20} & \frac{3}{4} & & \\ 1 & \frac{19}{44} & -\frac{15}{44} & \frac{40}{44} & \\ \hline & \frac{55}{360} & \frac{125}{360} & \frac{125}{360} & \frac{55}{360} \end{array}$$

Dieses entspricht der Formel:

$$y_{j+1} = y_j + h_j \left[\frac{55}{360} k_1 \left(t_j, y_j\right) + \frac{125}{360} k_2 \left(t_j, y_j\right) + \frac{125}{360} k_3 \left(t_j, y_j\right) + \frac{55}{360} k_4 \left(t_j, y_j\right) \right]$$

mit

$$\begin{aligned} k_1 \left(t_j, y_j\right) &= f \left(t_j, y_j\right) \\ k_2 \left(t_j, y_j\right) &= f \left(t_j + \frac{2}{5} h_j, y_j + \frac{2}{5} h_j k_1 \left(t_j, y_j\right) \right) \\ k_3 \left(t_j, y_j\right) &= f \left(t_j + \frac{3}{5} h_j, y_j - \frac{3}{20} h_j k_1 \left(t_j, y_j\right) + \frac{3}{4} h_j k_2 \left(t_j, y_j\right) \right) \\ k_4 \left(t_j, y_j\right) &= f \left(t_j + h_j, y_j + \frac{19}{44} h_j k_1 \left(t_j, y_j\right) - \frac{15}{44} h_j k_2 \left(t_j, y_j\right) + \frac{40}{44} h_j k_3 \left(t_j, y_j\right) \right). \end{aligned}$$

3.3.2 Explizite m-stufige Runge-Kutta-Verfahren mit Schrittweitensteuerung

Um bessere Ergebnisse in Bezug auf die optimale Schrittweite und das damit verbundene Zeitverhalten des Verfahrens zu erhalten, ist es sinnvoll, eine automatische Schrittweitensteuerung zu integrieren. Dafür muss ein geeigneter Weg zur Bestimmung der optimalen Schrittweite gefunden werden. Ein Ansatz für diese Bestimmung ist die Berechnung zweier expliziter Runge-Kutta-Verfahren unterschiedlicher Ordnung mit geeignet ausgewählten Koeffizienten. Üblicherweise werden dazu Verfahren mit aufeinander folgender Ordnung p und $p+1$ verwendet. Nach Berechnung eines Schrittes mit beiden Verfahren, wird die Differenz zwischen den beiden erhaltenen Werten berechnet. Diese Differenz, auch Fehler genannt, darf eine bestimmte obere Schranke nicht überschreiten. Übersteigt der Fehler die obere Schranke, wird die Schrittweite reduziert, unterschreitet er eine untere Schranke, kann die Schrittweite erhöht werden. Die Wahl der oberen und unteren Schranke richtet sich nach der Anwendung und ihren Genauigkeitsansprüchen sowie nach dem benötigten Rechenaufwand. Bei den Runge-Kutta-Verfahren werden häufig Verfahren der Ordnung 4 mit Verfahren der Ordnung 5 kombiniert. Für das entwickelte Werkzeug wurden zwei Verfahren der Ordnungen 4 und 5 ausgewählt und implementiert. Dies sind das eingebettete Runge-Kutta-Fehlberg-4(5)-Verfahren und das Verfahren von Dormand-Prince-5(4), deren Butcher-Diagramme auf den kommenden Seiten zu sehen sind. Letzteres wird auch als DOPRI5 bezeichnet und laut [Gan05] von Matlab für seinen Solver `ode45` verwendet.

Das allgemeine Vorgehen lässt sich wie folgt algorithmisch beschreiben:

Gegeben:
t_0, y_0, a
Anfangsschrittweite $h \leq a$
Toleranz τol
Intervall $[t_0, t_0 + a]$
2 Inkrementfunktionen ϕ_p, ϕ_{p+1}, zu Verfahren der Ordnung p und $p+1$

$i := 0$
$t := t_0$ - h
while $t + h < t_0 + a$ do
  $t_i := t + h$
  berechne $\phi_1 := \phi_p(t_i, y_i, h, f)$
  berechne $\phi_2 := \phi_{p+1}(t_i, y_i, h, f)$
  berechne $\tau := | \phi_2 - \phi_1 |$
  if $\tau < \tau ol$ then
    if $\tau > \frac{\tau ol}{100}$ then
      $y_{i+1} := y_i + h\phi_1$
      $t := t_i$
      $i := i + 1$
    else
      $h = 2h$
    endif
  else
    $h = \frac{h}{2}$
  endif
endwhile

$h := t_0 + a$ - t
$y_{i+1} := y_i + h\phi_p(t_i, y_i, h, f)$

Eingebettetes Runge-Kutta-Fehlberg-4(5)-Verfahren

0							
$\frac{1}{4}$	$\frac{1}{4}$						
$\frac{3}{8}$	$\frac{3}{32}$	$\frac{9}{32}$					
$\frac{12}{13}$	$\frac{1932}{2197}$	$-\frac{7200}{2197}$	$\frac{7296}{2197}$				
1	$\frac{439}{216}$	-8	$\frac{3680}{513}$	$-\frac{845}{4104}$			
$\frac{1}{2}$	$-\frac{8}{27}$	2	$-\frac{3544}{2565}$	$\frac{1859}{4104}$	$-\frac{11}{40}$		
y	$\frac{25}{216}$	0	$\frac{1408}{2565}$	$\frac{2197}{4104}$	$-\frac{1}{5}$	0	Ordnung 4
$\hat{y}$	$\frac{16}{153}$	0	$\frac{6656}{12825}$	$\frac{28561}{56430}$	$-\frac{9}{50}$	$\frac{2}{55}$	Ordnung 5

Verfahren von Dormand-Prince-5(4) (DOPRI5; Matlab-ODE45)

0								
$\frac{1}{5}$	$\frac{1}{5}$							
$\frac{3}{10}$	$\frac{3}{40}$	$\frac{9}{40}$						
$\frac{4}{5}$	$\frac{44}{45}$	$-\frac{56}{15}$	$\frac{32}{9}$					
$\frac{8}{9}$	$\frac{19372}{6561}$	$-\frac{25360}{2187}$	$\frac{64448}{6561}$	$-\frac{212}{729}$				
1	$\frac{9017}{3168}$	$-\frac{355}{33}$	$\frac{46732}{5247}$	$\frac{49}{176}$	$-\frac{5103}{18656}$			
1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$		
y_1	$\frac{35}{384}$	0	$\frac{500}{1113}$	$\frac{125}{192}$	$-\frac{2187}{6784}$	$\frac{11}{84}$	0	Ordnung 5
$\hat{y}_1$	$\frac{5179}{57600}$	0	$\frac{7571}{16695}$	$\frac{393}{640}$	$-\frac{92097}{339200}$	$\frac{187}{2100}$	$\frac{1}{40}$	Ordnung 6

3.4 Rosenbrock-Verfahren

Die Rosenbrock-Verfahren können aus den impliziten Runge-Kutta-Verfahren abgeleitet werden, gehören aber nicht mehr zu ihrer Klasse. Die Vereinfachung im Vergleich zu den sehr komplexen impliziten Runge-Kutta-Verfahren besteht darin, die Koeffizienten so zu wählen, dass pro Stufe genau ein Iterationsschritt mit einem vereinfachten Newton-Verfahren berechnet werden kann.

1 **Gegeben:**
2 t_0, y_0; a
3 Schrittweite h
4 Anzahl Stufen s
5
6 berechne $J := f_y(y_i)$ (nicht notwendig in jedem Zeitschritt)
7 zerlege $I - ahJ = LR$ mit dem Gauß-Algorithmus
8 for $j = 1, \ldots, s$ do
9 löse $(I - ahJ)\, k_j = f\left(y_i + h\sum_{\nu=1}^{j-1} a_{j\nu} k_\nu\right) - hJ\sum_{\nu=1}^{j-1} d_{j\nu} k_\nu$
10 endfor
11
12 $y_{i+1} := y_i + h\sum_{j=1}^{s} b_j k_j$

Die im Werkzeug implementierten Rosenbrock-Verfahren besitzen weitere Einschränkungen speziell bei der Wahl einer Approximation T der Jacobi-Matrix J. Dadurch lässt sich die Jacobi-Matrix zeitlich effizienter approximieren. Diese speziellen Rosenbrock-Verfahren wurden 1977 von Gerhard Wanner in [Wan77] eingeführt. Die Approximation macht die Einführung zusätzlicher Koeffizienten notwendig.

Für die Implementierung im entwickelten Werkzeug wurde eine Pascal-Implementierung von [Mor] in C++ übertragen und integriert. Die Pascal-Implementierung basiert ihrerseits auf einer Fortran77-Implementierung von den Entwicklern dieser Rosenbrock-Verfahren [Hai]. Im folgenden werden die Koeffizienten der sechs implementierten Rosenbrock-Methoden angegeben.

Rosenbrock-Methode von Shampine

$$
\begin{aligned}
a_{21} &= 2, \\
a_{31} &= \frac{48}{25}, \\
a_{32} &= \frac{6}{25}, \\
c_{21} &= -8, \\
c_{31} &= \frac{372}{25}, \\
c_{32} &= \frac{12}{5}, \\
c_{41} &= -\frac{112}{125}, \\
c_{42} &= -\frac{54}{125}, \\
c_{43} &= -\frac{2}{5}, \\
b_1 &= \frac{19}{9}, \\
b_2 &= \frac{1}{2}, \\
b_3 &= \frac{25}{108}, \\
b_4 &= \frac{125}{108}, \\
e_1 &= \frac{17}{54}, \\
e_2 &= \frac{7}{36}, \\
e_3 &= 0.0, \\
e_4 &= \frac{125}{108}, \\
\gamma &= 0.5, \\
c_2 &= 1, \\
c_3 &= 0.6, \\
d_1 &= 0.5, \\
d_2 &= -1.5, \\
d_3 &= 2.42, \\
d_4 &= 0.116
\end{aligned}
$$

Rosenbrock-Methode GRK4T von Kaps-Rentrop

$$
\begin{aligned}
a_{21} &= 2, \\
a_{31} &= 4.524708207373116, \\
a_{32} &= 4.163528788597648, \\
c_{21} &= -5.071675338776316, \\
c_{31} &= 6.020152728650786, \\
c_{32} &= 0.1597506846727117, \\
c_{41} &= -1.856343618686113, \\
c_{42} &= -8.505380858179826, \\
c_{43} &= -2.084075136023187, \\
b_1 &= 3.957503746640777, \\
b_2 &= 4.624892388363313, \\
b_3 &= 0.6174772638750108, \\
b_4 &= 1.282612945269037, \\
e_1 &= 2.302155402932996, \\
e_2 &= 3.073634485392623, \\
e_3 &= -0.8732808018045032, \\
e_4 &= -1.282612945269037, \\
\gamma &= 0.231, \\
c_2 &= 0.462, \\
c_3 &= 0.8802083333333334, \\
d_1 &= 0.231, \\
d_2 &= -3.962966775244303, \\
d_3 &= 0.5507789395789127, \\
d_4 &= -0.05535098457052764
\end{aligned}
$$

Rosenbrock-Methode GRK4A von Kaps-Rentrop

$$
\begin{aligned}
a_{21} &= 1.108860759493671, & b_4 &= 0.6329113924050632,\\
a_{31} &= 2.377085261983360, & e_1 &= 0.04831870177201765,\\
a_{32} &= 0.1850114988899692, & e_2 &= -0.6471108651049505,\\
c_{21} &= -4.920188402397641, & e_3 &= 0.218687666050024,\\
c_{31} &= 1.055588686048583, & e_4 &= -0.6329113924050632,\\
c_{32} &= 3.351817267668938, & \gamma &= 0.395,\\
c_{41} &= 3.846869007049313, & c_2 &= 0.438,\\
c_{42} &= 3.42710924126818, & c_3 &= 0.87,\\
c_{43} &= -2.162408848753263, & d_1 &= 0.395,\\
b_1 &= 1.84568324040584, & d_2 &= -0.372672395484092,\\
b_2 &= 0.1369796894360503, & d_3 &= 0.06629196544571492,\\
b_3 &= 0.7129097783291559, & d_4 &= 0.4340946962568634
\end{aligned}
$$

Rosenbrock-Methode von Van Veldhuizen [$\gamma = \frac{1}{2}$]

$$
\begin{aligned}
a_{21} &= 2, & b_4 &= 1.333333333333333,\\
a_{31} &= 1.75, & e_1 &= -0.3333333333333333,\\
a_{32} &= 0.25, & e_2 &= -0.3333333333333333,\\
c_{21} &= -8, & e_3 &= -0,\\
c_{31} &= -8, & e_4 &= -1.333333333333333,\\
c_{32} &= -1, & \gamma &= 0.5,\\
c_{41} &= 0.5, & c_2 &= 1,\\
c_{42} &= -0.5, & c_3 &= 0.5,\\
c_{43} &= 2, & d_1 &= 0.5,\\
b_1 &= 1.333333333333333, & d_2 &= -1.5,\\
b_2 &= 0.6666666666666667, & d_3 &= -0.75,\\
b_3 &= -1.333333333333333, & d_4 &= 0.25
\end{aligned}
$$

Rosenbrock-Methode von Van Veldhuizen [D-stabil]

$$
\begin{aligned}
a_{21} &= 2, & b_4 &= 1.355958941201148,\\
a_{31} &= 4.812234362695436, & e_1 &= 2.175672787531755,\\
a_{32} &= 4.578146956747842, & e_2 &= 2.950911222575741,\\
c_{21} &= -5.333333333333331, & e_3 &= -0.785974454488743,\\
c_{31} &= 6.100529678848254, & e_4 &= -1.355958941201148,\\
c_{32} &= 1.804736797378427, & \gamma &= 0.2257081148225682,\\
c_{41} &= -2.540515456634749, & c_2 &= 0.4514162296451364,\\
c_{42} &= -9.443746328915205, & c_3 &= 0.8755928946018455,\\
c_{43} &= -1.988471753215993, & d_1 &= 0.2257081148225682,\\
b_1 &= 4.289339254654537, & d_2 &= -0.04599403502680582,\\
b_2 &= 5.036098482851414, & d_3 &= 0.5177590504944076,\\
b_3 &= 0.6085736420673917, & d_4 &= -0.03805623938054428,
\end{aligned}
$$

Eine L-stabile Rosenbrock-Methode

$$
\begin{aligned}
a_{21} &= 2, & b_4 &= 1.093502252409163,\\
a_{31} &= 1.867943637803922, & e_1 &= -0.2815431932141155,\\
a_{32} &= 0.2344449711399156, & e_2 &= -0.0727619912493892,\\
c_{21} &= -7.13761503641231, & e_3 &= -0.1082196201495311,\\
c_{31} &= 2.580708087951457, & e_4 &= -0.1.093502252409163,\\
c_{32} &= 0.6515950076447975, & \gamma &= 0.57282,\\
c_{41} &= -2.137148994382534, & c_2 &= 0.1.14564,\\
c_{42} &= -0.3214669691237626, & c_3 &= 0.65521686381559,\\
c_{43} &= -0.6949742501781779, & d_1 &= 0.57282,\\
b_1 &= 2.255570073418735, & d_2 &= -1.769193891319233,\\
b_2 &= 0.2870493262186792, & d_3 &= 0.7592633437920482,\\
b_3 &= 0.4353179431840180, & d_4 &= -0.104902108710045,
\end{aligned}
$$

Kapitel 4

Implementierung

Im folgenden Kapitel werden bestehende Programmpakete analysiert und danach die getroffenen Entwurfsentscheidungen für das entwickelte Werkzeug erläutert.

4.1 Analyse bestehender Programmpakete

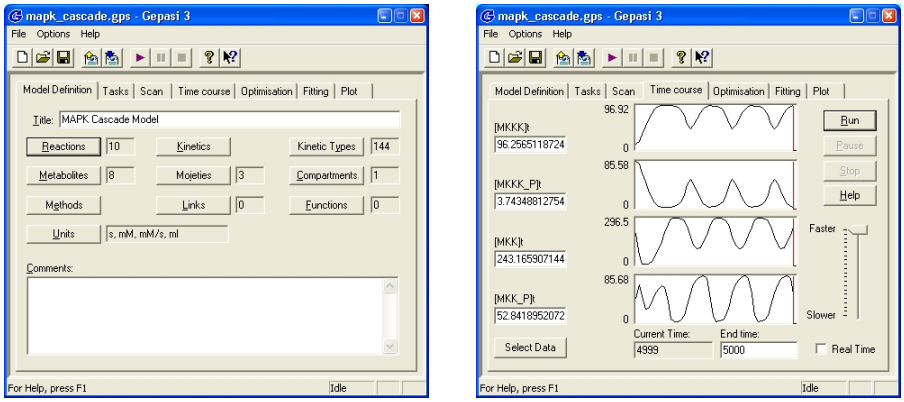
Zu Beginn der Ausführungen über die konkrete Umsetzung der kontinuierlichen Petrinetze in einem Werkzeug, wird kurz auf einige bestehende Werkzeuge eingegangen und es werden ihre jeweiligen Vorzüge und Nachteile aufgezeigt. Dabei wird hervorgehoben, welche Funktionen in das entwickelte Werkzeug übernommen wurden.

4.1.1 Gepasi / Copasi

Die Softwarepakete Copasi [MK] und das Vorgängerprojekt Gepasi [Men] sind beliebte Simulationswerkzeuge zur Simulation von biochemischen Prozessen. Sie bieten umfangreiche Möglichkeiten, bei denen ausgiebig von gewöhnlichen Differentialgleichungen zur Beschreibung der Prozessabläufe Gebrauch gemacht wird. Die beiden untersuchten Werkzeuge besitzen eine umfangreiche Auswahl an numerischen Lösungsverfahren, welche vom Benutzer allerdings nur beschränkt beeinflusst werden können. Die konkret verwendeten Lösungsverfahren werden automatisch anhand numerischer Optimalitätskriterien ausgewählt. Konkrete Informationen über vorhandene Lösungsverfahren in Gepasi wurden bereits 1998 von Pedro Mendes in [MK98] veröffentlicht.

Beide Werkzeuge besitzen eine, in Abbildung 4.1 dargestellte, graphische Benutzeroberfläche zur textuellen Eingabe der Differentialgleichungen und zur Simulation selbiger. Zur Definition der Differentialgleichungen stehen viele, in biochemischen Anwendungen gebräuchliche Gleichungsmuster zur Verfügung. Den in diesen Mustern enthaltenen Variablen müssen nur die konkreten Werte oder Variablen der modellierten Umgebung zugewiesen werden. Dabei ist es möglich, zusätzlich zu den existierenden Mustern, eigene Muster zu erstellen und zu verwenden. Das ältere Werkzeug Gepasi 3.30, welches seit September 2002 unverändert vorliegt, speichert die Simulationsergebnisse in einer kommaseparierten Textdatei und besitzt die Möglichkeit, während der Simulation bis zu vier Variablen in separaten Fenstern mit automatischer Normierung der y-Achse graphisch darzustellen. Ferner existiert eine Schnittstelle zum Graphikwerkzeug **gnuplot**, welches

eine komfortable Auswertung der Daten ermöglicht. Ein Datenaustausch mit anderen biochemischen Werkzeugen ist über einen Import bzw. Export in die Systems Biology Markup Language (SBML) Version 1 möglich.



(a) Modelleigenschaften (b) Graphische Ausgabe

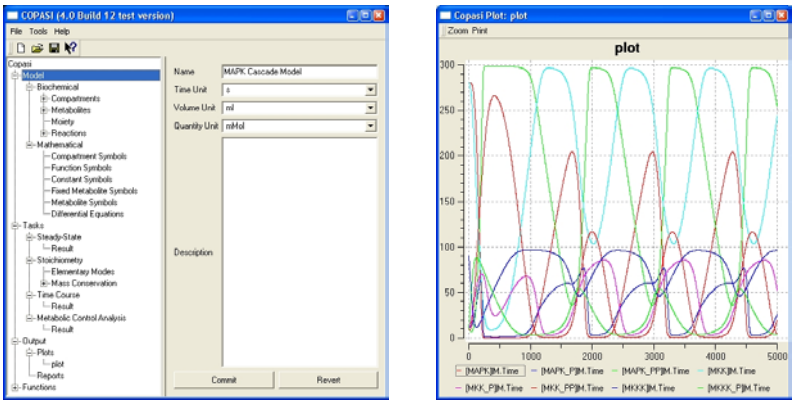
Abbildung 4.1: Gepasi

In Zusammenarbeit mit der Arbeitsgruppe von Ursula Kummer am European Media Lab Research Heidelberg entwickelte die Gruppe um Pedro Mendes am Virginia Bioinformatics Institute die Benutzeroberfläche komplett neu und es entstand das Werkzeug Copasi. Die Oberfläche kommt im Gegensatz zu Gepasi nahezu komplett ohne Dialoge aus. Abbildung 4.2 zeigt exemplarisch die Modelleigenschaften und die graphische Ausgabe. Von den grundlegenden Möglichkeiten ist es aber weiterhin mit Gepasi vergleichbar. Wesentlich verbessert wurde die graphische Ausgabe der Simulationsergebnisse. Ferner wurde verändert, dass die Simulationsergebnisse intern im Programm gespeichert und nach der Simulation in einer Datei gesichert werden können. Dieses garantiert, dass ältere Simulationsergebnisse nicht mehr überschrieben werden. In den letzten Jahren wurden weitere Lösungsverfahren integriert, welche vom Programm eigenständig ausgewählt werden. Der Nutzer hat nur wenige Möglichkeiten, die vorhandenen Lösungsverfahren auszuwählen und zu parametrisieren. Neu hinzugekommen ist unter anderem eine Ausgabemöglichkeit für die mathematischen Parameter, Funktionen und Werte. Außerdem wird ein Import bzw. Export in SBML Version 2 unterstützt.

Zusammenfassend kann eingeschätzt werden, dass die beiden Werkzeuge für die Simulation von biochemischen Prozessen sehr gut geeignet sind. Allerdings sind einige Schwächen in der Verständlichkeit der verwendeten Differentialgleichungen zu sehen, weil kaum eine Strukturierung möglich ist und diese mit beiden Werkzeugen nicht graphisch unterstützt und aufbereitet werden kann. Sehr hilfreich ist das Angebot an Gleichungsmustern, so dass eine vergleichbare Funktion im Werkzeug Snoopy integriert wurde.

4.1.2 Visual Object Net / Genomic Object Net / Cell Illustrator

Das dritte untersuchte Programmpaket **Visual Object Net Version 2.0a** wurde am Institut für Automatisierungs- und Systemtechnik der TU Ilmenau im Rahmen eines DFG Projektes zur Automatisierungstechnik entwickelt. Dabei handelt es sich um ein Werkzeug zur Modellierung und Simulation hybrider Petrinetze. Bei der Simulation der modellierten Probleme werden keinerlei



(a) Modelleigenschaften (b) Graphische Ausgabe

Abbildung 4.2: Copasi

Konsistenzprüfungen bezüglich der Verwendung von Plätzen, die nicht Vorplätze der betreffenden Transition sind, gemacht, sondern es können beliebige Differentialgleichungen verwendet werden. Ferner ist es nicht gelungen, Informationen über die zugrunde liegenden numerischen Lösungsverfahren und deren Implementierung zu bekommen. Dies könnte dadurch zu erklären sein, dass das Programmpaket nicht mehr weiterentwickelt wird.

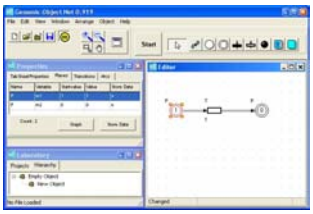
Auf die Arbeiten zu Visual Object Net hat eine japanische Forschungsgruppe um Hiroshi Matsuno an der Yamaguchi Universität im Projekt **Genomic Object Net** [MMN] aufgebaut und ein zur Simulation biochemischer Prozesse einsetzbares Werkzeug entwickelt. Die vorliegende Version ist im Aussehen und Verhalten nahezu identisch zu Visual Object Net. Weiterentwickelte Versionen sind in das kommerzielle Werkzeug **Cell Illustrator** eingeflossen.

Zur graphischen Auswertung der Daten stehen für jeden Platz eigene Graphen zur Verfügung. Es sind automatische und manuelle Änderungen an der Skalierung und Lage der Achsen möglich. Weiterhin kann die Ausgabe graphisch und textuell exportiert werden. Die Abbildung 4.3 zeigt einige Ansichten des Programmes Genomic Object Net.

Nachteilig ist die Beschränkung auf einen überwachten Platz pro graphischer Ausgabe. Mehrere Plätze werden in jeweils eigenen Fenstern mit eigener Skalierung dargestellt. Dadurch ist ein direkter Vergleich der Markierungsänderung mehrerer Plätze schwer möglich. Außerdem gibt es keine Möglichkeit, die Simulation zu konfigurieren. Ferner wird das Petrinetzmodell vor der Simulation keiner Konsistenzprüfung unterzogen. Dadurch sind beliebige Geschwindigkeitsfunktionen möglich und dabei können Markierungswerte von Plätzen verwendet werden, die in der Netzstruktur nicht mit der betreffenden Transition verbunden sind. Aufgenommen wurde in das Werkzeug Snoopy das Design für kontinuierliche Knoten mit einer Umrandung in Form einer Doppellinie.

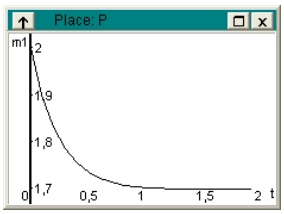
4.1.3 ModelKinetix ModelMaker

ModelKinetix ModelMaker, ein kommerzielles Softwarepaket, bietet viele Möglichkeiten biochemische Anwendungen zu modellieren. Es unterstützt direkt Objekte wie *Compartment*, *Variable*, *Define*, *Flow* und *Influence*. Diese drei Knoten und zwei Kantentypen sind entsprechend semantisch vorbelegt. Jedem Knoten lassen sich Differentialgleichungen und den Compartment auch Werte zuweisen. Mit den angebotenen Elementen lassen sich in ModelMaker auch konti-



(a) Hauptfenster

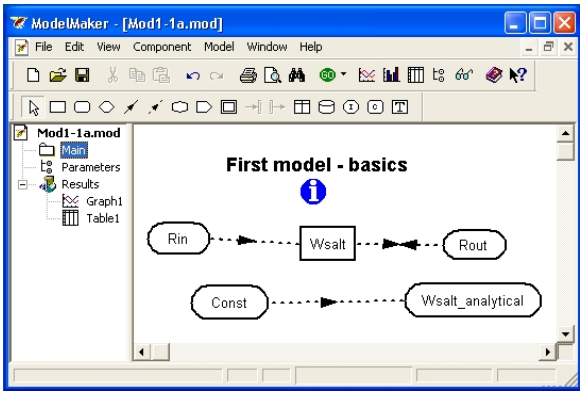
(b) Simulation



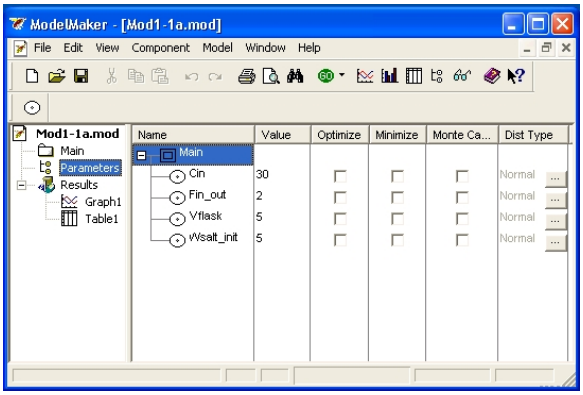
(c) Graphische Ausgabe

Abbildung 4.3: Genomic Object Net

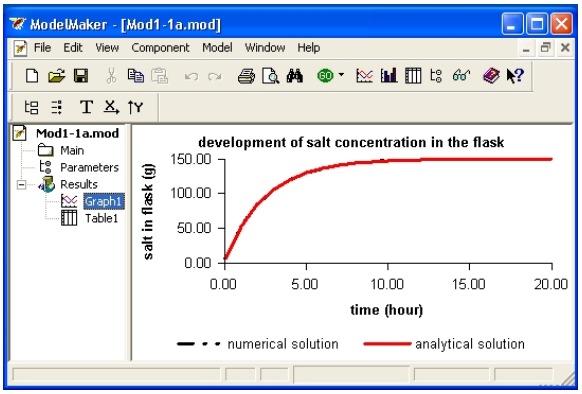
nuierliche Petrinetze modellieren. Weiterhin werden die Nutzung von Parametern, zur einfachen Manipulation des Modellverhaltens, und umfangreiche Auswertungsmöglichkeiten angeboten. So können Simulationsergebnisse in Diagrammen und in Tabellen ausgegeben werden. Das Konzept der Parameter hat positiv überzeugt und wurde in Form von kontinuierlichen Parametern im Werkzeug Snoopy integriert. Wiederum werden in Abbildung 4.4 einige Ansichten des beschriebenen Werkzeuges abgebildet.



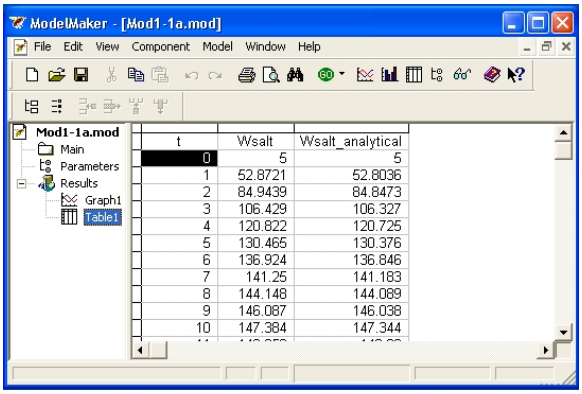
(a) Hauptfenster



(b) Parameter



(c) Diagramm



(d) Tabelle

Abbildung 4.4: ModelKinetix ModelMaker

4.2 Entwurfsentscheidungen für Snoopy

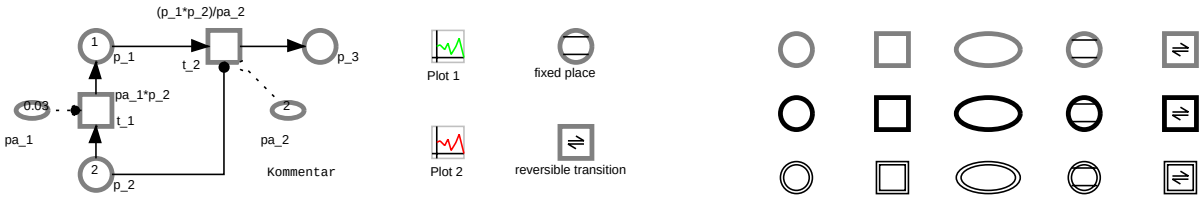
In diesem Abschnitt werden die getroffenen Entwurfsentscheidungen für das Werkzeug Snoopy dargelegt. Dabei wird kurz auf die Hintergründe dieser Entscheidung eingegangen.

4.2.1 Darstellung der kontinuierlichen Petrinetze

Die kontinuierlichen Petrinetze werden in Snoopy in enger Anlehnung an die diskreten Petrinetze als Kreise und Rechtecke dargestellt. Zur Unterscheidung der beiden Netzklassen und um eine Erweiterung zu hybriden Petrinetzen zu ermöglichen, wurden die kontinuierlichen Knoten mit einem veränderten Design versehen. Dafür existieren drei Varianten, aus denen für jeden Knoten ausgewählt werden kann: eine dicke graue Linie, eine dicke schwarze Linie und zwei verschachtelte schwarze Kreise bzw. Rechtecke. Dabei wird die erstgenannte Variante als Standardvariante verwendet. Das Symbol der Ellipse wurde für die Parameter gewählt.

Die drei verschiedenen Kantenarten werden als normale schwarze Linie mit unterschiedlichen Enden dargestellt. Dabei wird eine normale Kante mit einer Pfeilspitze gekennzeichnet. Lese- und Inhibitorkanten besitzen einen ausgefüllten bzw. unausgefüllten Kreis am Ende der Kante. Die beiden Kantenarten zur Verbindung von Parametern und Transitionen sind analog zur Lesekante bzw. Inhibitorkante mit einem ausgefüllten bzw. unausgefüllten Kreis am Ende gezeichnet. Die Linien bei den Parameterkanten werden zur Unterscheidung gestrichelt dargestellt.

Die folgende Abbildung 4.5 (a) zeigt ein einfaches Beispiel und enthält alle verfügbaren Netzelemente in der Standardvariante. Außerdem werden bei den Transitionen neben dem Namen auch die Formeln der Schaltgeschwindigkeitsfunktion angezeigt. Gesondert hingewiesen sei auf das Symbol für die graphische Ausgabe, welche im Abschnitt 4.2.5 eingeführt wird. Diese graphische Ausgabe wird mit einem stilistischen Funktionsgraphen in einem Koordinatensystem dargestellt. Abbildung 4.5 (b) präsentiert die verfügbaren Knotenarten in den drei möglichen Designs.



(a) Standarddesign (b) Designübersicht

Abbildung 4.5: Snoopy: Darstellung der kontinuierlichen Petrinetzelemente

4.2.2 Verwendung von Gleichungsmustern

In den Softwarepaketen Gepasi und Copasi hat die Möglichkeit zur Verwendung von Gleichungsmustern, durch ihren Nutzen für die effiziente Arbeit mit den Werkzeugen und vor allem die Vermeidung von Fehlern bei der Modellierung von Anwendungen, überzeugt. In Snoopy wurde deswegen eine Möglichkeit zur Auswahl von bestehenden Gleichungsmustern geschaffen. Dabei

wurde darauf geachtet, dass vom Entwickler des Werkzeuges problemlos weitere Gleichungsmuster hinzugefügt werden können. Bisher stehen, zusätzlich zur freien Eingabe, zwei häufig angewendete Gleichungsmuster zur Verfügung:

- Michaelis-Menten-Gleichung: $\frac{v_0 * M}{M + K_m}$
- Massengleichung: $k * M$

4.2.3 Übertragung der kontinuierlichen Petrinetze in gewöhnliche Differentialgleichungen

Die Übertragung der kontinuierlichen Petrinetze in gewöhnliche Differentialgleichungen gestaltet sich relativ einfach. Wie in Definition 2.4.2 definiert, wird die Differentialgleichung für die Markenänderung eines Platzes über die Zeit aus der Summe der Schaltgeschwindigkeiten aller Vortransitionen abzüglich der Summe der Schaltgeschwindigkeiten aller Nachtransitionen gebildet. Dabei entstehen für das einfache Beispielnetz aus Abbildung 4.5 (a) folgende Differentialgleichungen:

$$\begin{aligned}
\frac{d\mathrm{p}_1}{d\mathrm{t}} &= g(\mathrm{t}_1, \mathrm{p}_1)\, v(\mathrm{t}_1) - g(\mathrm{p}_1, \mathrm{t}_2)\, v(\mathrm{t}_2) &&= \mathrm{pa}_1 * \mathrm{p}_2 - \frac{\mathrm{p}_1 * \mathrm{p}_2}{\mathrm{pa}_2} \\
\frac{d\mathrm{p}_2}{d\mathrm{t}} &= -g(\mathrm{p}_2, \mathrm{t}_1)\, v(\mathrm{t}_1) &&= -\mathrm{pa}_1 * \mathrm{p}_2 \\
\frac{d\mathrm{p}_3}{d\mathrm{t}} &= g(\mathrm{t}_2, \mathrm{p}_3)\, v(\mathrm{t}_2) &&= \frac{\mathrm{p}_1 * \mathrm{p}_2}{\mathrm{pa}_2}
\end{aligned}$$

Bei der Generierung der gewöhnlichen Differentialgleichungen wird überprüft, ob zu jedem in der Gleichung verwendeten Variablennamen ein Vorplatz mit gleichem Namen oder ein Parameter existiert. Dabei müssen Parameter, wie bereits in Abschnitt 2.3.2 definiert, nicht über eine Kante mit der Transition verbunden sein. In diesem Fall wird der erste kontinuierliche Parameter mit entsprechendem Namen verwendet.

Falls zu einem Variablennamen kein über eine eingehende Kante verbundener Platz und kein Parameter gleichen Namens existiert, wird eine Fehlermeldung ausgegeben und die Simulation abgebrochen. Die Fehlermeldung enthält die Namen der betreffenden Transition, des betreffenden Variablenbezeichners und die, der Transition zugeordnete, Differentialgleichung.

4.2.4 Konfigurationsmöglichkeiten der Simulationen

Bei der Simulation der modellierten kontinuierlichen Petrinetze bestehen mehrere Konfigurationsmöglichkeiten. Es können folgende Einstellungen ausgewählt werden:

Parametersatz gibt den zu verwendenden Parametersatz an.

initiale Markierung gibt an, welcher Satz von Markierungswerten als initiale Markierung verwendet wird

Schrittweite gibt an, in welchen Abständen berechnete Werte ausgegeben werden. Bei den später aufgeführten Verfahren mit fester Schrittweite wird diese Schrittweite als Berechnungsgrundlage verwendet.

Rundenanzahl bestimmt, wie viele Runden simuliert werden sollen und wann die Simulation angehalten werden soll. Dabei wird immer bis zum nächsten ganzzahligen Vielfachen dieser Rundenanzahl berechnet.

Toleranz ist nur für die Verfahren mit variabler Schrittweitensteuerung relevant. Sie wird bei diesen Verfahren verwendet, um die geeignete interne Schrittweite für die Berechnung der Runden zu ermitteln.

Verfahren bestimmt, das zur Simulation zu verwendende numerische Verfahren. Es stehen die in Kapitel 3 eingeführten numerischen Berechnungsverfahren zur Verfügung.

Ausgabekanal legt fest, welche der im Abschnitt 4.2.5 beschriebenen Ausgabealternativen verwendet wird.

Weitere Informationen über die Konfigurationsmöglichkeiten sind im Anhang A.2 zu finden.

4.2.5 Ergebnisdarstellung

Die berechneten Ergebnisse können auf zwei Arten weiterverarbeitet werden. Zum einen wird ein Export in eine Textdatei im CSV-Format angeboten. Alternativ können die Daten im internen Speicher des Werkzeuges gespeichert und graphisch dargestellt werden. Dafür werden bei der Modellierung des Petrinetzes sogenannte **Plots** angelegt, in denen festgelegt wird, welche Plätze bzw. Markierungen selbiger dargestellt werden sollen. Die Plots bauen sich automatisch während der Simulation auf und werden nach jeder Runde aktualisiert. Für die graphische Ausgabe kann dabei jeder Plot einzeln aktiviert und deaktiviert werden. Das zu dem jeweiligen Plot zugehörige Symbol im Petrinetzmodell erscheint mit grünem bzw. rotem Funktionsgraphen. Nach Abschluss der Simulation können die im Speicher gehaltenen Werte auch als Textdatei im bereits erwähnten CSV-Format gesichert werden.

4.3 Validierung

Die in Snoopy implementierten Verfahren wurden gegen die Ergebnisse des Softwarepaketes Copasi getestet. Die erzielten Ergebnisse wurden verglichen und analysiert. Genauere Daten finden sich im Anhang B ab Seite 61. Die Validierung hat ergeben, dass die Simulationsergebnisse mit höherer Genauigkeit als vom Referenzwerkzeug Copasi ausgegeben werden. Dabei sind die absoluten Fehler kleiner als die Ausgabegenauigkeit von Copasi. Beim Zeitverhalten besteht noch Verbesserungsbedarf, weil Snoopy um den Faktor 10 bis 4000 langsamer rechnet. Dies ist damit zu erklären, dass Copasi wesentlich mehr numerische Verfahren implementiert hat. Nach [MK98] sind bereits in *Gepasi Version 3.20* 13 numerische Verfahren mit jeweils mehreren alternativen Koeffizienten integriert. Diese Vielfalt an numerischen Verfahren ist in Snoopy nicht verfügbar, womit die Unterschiede im Zeitverhalten erklärbar sind. Weiterhin haben Gepasi und Copasi eine langjährige Entwicklungszeit und vielfältige Möglichkeiten zur Optimierung durchlaufen, die dem in dieser Arbeit entwickelten Werkzeug fehlen.

Kapitel 5

Beispielanwendungen

Dieses Kapitel widmet sich einigen Beispielen für kontinuierliche Petrinetze. Dabei werden zuerst theoretische Beispiele zur Betrachtung linearer Differentialgleichungen und deren Übertragung in kontinuierliche Petrinetze vorgestellt. Im weiteren Verlauf folgen fünf praktische Beispiele aus der aktuellen Forschung an biochemischen Problemstellungen. Diese Anwendungsbeispiele wurden Artikeln der zitierten Autoren entnommen und in ein kontinuierliches Petrinetz überführt. Gleichzeitig dienten diese Anwendungen zur Validierung des entwickelten Werkzeuges.

5.1 Künstlich konstruierte Beispiele

In diesem Abschnitt sollen die Möglichkeiten der Analyse der Struktur kontinuierlicher Petrinetze durch Überführung in diskrete Petrinetze demonstriert werden. Dazu werden Beispiele für Petrinetze mit möglichen Kombinationen für die Eigenschaften *Beschränktheit*, *Lebendigkeit* und *Rücksetzbarkeit* erstellt, welche lineare Differentialgleichungssysteme erzeugen. Vorher werden einige Vorbetrachtungen zur Modellierung linearer Probleme in kontinuierlichen Petrinetzen getätigt. Anschließend wird aufgezeigt, welche Petrinetzstrukturen Linearität erhalten, und welche sie zerstören.

Bei den Beispielen werden jeweils die gewöhnlichen Differentialgleichungen in Gleichungs- und Matrixschreibweise, das kontinuierliche Petrinetz als Graph und der Ergebnisvektor des Petrinetz-Analyse-Werkzeuges **Integrated Net Analyzer INA** [Sta03] angegeben. Dazu wurden die kontinuierlichen Petrinetze in diskrete Petrinetze überführt und danach mit INA analysiert. Weiterhin werden in den Abbildungen 5.3 (b) bis (h) die Diagramme für die sieben Kombinationen gezeigt, zu denen lineare Beispiele existieren. Abbildung 5.3 (a) bildet die für die Kurvenverläufe verwendete Legende ab.

Zu jedem Beispiel wird zusätzlich kurz auf die Eigenschaften des jeweiligen diskreten Petrinetzes und ihre Wiederspiegelung im kontinuierlichen Verhalten des Beispieles eingegangen. Dabei werden teilweise Unterschiede zwischen den analysierten Eigenschaften des diskretisierten Modells und dem Verhalten im kontinuierlichen System auftreten. Diese Unterschiede resultieren aus dem großen Einfluss den die Wahl der konkreten Geschwindigkeitsfunktion auf das Verhalten des kontinuierlichen Petrinetzes hat. Andererseits sind zur Analyse des diskreten Petrinetzes ausschließlich die Struktur und die Startmarkierung ausschlaggebend.

5.1.1 Lineare Beispiele - Vorbetrachtungen

Wie im Vorspann angedeutet wurde, soll dieser Abschnitt aufzeigen, welche Strukturen in einem Petrinetz erlaubt sind, um lineare Differentialgleichungssysteme zu erhalten. Dies ist mathematisch besonders interessant, weil lineare Differentialgleichungssysteme gut analysierbar und berechenbar sind. Außerdem können lineare Systeme in Matrixschreibweise aufgeschrieben werden und es existieren Methoden zum Nachweis von Verhaltensweisen solcher Systeme. Gesondert verwiesen sei hierbei auf Verfahren zur Linearen Programmierung und zum Lösen von linearen Gleichungssystemen, wie dem Gauss-Algorithmus.

Für die Analyse erlaubter Strukturen im Petrinetz werden die möglichen Kombinationen von ein bis zwei mit einer Transition verbundenen Plätzen betrachtet, und unter Berücksichtigung aller eingehenden Kanten bei der Geschwindigkeitsfunktion die Linearität untersucht. Für den Erhalt der Linearität ist es wichtig, dass maximal ein Vorplatz pro Transition existiert, weil jeder weitere Vorplatz multiplikativ in die Gleichung zur Schaltgeschwindigkeit eingeht. Nachplätze dürfen allerdings beliebig viele auftauchen, weil sie keinen Einfluss auf die Geschwindigkeitsfunktion ausüben und die nach Definition 2.4.2 konstruierte Markierungsänderung für Plätze nur additive Anteile besitzt.

Erlaubte Strukturen

Im folgenden werden Strukturen dargestellt, welche die Linearität erhalten. Zur besseren Verdeutlichung der Linearität der gezeigten Beispiele werden die Differentialgleichungen in Gleichungs- und in Matrixschreibweise angegeben.

1. Dieses Beispiel verdeutlicht, dass Konfliktsituationen unproblematisch in linearen Gleichungen dargestellt werden können. Das zugehörige kontinuierliche Petrinetz ist in Abbildung 5.1 (a) abgebildet.

$$\mathrm{x}_1 = 1 \qquad \begin{aligned} \mathrm{c}_1 &= 0.3 \\ \mathrm{c}_2 &= 0.4 \end{aligned} \qquad \frac{d\mathrm{x}_1}{d\mathrm{t}} = -\mathrm{t}_1 - \mathrm{t}_2 \qquad \begin{aligned} \mathrm{t}_1 &= \mathrm{c}_1 * \mathrm{x}_1 \\ \mathrm{t}_2 &= \mathrm{c}_2 * \mathrm{x}_1 \end{aligned}$$

$$x_{i+1} = x_i + Bx_i \qquad x_0 = \begin{pmatrix} 1 \end{pmatrix} \qquad B = \begin{pmatrix} -0.3 - 0.4 \end{pmatrix}$$

2. Das zweite erlaubte Strukturbeispiel aus Abbildung 5.1 (b) zeigt eine Transition mit zwei Nachplätzen. Auch dies ist problemlos als lineares System darstellbar.

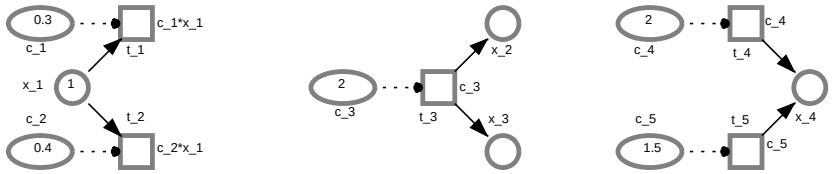
$$\begin{aligned} \mathrm{x}_2 &= 0 \\ \mathrm{x}_3 &= 0 \end{aligned} \qquad \mathrm{c}_3 = 2.0 \qquad \begin{aligned} \frac{d\mathrm{x}_2}{d\mathrm{t}} &= \mathrm{t}_3 \\ \frac{d\mathrm{x}_3}{d\mathrm{t}} &= \mathrm{t}_3 \end{aligned} \qquad \mathrm{t}_3 = \mathrm{c}_3$$

$$x_{i+1} = x_i + c \qquad x_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad c = \begin{pmatrix} 2.0 \end{pmatrix}$$

3. Analog dazu ist es möglich, einem Platz mehrere Vortransitionen zuzuweisen. Dieser Fall ist in Abbildung 5.1 (c) dargestellt.

$$x_4 = 0 \qquad \begin{aligned} c_4 &= 2.0 \\ c_5 &= 1.5 \end{aligned} \qquad \frac{dx_4}{dt} = t_4 + t_5 \qquad \begin{aligned} t_4 &= c_4 \\ t_5 &= c_5 \end{aligned}$$

$$x_{i+1} = x_i + c \qquad x_0 = \begin{pmatrix} 0 \end{pmatrix} \qquad c = \begin{pmatrix} 2.0 + 1.5 \end{pmatrix}$$



(a) (b) (c)

Abbildung 5.1: Lineare Beispiele: Erlaubte Strukturen

Verbotene Struktur

Als einzige Struktur, welche die Linearität verletzt, ist die Struktur mit mehreren Vorplätzen einer Transition anzusehen. Dazu muss vorausgesetzt werden, dass die Markierungen aller Vorplätze in die Berechnung der Geschwindigkeitsfunktion einer Transition eingehen und rein additive Teile dieser Geschwindigkeitsfunktionen auf mehrere Transitionen verteilt werden. Dies ist nach Definition 2.4.2 auch ohne Verlust an Modellierungskraft möglich. In Abbildung 5.2 ist ein kleines Beispiel für diesen Fall modelliert. Die folgenden Differentialgleichungen zeigen das mathematische Äquivalent dazu.

$$\begin{aligned} x_1 &= 1 \\ x_2 &= 2 \end{aligned} \qquad c_1 = 0.5 \qquad \begin{aligned} \frac{dx_1}{dt} &= -t_1 \\ \frac{dx_2}{dt} &= -t_1 \end{aligned} \qquad t_1 = c_1 * x_1 * x_2$$

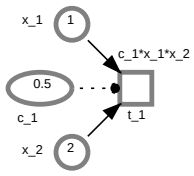
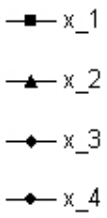
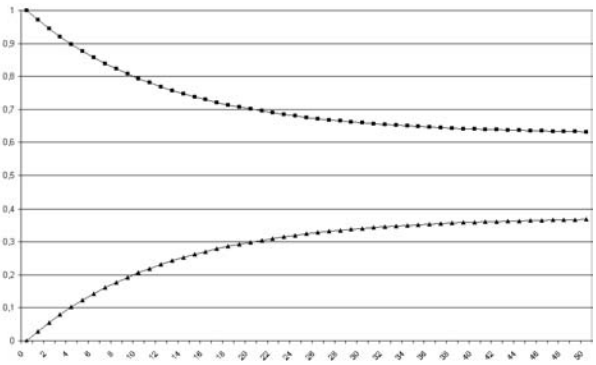


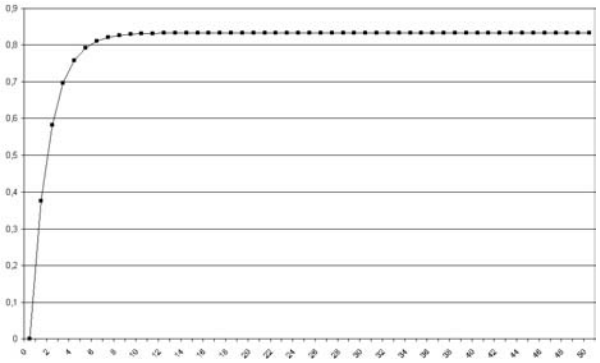
Abbildung 5.2: Lineares Beispiel: Verbotene Struktur



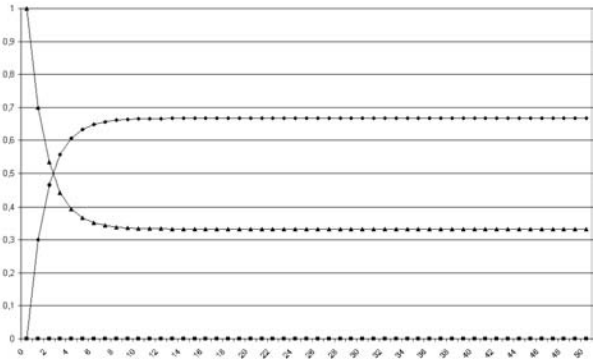
(a) Legende



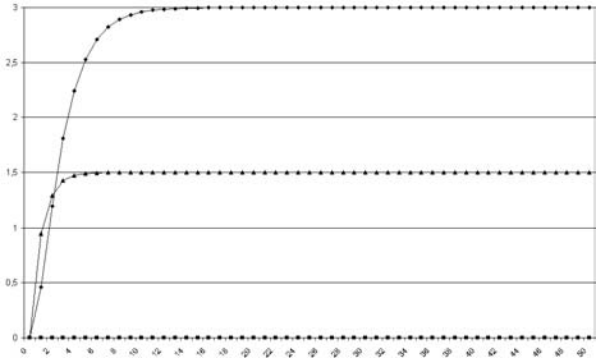
(b) BLR



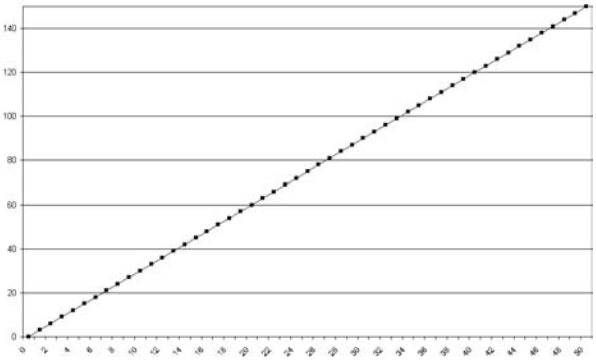
(c) ¬BLR



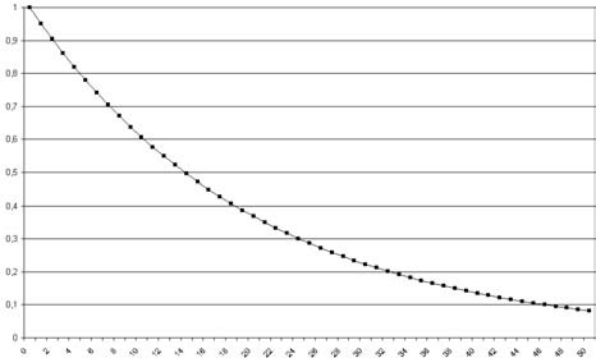
(d) B¬LR



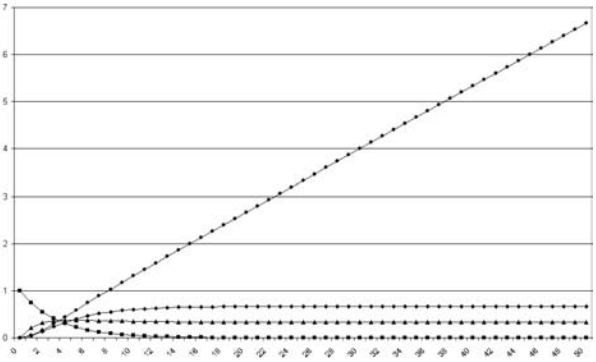
(e) ¬B¬LR



(f) ¬BL¬R



(g) B¬L¬R



(h) ¬B¬L¬R

Abbildung 5.3: Lineare Beispiele: Kurvenverlaufsdiagramme

5.1.2 Lineares Beispiel - beschränkt, lebendig, rücksetzbar

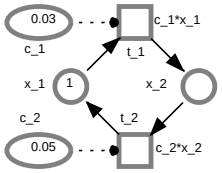
Das beschränkte, lebendige und rücksetzbare Beispiel spiegelt die erwartbaren Eigenschaften im in Abbildung 5.3 (b) dargestellten Kurvenverlauf wieder. Der Kurvenverlauf beider Plätze nähert sich einem Ausgleichszustand an, in dem die Flussgeschwindigkeiten der beiden Transitionen gleich groß sind und sich, aufgrund der entgegengesetzten Richtung der Kanten, die Waage halten. Im Folgenden werden die Differentialgleichungen des Beispieles in Gleichungsform und in Matrixform angegeben, sowie in Abbildung 5.4 das kontinuierliche Petrinetz und der INA-Ergebnisvektor dargestellt.

Differentialgleichungen in Gleichungsform

$$\begin{array}{llll} x_1 = 1 & c_1 = 0.03 & \frac{dx_1}{dt} = t_2 - t_1 & t_1 = c_1 * x_1 \\ x_2 = 0 & c_2 = 0.05 & \frac{dx_2}{dt} = t_1 - t_2 & t_2 = c_2 * x_2 \end{array}$$

Differentialgleichungen in Matrixform

$$x_{i+1} = x_i + Bx_i \qquad x_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad B = \begin{pmatrix} -0.03 & 0.05 \\ 0.03 & -0.05 \end{pmatrix}$$



ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
Y	Y	Y	Y	Y	Y	Y	Y	N	N	N	N	Y	Y	Y	Y	Y

DTP	SMC	SMD	SMA	CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S
Y	Y	Y	Y	Y	Y	Y	Y	Y	N	?	N	Y	Y	Y	Y

Abbildung 5.4: Lineares Beispiel: BLR - kontinuierliches Petrinetz, INA-Ergebnisvektor

5.1.3 Lineares Beispiel - nicht beschränkt, lebendig, rücksetzbar

Im zweiten Beispiel erfüllt das kontinuierliche Verhalten die erwarteten Eigenschaften Lebendigkeit und Rücksetzbarkeit. Dabei stellt sich ein Ausgleichszustand ein, in dem die beiden Transitionen mit gleicher Flussgeschwindigkeit schalten und das Markierungslevel des Platzes x_1 konstant gehalten wird. Die Eigenschaft der Unbeschränktheit tritt bei dem Beispiel im kontinuierlichen Modell nicht auf. Damit sind dieses Beispiel und drei der folgenden Beispiele ein guter Beleg

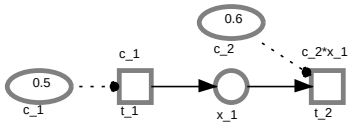
dafür, dass die Eigenschaften des diskretisierten Petrinetzes nur Indizien für das Verhalten des kontinuierlichen Petrinetzes geben. Die zugrunde liegenden Gleichungen sind im Folgenden als Gleichungssystem und in Matrixschreibweise angeben. Zusätzlich folgen das kontinuierliche Petrinetz und der INA-Ergebnisvektor in Abbildung 5.5 und der Kurvenverlauf in Abbildung 5.3 (c).

Differentialgleichungen in Gleichungsform

$$x_1 = 0 \qquad \begin{aligned} c_1 &= 0.5 \\ c_2 &= 0.6 \end{aligned} \qquad \frac{dx_1}{dt} = t_1 - t_2 \qquad \begin{aligned} t_1 &= c_1 \\ t_2 &= c_2 * x_1 \end{aligned}$$

Differentialgleichungen in Matrixform

$$x_{i+1} = x_i + Bx_i + c \qquad (x_0|c) = \left(\begin{array}{c|c} 0 & 0.5 \end{array} \right) \qquad B = \left(-0.6 \right)$$




```
ORD HOM NBM PUR CSV SCF CON SC  Ft0 tF0 Fp0 pF0 MG  SM  FC  EFC ES
 Y   Y   Y   Y   N   Y   Y   N   Y   Y   N   N   Y   N   Y   Y   Y
DTP SMC SMD SMA CPI CTI  B  SB  REV DSt BSt DTr DCF  L   LV L&S
 Y   N   N   N   N   Y   N   N   ?   N   ?   N   Y   Y   Y   N
```

Abbildung 5.5: Lineares Beispiel: ¬BLR - kontinuierliches Petrinetz, INA-Ergebnisvektor

5.1.4 Lineares Beispiel - beschränkt, nicht lebendig, rücksetzbar

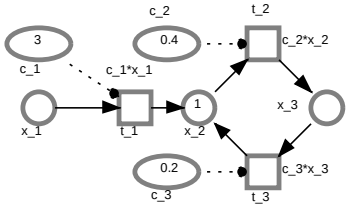
Die Eigenschaften Beschränktheit, Nichtlebendigkeit und Rücksetzbarkeit sind im Kurvenverlaufsdiagramm des kontinuierlichen Verhalten dieses Beispieles erkennbar. Alle Plätze nähern sich einem Ausgleichszustand an, womit die Eigenschaften Beschränktheit und Rücksetzbarkeit abgedeckt werden. Gleichzeitig hat der Platz x_1 einen unveränderten Wert von Null und die Transition t_1 eine daraus folgende Flussgeschwindigkeit von ebenfalls Null. Beides kann als Nichtlebendigkeit im diskreten Sinne interpretiert werden. Das Beispiel mit Differentialgleichungen in Gleichungsform und Matrixform, sowie das kontinuierliche Petrinetz mit INA-Ergebnisvektor folgt im Anschluss an diese Ausführungen.

Differentialgleichungen in Gleichungsform

$$
\begin{array}{lcl}
x_1 = 0 & c_1 = 3.0 & \frac{dx_1}{dt} = -t_1 \\
x_2 = 1 & c_2 = 0.4 & \frac{dx_2}{dt} = t_1 + t_3 - t_2 \\
x_3 = 0 & c_3 = 0.2 & \frac{dx_3}{dt} = t_2 - t_3
\end{array}
\qquad
\begin{array}{lcl}
t_1 = c_1 * x_1 \\
t_2 = c_2 * x_2 \\
t_3 = c_3 * x_3
\end{array}
$$

Differentialgleichungen in Matrixform

$$x_{i+1} = x_i + Bx_i \qquad x_0 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \qquad B = \begin{pmatrix} -3.0 & 0 & 0 \\ 3.0 & -0.4 & 0.2 \\ 0 & 0.4 & -0.2 \end{pmatrix}$$




```
ORD HOM NBM PUR CSV SCF CON SC  Ft0 tF0 Fp0 pF0 MG  SM  FC  EFC ES
 Y   Y   N   Y   Y   Y   Y   N   N   N   Y   N   N   Y   Y   Y   Y
DTP SMC SMD SMA CPI CTI  B  SB  REV DSt BSt DTr DCF  L   LV L&S
 N   ?   N   N   Y   N   Y   Y   ?   N   ?   ?   Y   N   ?   N
```

Abbildung 5.6: Lineares Beispiel: B¬LR - kontinuierliches Petrinetz, INA-Ergebnisvektor

5.1.5 Lineares Beispiel - nicht beschränkt, nicht lebendig, rücksetzbar

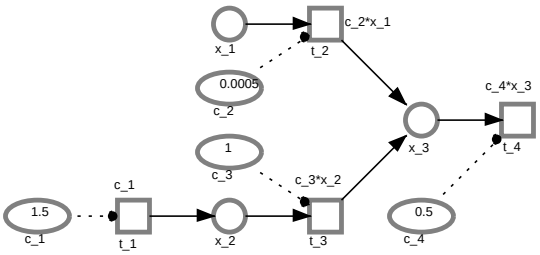
Dieses Beispiel erfüllt in seinem kontinuierlichen Verhalten die erwarteten Eigenschaften nur teilweise. Es stellt sich ein Ausgleichszustand ein und die Flussgeschwindigkeit für Transition t_2 ist zu jedem Zeitpunkt gleich Null. Damit sind die Eigenschaften Nichtlebendigkeit und Rücksetzbarkeit erfüllt. Allerdings ist ein Ausgleichszustand immer beschränkt, so dass sich die Eigenschaft der Nichtbeschränktheit im kontinuierlichen Verhalten nicht äußert.

Differentialgleichungen in Gleichungsform

$$\begin{aligned} x_1 &= 0 \\ x_2 &= 0 \\ x_3 &= 0 \end{aligned} \qquad \begin{aligned} c_1 &= 1.5 \\ c_2 &= 0.0005 \\ c_3 &= 1.0 \\ c_4 &= 0.5 \end{aligned} \qquad \begin{aligned} \frac{dx_1}{dt} &= -t_2 \\ \frac{dx_2}{dt} &= t_1 - t_3 \\ \frac{dx_3}{dt} &= t_3 + t_2 - t_4 \end{aligned} \qquad \begin{aligned} t_1 &= c_1 \\ t_2 &= c_2 * x_1 \\ t_3 &= c_3 * x_2 \\ t_4 &= c_4 * x_3 \end{aligned}$$

Differentialgleichungen in Matrixform

$$x_{i+1} = x_i + Bx_i + c \qquad (x_0|c) = \left(\begin{array}{c|c} 0 & 0 \\ 0 & 1.5 \\ 0 & 0 \end{array}\right) \qquad B = \begin{pmatrix} -0.0005 & 0 & 0 \\ 0 & -1.0 & 0 \\ 1.5 & 1.0 & -0.5 \end{pmatrix}$$



ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
Y	Y	N	Y	N	Y	Y	N	Y	Y	Y	N	N	N	Y	Y	Y
DTP	SMC	SMD	SMA	CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S	
N	N	N	N	N	N	N	N	?	?	?	?	Y	N	?	N	

Abbildung 5.7: Lineares Beispiel: ¬B¬LR - kontinuierliches Petrinetz, INA-Ergebnisvektor

5.1.6 Lineares Beispiel - beschränkt, lebendig, nicht rücksetzbar

Für diese Kombination der Eigenschaften Beschränktheit, Lebendigkeit und Rücksetzbarkeit existieren nur Netze mit mindestens einer Transition mit zwei Vorplätzen. Nach den in Kapitel 5.1.1 getroffenen Restriktionen an das kontinuierliche Petrinetz zum Erhalt der Linearität, existiert kein lineares kontinuierliches Petrinetz mit den Eigenschaften beschränkt, lebendig und nicht rücksetzbar.

5.1.7 Lineares Beispiel - nicht beschränkt, lebendig, nicht rücksetzbar

Der Kurvenverlauf des in diesem Beispiel verwendeten Platzes spiegelt die erwarteten Eigenschaften des diskretisierten Petrinetzes wieder. Die Flussgeschwindigkeit ist konstant positiv, wodurch

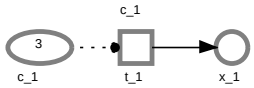
die Markierung des Platzes x_1 konstant ansteigt und unbeschränkt wächst. Aus diesem Verhalten resultiert weiterhin, dass das kontinuierliche Petrinetz lebendig und nicht rücksetzbar ist.

Differentialgleichungen in Gleichungsform

$$x_1 = 0 \qquad c_1 = 3.0 \qquad \frac{dx_1}{dt} = t_1 \qquad t_1 = c_1$$

Differentialgleichungen in Matrixform

$$x_{i+1} = x_i + c \qquad (x_0|c) = \left(\begin{array}{c|c} 0 & 3.0 \end{array} \right)$$



ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
Y	Y	Y	Y	N	Y	Y	N	Y	N	N	Y	N	N	Y	Y	Y
DTP	SMC	SMD	SMA	CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S	
Y	N	N	N	N	N	N	N	?	N	?	N	Y	Y	Y	N	

Abbildung 5.8: Lineares Beispiel: ¬BL¬R - kontinuierliches Petrinetz, INA-Ergebnisvektor

5.1.8 Lineares Beispiel - beschränkt, nicht lebendig, nicht rücksetzbar

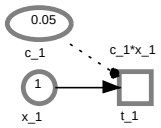
In dem folgenden Beispiel zeigt die Kurve des Platzes einen asymptotisch gegen Null gehenden Verlauf. Dieser Grenzwert wird aber nicht erreicht und die Transition besitzt eine Flussgeschwindigkeit größer Null. Daraus resultiert, dass das Beispiel beschränkt und lebendig ist. Aufgrund der stetigen Abnahme der Markierung des Platzes ist das kontinuierliche Verhalten nicht rücksetzbar.

Differentialgleichungen in Gleichungsform

$$x_1 = 1 \qquad c_1 = 0.05 \qquad \frac{dx_1}{dt} = -t_1 \qquad t_1 = c_1 * x_1$$

Differentialgleichungen in Matrixform

$$x_{i+1} = x_i + Bx_i \qquad x_0 = \left(\begin{array}{c} 1 \end{array} \right) \qquad B = \left(\begin{array}{c} -0.05 \end{array} \right)$$



ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
Y	Y	N	Y	Y	Y	Y	N	N	N	Y	N	N	Y	Y	Y	Y
DTP	SMC	SMD	SMA	CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S	
N	?	N	N	Y	N	Y	Y	?	N	?	?	Y	N	?	N	

Abbildung 5.9: Lineares Beispiel: B¬L¬R - kontinuierliches Petrinetz, INA-Ergebnisvektor

5.1.9 Lineares Beispiel - nicht beschränkt, nicht lebendig, nicht rücksetzbar

Für die letzte Kombination der drei unabhängigen Eigenschaften zeigt das gewählte Beispiel eine stetig mit konstantem Anstieg wachsende Kurve und drei Kurven, die einen Ausgleichszustand erreichen. Damit ist die Eigenschaft der Nichtlebendigkeit verletzt, da keine Transition eine Flussgeschwindigkeit von Null erreicht. Die Unbeschränktheit und die Nichtrücksetzbarkeit ist mit dem Platz mit stetig wachsender Markierung erfüllt.

Differentialgleichungen in Gleichungsform

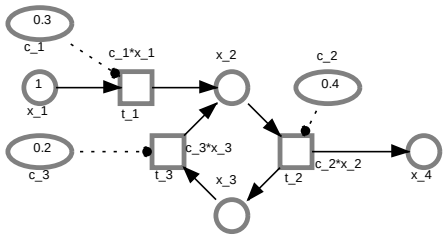
$$\begin{aligned} x_1 &= 1 \\ x_2 &= 0 \\ x_3 &= 0 \\ x_4 &= 0 \end{aligned} \qquad \begin{aligned} c_1 &= 0.3 \\ c_2 &= 0.4 \\ c_3 &= 0.2 \end{aligned} \qquad \begin{aligned} \frac{dx_1}{dt} &= -t_1 \\ \frac{dx_2}{dt} &= t_1 + t_3 - t_2 \\ \frac{dx_3}{dt} &= t_2 - t_3 \\ \frac{dx_4}{dt} &= t_2 \end{aligned} \qquad \begin{aligned} t_1 &= c_1 * x_1 \\ t_2 &= c_2 * x_2 \\ t_3 &= c_3 * x_3 \end{aligned}$$

Differentialgleichungen in Matrixform

$$x_{i+1} = x_i + Bx_i \qquad x_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \qquad B = \begin{pmatrix} -0.3 & 0 & 0 \\ 0.3 & -0.4 & 0.2 \\ 0 & 0.4 & -0.2 \end{pmatrix}$$

5.2 Praktische Beispiele aus der aktuellen Forschung

Zusätzlich zu den vorstehenden theoretischen Beispielen wurden fünf Anwendungen aus der aktuellen Forschung in kontinuierliche Petrinetze überführt. Ziel war es, die praktische Anwendbarkeit



ORD	HOM	NBM	PUR	CSV	SCF	CON	SC	Ft0	tF0	Fp0	pF0	MG	SM	FC	EFC	ES
Y	Y	N	Y	N	Y	Y	N	N	N	Y	Y	N	N	Y	Y	Y
DTP	SMC	SMD	SMA	CPI	CTI	B	SB	REV	DSt	BSt	DTr	DCF	L	LV	L&S	
N	N	N	N	N	N	N	N	?	?	?	N	Y	N	N	N	

Abbildung 5.10: Lineares Beispiel: ¬B¬L¬R - kontinuierliches Petrinetz, INA-Ergebnisvektor

der kontinuierlichen Petrinetze als Modellierungsvariante nachzuweisen und für die Validierung der Simulationseinheit des entwickelten Werkzeuges, Beispiele unterschiedlicher Größe zu erhalten. Alle Beispiele sind Vertreter biochemischer Abläufe, die zugrunde liegenden Differentialgleichungen wurden den zitierten Artikeln entnommen und es lag eine Referenzmodellierung im Dateiformat von Gepasi/Copasi oder in SBML vor. Dadurch war gesichert, dass für alle fünf Beispiele eine Validierung an einem anerkannten Werkzeug durchgeführt werden konnte. Als Beispiele wurden ausgewählt:

- Einfluss des Raf Kinase Inhibitor Proteins (RKIP) auf die außerzelluläre signalregulierte Kinase (ERK), modelliert in [CSK+03];
- Mitogen-aktivierte Proteinkinasen (MAPK) Kaskade, konstruiert in [Kho00];
- Pfadmodell von Fettmetaboliten in der Leber, vorgestellt in [KSLR00];
- Kinetisches Modell des Sucroseabbaus in der Kartoffel, entwickelt in [Jun04] und
- Dynamik des Pheromonpfadweges in der Hefe, entworfen in [KK04]

In den folgenden Abschnitten werden zu allen fünf Beispielen die erstellten kontinuierlichen Petrinetze abgebildet. Die resultierenden Differentialgleichungen, sowie Simulationsergebnisse und Tabellen zum Vergleich der in Snoopy implementierten Simulationsverfahren mit den Ergebnissen des Validierungswerkzeuges Copasi sind im Anhang B zu finden.

5.2.1 RKIP-Beispiel

Die Struktur des zu diesem Beispiel erstellten kontinuierlichen Petrinetzmodells ist in Abbildung 5.11 abgebildet. Die für die Simulation gewählten Parameter werden in Abbildung 5.12 gezeigt. Differentialgleichungen, Zahlenmaterial und Diagramme zu diesem Beispiel sind im Anhang B.1 zu finden.

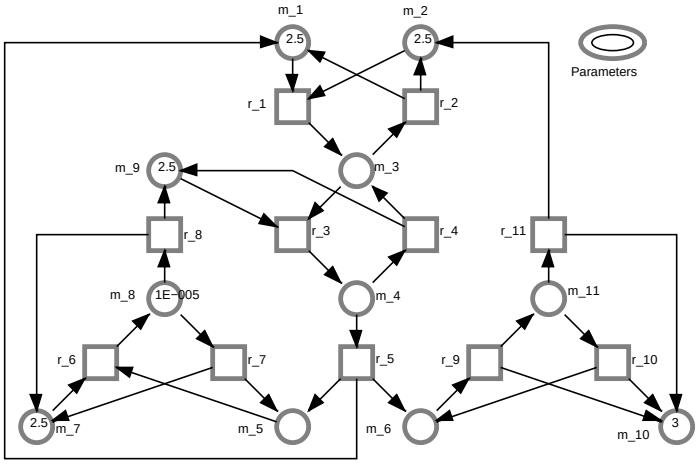


Abbildung 5.11: RKIP: Hauptstruktur

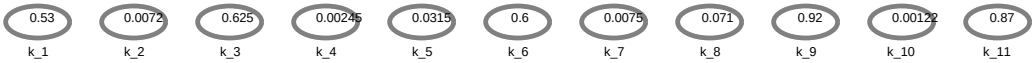


Abbildung 5.12: RKIP: Parameter

5.2.2 MAPK-Kaskaden-Beispiel von Kholodenko

In Abbildung 5.14 ist die Struktur der MAPK-Kaskade, modelliert nach der Veröffentlichung von Boris Kholodenko, abgebildet und in Abbildung 5.13 sind die gewählten Parameter aufgeführt.

Abbildung 5.13: MAPK-Kaskade: Parameter

5.2.3 Leber-Beispiel von Knoblauch/Reich

Die Komplexität der Struktur hat es bei diesem Beispiel notwendig gemacht, die Abhängigkeiten der einzelnen Geschwindigkeitsfunktionen, über die Grundstruktur des Problems hinausgehend, in Unternetze zu legen. Die Grundstruktur ist in Abbildung 5.15 und die verwendeten Parameter sind in Abbildung 5.16 abgebildet. Die einzelnen Unternetze sind aufgrund ihrer Anzahl im Anhang B.3 zusammen mit den Daten zur Simulation des Beispieles aufgeführt.

5.2.4 Kartoffel-Beispiel von Junker

Beim Kartoffel-Beispiel sind umkehrbare Transitionen und fixierte Plätze verwendet worden. Damit werden bei der Validierung der Simulation auch diese Netzelemente einem Test unterzogen. Das entworfene kontinuierliche Petrinetz ist in Abbildung 5.17 und die eingesetzten Parameter sind in Abbildung 5.18 abgebildet.

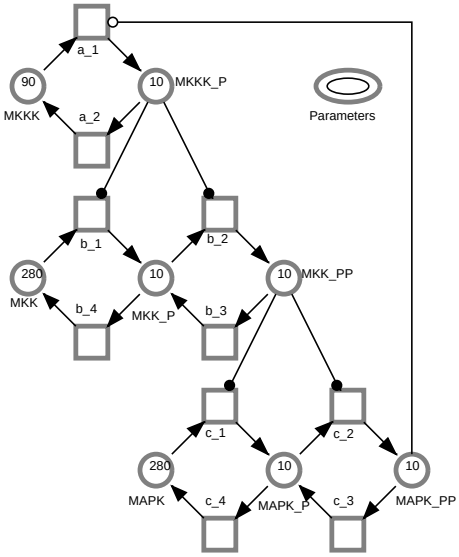


Abbildung 5.14: MAPK-Kaskade: Hauptstruktur

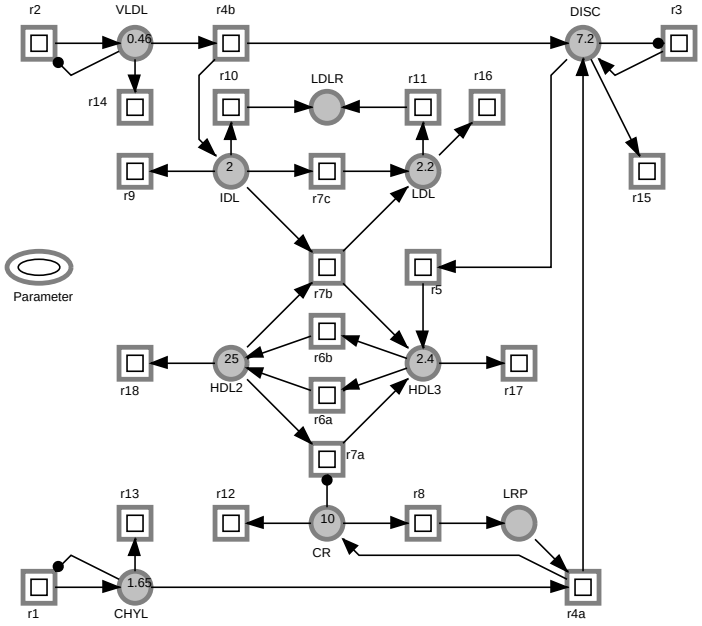


Abbildung 5.15: Leber: Hauptstruktur

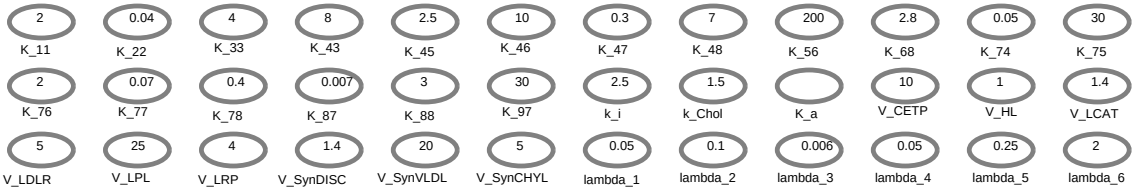


Abbildung 5.16: Leber: Parameter

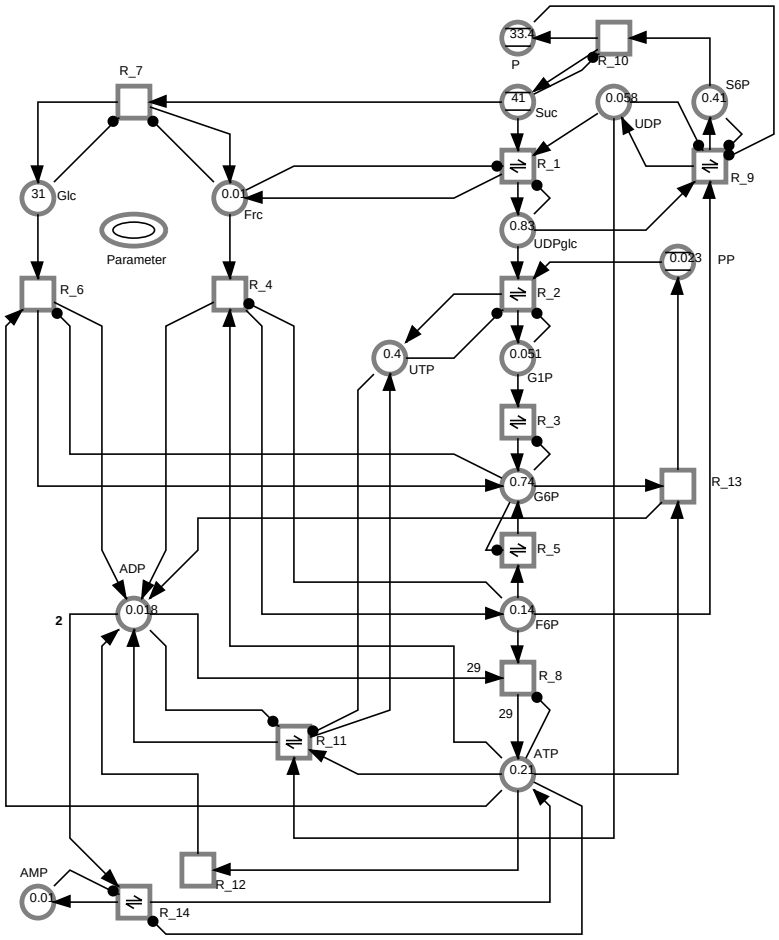


Abbildung 5.17: Kartoffel: Hauptstruktur

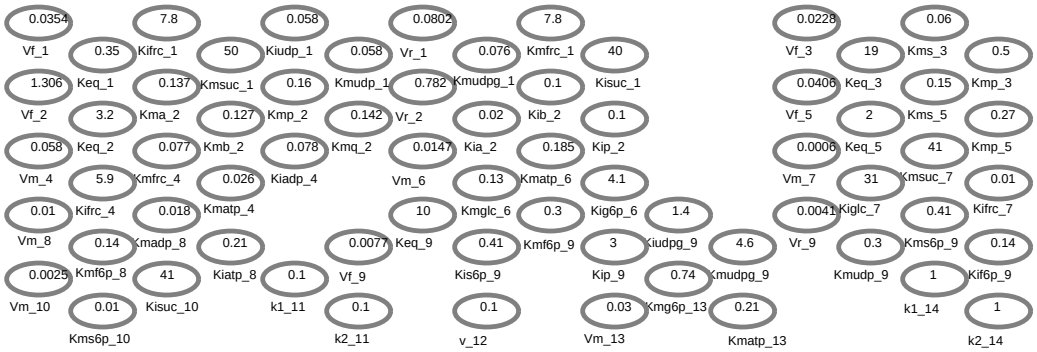


Abbildung 5.18: Kartoffel: Parameter

5.2.5 Hefe-Beispiel von Klipp/Kofahl

Der Pheromonpfadweg in der Hefe stellt das komplexeste der hier betrachteten Beispiele dar. Das Modell ist mit 47 Transitionen, 35 Plätzen und 47 Parametern das mit Abstand größte Netz, welches simuliert wurde. Die Transitionen wurden aufgrund der Komplexität und um mehr Übersicht zu schaffen in einzelne Teilnetze gekapselt. Dabei erhielt jede Transition ihr eigenes Teilnetz. Dadurch ist die Grundstruktur der Anwendung im Hauptnetz erkennbar und in den Teilnetzen können gleichzeitig die Zusammenhänge zwischen einzelnen Transitionen und beteiligten Plätzen gesehen werden. Die Hauptstruktur ist in Abbildung 5.20 und die verwendeten Parameter in Abbildung 5.19 dargestellt. Die Teilnetze sind im Anhang B.5 zu finden.

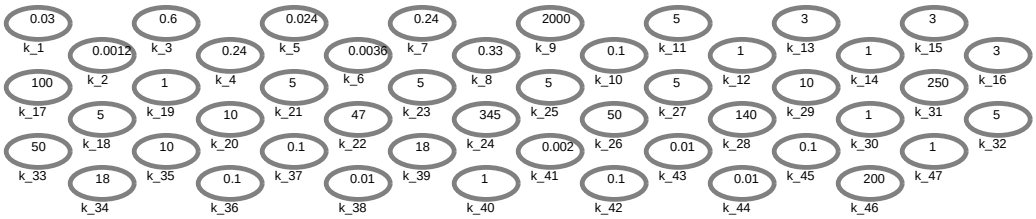


Abbildung 5.19: Hefe: Parameter

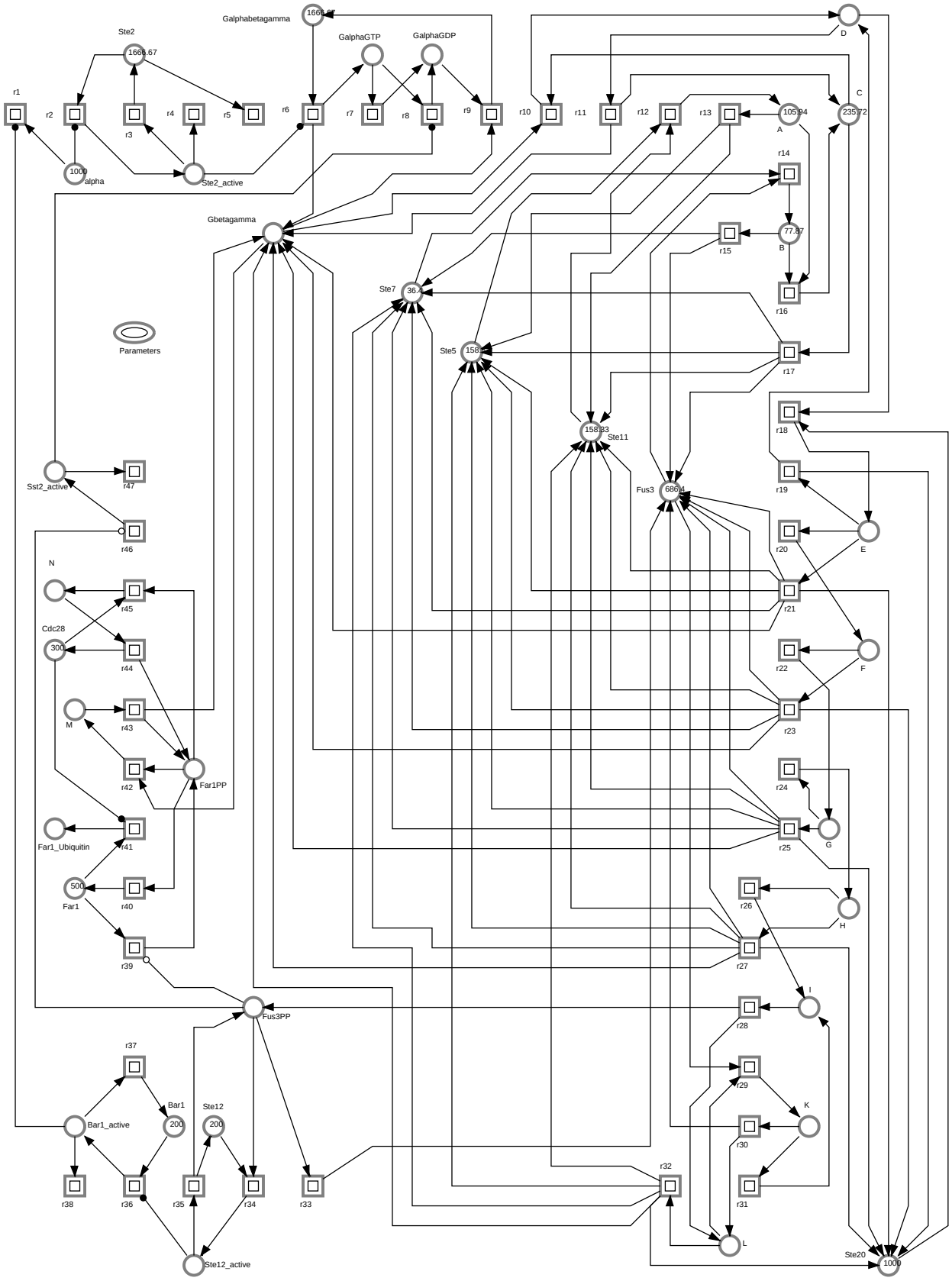


Abbildung 5.20: Hefe: Hauptstruktur

Kapitel 6

Erkenntnisse zum Zusammenhang diskreter Petrinetze und gewöhnlicher Differentialgleichungen

Im Rahmen der Tätigkeiten dieser Diplomarbeit wurden mehrere Erkenntnisse zum Zusammenhang von diskreten Petrinetzen und gewöhnlichen Differentialgleichungen gewonnen. Teilweise sind es unbewiesene Vermutungen, teilweise zeigen sie Verhaltensmuster auf, die besonders betrachtet werden sollten. Aufgrund der vielen Möglichkeiten, die mit den Freiheiten bei der Wahl der Differentialgleichungen einher gehen, sind alle folgenden Argumentationen nur Indizien für beobachtbares Verhalten. Bei strukturgemäßer Modellierung sind es aber sehr wahrscheinlich korrekte Verhaltensweisen.
Die Arbeit mit den Beispielanwendungen hat frühzeitig folgende Zusammenhänge aufgedeckt:

- Zusammenhang zwischen additiven Anteilen einer gewöhnlichen Differentialgleichung und mehrere Transitionen
- Zusammenhang zwischen Rücksetzbarkeit des Petrinetzes und oszillierendem Verhalten des gewöhnlichen Differentialgleichungssystems
- Zusammenhang zwischen Rücksetzbarkeit des Petrinetzes und Gleichgewichtszustand des gewöhnlichen Differentialgleichungssystems
- Zusammenhang zwischen Nicht-Lebendigkeit des Petrinetzes und Schaltgeschwindigkeit ist asymptotisch zu Null
- Zusammenhang zwischen P-Invarianten des Petrinetzes und Moieties der gewöhnlichen Differentialgleichungen

In den folgenden Absätzen werden die Zusammenhänge eingehender beschrieben und an Hand kurzer Beispiele illustriert.

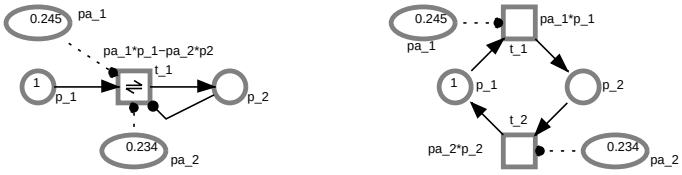
Zusammenhang zwischen additiven Anteilen einer gewöhnlichen Differentialgleichung und mehrere Transitionen

Ein nach Definition 2.2.3 beweisbarer Zusammenhang besteht zwischen additiven Anteilen einer gewöhnlichen Differentialgleichung und einer äquivalenten Modellierung mit Hilfe mehrerer

Transitionen ohne additive Anteile. Beispielsweise kann eine Differentialgleichung der Form

$$pa_1 * p_1 - pa_2 * p_2$$

in einer Transition oder in je einer Transition für jeden Term modelliert werden. Die zugehörigen beiden Petrinetzvarianten sind in Abbildung 6.1 dargestellt.

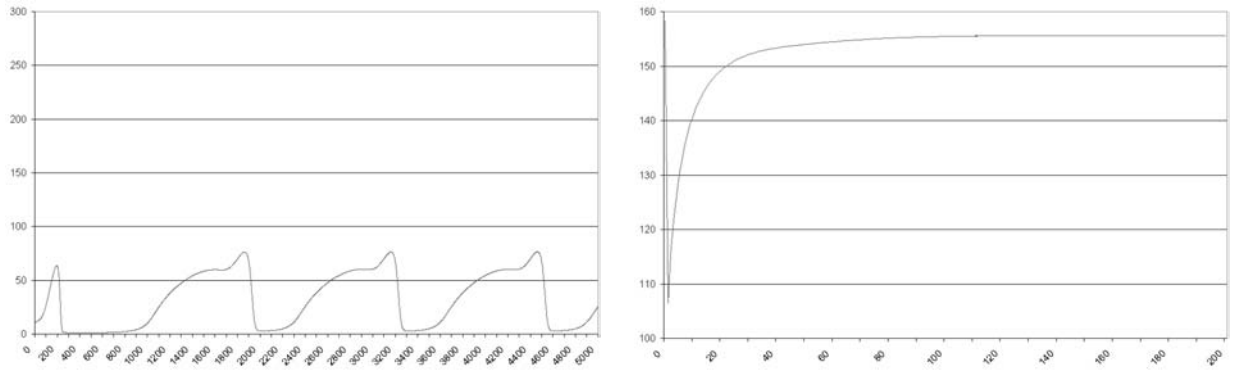


(a) Umkehrbare Transition (b) Zwei Transitionen

Abbildung 6.1: Umkehrbare Transition und zwei Transitionen

Zusammenhang wischen Rücksetzbarkeit des Petrinetzes und Verhalten des gewöhnlichen Differentialgleichungssystems

Aus der Petrinetzeigenschaft der Rücksetzbarkeit kann zweierlei Verhalten der gewöhnlichen Differentialgleichungen resultieren. Einerseits kann das System ein oszillierendes Verhalten zeigen oder andererseits einen Gleichgewichtszustand mit (fast) konstanten Markierungswerten der einzelnen Plätze bzw. Variablen erreichen. Beide Systeme sind lebendig, da die Schaltgeschwindigkeit ungleich Null ist. Typische Kurvenverläufe für beide Fälle sind im Kurvenverlauf von $MAPK_P$ aus dem MAPK-Kaskaden-Beispiel und im Kurvenverlauf von Ste5 aus dem Hefe-Beispiel zu finden. In Abbildung 6.2 sind die Kurvenverläufe dargestellt.



(a) Oszillation: $MAPK_P$ (b) Gleichgewichtszustand: Ste5

Abbildung 6.2: Beispiel für Oszillation und Gleichgewichtszustand

Zusammenhang zwischen Nicht-Lebendigkeit und Schaltgeschwindigkeit

Der Zusammenhang zwischen Nicht-Lebendigkeit und einer Schaltgeschwindigkeit nahe Null ist schwer erkennbar. Aufgrund der hohen Freiheit bei der Wahl der Differentialgleichungen zur Be-

rechnung der Schaltgeschwindigkeit lassen sich kaum Rückschlüsse auf die Lebendigkeit des kontinuierlichen Petrinetzes ziehen. Betrachtet man diese Argumentation invers, kann ein nicht lebendiges Petrinetzmodell eines lebendigen Differentialgleichungssystems als fehlerbehaftet betrachtet werden. Weitergehend zeigen die toten Teile des Petrinetzes direkt die genauer zu untersuchenden Modellbereiche an.

Zusammenhang zwischen P-Invarianten und Moieties

Als **Moiety** werden in der Literatur chemische Stoffgruppen bezeichnet, deren Konzentration in Summe konstant ist. Dieser Begriff wird im bereits vorgestellten Werkzeug Gepasi/Copasi in dieser Bedeutung verwendet.

Die beschriebene Eigenschaft der konstanten Konzentrationen von Stoffen zeigt Ähnlichkeit mit den P-Invarianten der Petrinetze. Einige Beispiele haben diese Vermutung unterstrichen und diese Eigenschaft wurde beim Kartoffelbeispiel auch genutzt, um zwei P-Invarianten zu identifizieren, welche im Gleichgewichtszustand eine Flussgeschwindigkeit von Null besitzen. Diese Teile des Modells sind für den aktuellen Fall überflüssig und können ohne Informationsverlust aus dem Modell entfernt werden.

Für das MAPK-Kaskaden-Beispiel sind im Artikel [Kho00] von Kholodenko drei Moieties angeben:

$$\begin{aligned} \mathrm{MKKK} + \mathrm{MKKK_P} &= 100 \\ \mathrm{MKK} + \mathrm{MKK_P} + \mathrm{MKK_{PP}} &= 300 \\ \mathrm{MAPK} + \mathrm{MAPK_P} + \mathrm{MAPK_{PP}} &= 300 \end{aligned}$$

Diese entsprechen genau den drei P-Invarianten des Beispieles.

Kapitel 7

Zusammenfassung und Ausblick

Zum Abschluss dieser Diplomarbeit wird eine Zusammenfassung und ein Ausblick auf weitere mögliche Aufgaben gegeben. Als Zusammenfassung soll dabei auf die geschaffenen Möglichkeiten der Modellierung und Analyse biochemischer Prozesse eingegangen werden. Im Ausblick werden weitergehende Arbeiten am entwickelten Werkzeug vorgeschlagen.

7.1 Zusammenfassung

Im Rahmen dieser Diplomarbeit wurde ein Werkzeug implementiert, welches es ermöglicht, kontinuierliche Petrinetze zu modellieren und zu simulieren. Dazu wurden zu Beginn kontinuierliche Petrinetze definiert und mit Ausrichtung auf biochemische Netze mit speziellen Erweiterungen, wie Parametern, umkehrbaren Transitionen und fixierten Plätzen, versehen. Die gewählte Definition der kontinuierlichen Petrinetze lässt es zu, dass die Netze für die Simulation als gewöhnliche Differentialgleichungssysteme interpretiert werden. Die Arbeiten an den praktischen Beispielanwendungen haben gezeigt, dass mittels gewöhnlichen Differentialgleichungen modellierte biochemische Prozesse übersichtlich und strukturiert als kontinuierliche Petrinetze umgesetzt werden können, und es dabei zu keinem Informationsverlust kommt.

In der Literatur werden kontinuierliche Petrinetze über zeitbewertete Petrinetze oder über autonome Netze mit reellwertigen Marken definiert. Die in dieser Arbeit verwendete Sichtweise, kontinuierliche Petrinetze als Äquivalent zu gewöhnlichen Differentialgleichungen zu betrachten, wurde von keinem Autor eingeführt.

Die Simulation in Snoopy bietet reichhaltige Konfigurationsmöglichkeiten und orientiert sich dabei an gängigen Entwicklungswerkzeugen für biochemische Anwendungen. Dazu stehen mehrere Verfahren zur Verfügung, welche vom Nutzer ausgewählt werden können. Dies sind mehrere Varianten vom expliziten Runge-Kutta-Verfahren und das Rosenbrock-Verfahren mit mehreren alternativen Koeffizienten. Weiterhin können Einstellungen wie Schrittweite, Schrittanzahl, relativer Fehler und Ausgabekanal konfiguriert werden. Die untersuchten Werkzeuge für die biochemische Anwendungsdomäne bieten oftmals vergleichbare Konfigurationsmöglichkeiten. Sie haben zusätzlich eine automatische Verfahrensauswahl integriert, um den Anwender von der Entscheidung, welches Verfahren das günstigste ist, zu befreien. Diese Funktion fehlt dem entwickelten Werkzeug Snoopy bisher.

Die Verifikation des entwickelten Werkzeuges an Hand der vorgestellten Beispielanwendungen hat ergeben, dass die Simulationsergebnisse mit höherer Genauigkeit ausgegeben werden. Das als Vergleichswerkzeug dienende Werkzeug Copasi liefert Daten mit sechs Stellen Genauigkeit. Das entwickelte Werkzeug gibt die Ergebnisse mit maximaler Rechengenauigkeit der zugrunde liegenden Programmiersprache C++ aus. Die dadurch erzielte Rechengenauigkeit ist besser als die vom Referenzwerkzeug erzielte, was sich im absoluten Fehler wiederspiegelt, welcher niedriger als die Ausgabegenauigkeit von Copasi ist. Beim Zeitverhalten ist Copasi als Referenzwerkzeug dem vorliegenden Simulationswerkzeug für kontinuierliche Petrinetze wesentlich überlegen. Dies lässt sich, wie bereits in Abschnitt 4.3 festgestellt, im Wesentlichen auf die große Vielfalt an numerischen Verfahren und die lange Entwicklungszeit des Werkzeuges Copasi zurückführen.

7.2 Ausblick und weitergehende Aufgaben

Zukünftige Arbeiten im Umfeld der kontinuierlichen Petrinetze können fundierte Aussagen über das Zusammenspiel von diskreten Petrinetzen mit den ausgeprägten Analysemethoden und kontinuierlichen Petrinetzen mit den größeren Freiheiten beim Verhalten hervorbringen. Dazu können die im Kapitel 6 genannten Beobachtungen Ausgangspunkt sein. Weitergehende Zusammenhänge zum Beispiel zwischen T-Invarianten und Reaktionsgruppen sind bereits Gegenstand von Forschungen zur Unterstützung des biochemischen Modellierens.

Mögliche weitergehende Aufgaben am Werkzeug Snoopy sind im Kontext der kontinuierlichen Petrinetze:

- Integration weiterer numerischer Verfahren zur Lösung gewöhnlicher Differentialgleichungen, um das Zeitverhalten günstiger zu gestalten
- Implementierung einer optionalen, automatischen Verfahrensauswahl, um den Nutzer von der Wahl des optimalen Verfahrens zu entlasten
- Weitergehende Konsistenzprüfungen, wie zum Beispiel eine Prüfung der Markierungswerte von kontinuierlichen Plätzen auf negative Werte
- Zusätzliche Auswertungsverfahren für die Simulationsergebnisse und Verbesserung der bestehenden graphischen Auswertungsmöglichkeiten
- Unterstützung einer Messung der Simulationszeiten
- Informationserhaltende Konvertierung in diskrete Petrinetze und später zurück in kontinuierliche Petrinetze
- Analyse des Zeitverhaltens und Beseitigung von eventuellen Engpässen
- Stärkere Unterstützung des Modellierungsprozesses durch Umwandlung diskreter in kontinuierliche Petrinetze und umgekehrt, ohne Informationsverlust

Weiterhin kann eine Erweiterung der in dieser Arbeit eingeführten kontinuierlichen Petrinetze zu hybriden Petrinetzen angestrebt werden. Die untersuchten Programmpakete Visual Object Net und Genomic Object Net arbeiten auf Petrinetzebene mit hybriden Petrinetzen. Theoretische Vorarbeiten zu hybriden Petrinetzen wurden in [DA05] geleistet.

Anhang A

Implementierungsstruktur

Dieser Anhang beschreibt die Struktur des Werkzeuges und zeigt die für den Nutzer zur Verfügung stehenden Einstellungen. Es werden die implementierte Netzklasse der kontinuierlichen Petrinetze mit ihren Knoten- und Kantenarten, sowie die zu den jeweiligen Netzelementen gehörigen Attribute benannt. Außerdem werden die Möglichkeiten des Exports und die Konfigurationsmöglichkeiten für die Simulation aufgezählt.

A.1 Netzklasse Continuous Petri Net

Die neu erstellte Netzklasse der kontinuierlichen Petrinetze innerhalb des Graphmanipulationswerkzeuges Snoopy heißt **Continuous Petri Net** und hat folgende Strukturelemente:

- Knotenarten:
    1. Place, Continuous
    2. Transition, Continuous
    3. Parameter, Continuous
    4. Coarse Place, Continuous
    5. Coarse Transition, Continuous
    6. Coarse Parameter, Continuous
    7. Plot
- Kantenarten:
    1. Edge, Continuous
    2. Test Edge, Continuous
    3. Inhibitor Edge, Continuous
    4. Parameter Test Edge, Continuous
    5. Parameter Inhibitor Edge, Continuous

A.1.1 Kontinuierliche Petrinetzknoten

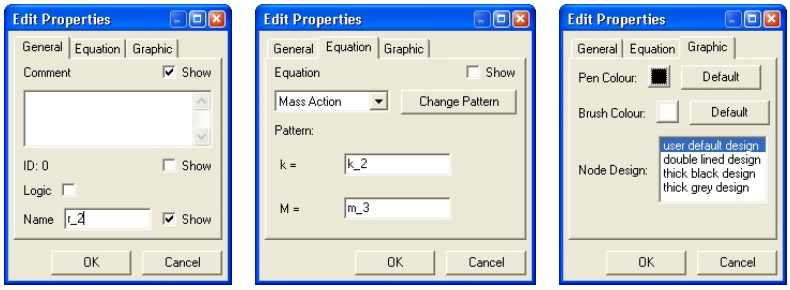
Das angestrebte Anwendungsgebiet und die Kontinuierlichkeit der Markenwerte bedingen, dass die Plätze und Transitionen des Petrinetzes spezielle Eigenschaften erhalten. Zusätzlich wurden Parameter eingeführt, um eine einfache Manipulierbarkeit der zugrunde liegenden Differentialgleichungen zu ermöglichen. In den folgenden Bereichen werden kurz die einzelnen Knotenarten vorgestellt und die jeweiligen Attribute benannt.

Kontinuierliche Plätze

Der Knotentyp **Place, Continuous** besitzt die Attribute **ID**, **Name**, **Markierung**, **Fixiertheit** und **Kommentar**.

Kontinuierliche Transitionen

Der Knotentyp **Transition, Continuous** besitzt die Attribute **ID**, **Name**, **Gleichung**, **Umkehrbarkeit** und **Kommentar**.



(a) Hauptseite (b) Gleichungsseite (c) Graphikseite

Abbildung A.1: Snoopy: Eigenschaftendialog für kontinuierliche Transitionen

Kontinuierliche Parameter

Der Knotentyp **Parameter, Continuous** besitzt die Attribute **ID**, **Name**, **Markierungsliste** und **Kommentar**.

Kontinuierliche Grobplätze, -transitionen und -parameter

Die Knotentypen **Coarse Place, Continuous**, **Coarse Transition, Continuous** und **Coarse Parameter, Continuous** besitzen das Attribut **Name**.

Plot

Der Knotentyp **Plot** besitzt die Attribute **ID**, **Name**, **Platzliste**, $\mathbf{X_{min}}$, $\mathbf{X_{max}}$, $\mathbf{Y_{min}}$, $\mathbf{Y_{max}}$ und **Kommentar**.

A.1.2 Kontinuierliche Petrinetzkanten

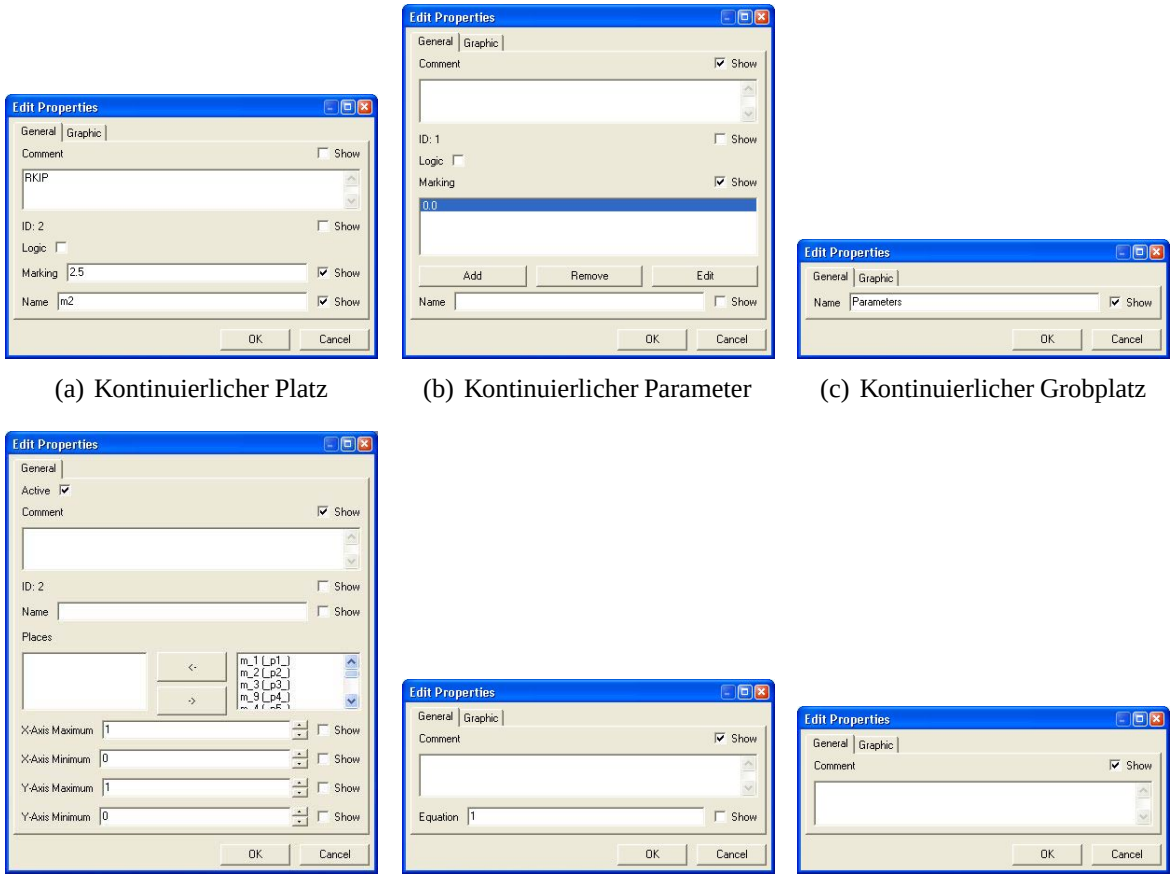
Aufgrund der kontinuierlichen Markierungswerte sind auch kontinuierliche Petrinetzkanten notwendig. Diese werden in diesem Bereich kurz vorgestellt.

Kontinuierliche Kante

Der Knotentyp **Edge, Continuous** besitzt die Attribute **Gleichung** und **Kommentar**.

Kontinuierliche Inhibitor- und Testkante, sowie Kontinuierliche Parameter Inhibitor- und Testkante

Die Knotentypen **Inhibitor Edge, Continuous**, **Test Edge, Continuous**, **Parameter Inhibitor Edge, Continuous** und **Parameter Test Edge, Continuous** besitzen das Attribut **Kommentar**.



(a) Kontinuierlicher Platz (b) Kontinuierlicher Parameter (c) Kontinuierlicher Grobplatz

(d) Plot (e) Kontinuierliche Kante (f) Kontinuierliche Inhibitorkante

Abbildung A.2: Snoopy: Eigenschaftendialoge

A.1.3 Export in SBML

Für die Netzklasse *Continuous Petri Net* wurde ein Export in das in der Biochemie gebräuchliche Austauschformat SBML implementiert. Dabei werden alle in der Netzklasse konfigurierbaren

Eigenschaften verwendet. Insbesondere die Eigenschaft Umkehrbarkeit findet an dieser Stelle Anwendung.

A.1.4 Export in LaTeX

Es wurde ein Export in LaTeX implementiert. Die Exportroutine besitzt mehrere Einstellungsmöglichkeiten. Zum ersten kann zwischen sieben verschiedenen Exportmodi gewählt werden:

1. Export der Differentialgleichungen in LaTeX
2. Export der Startmarkierungen in LaTeX
3. Export der Reaktionsgleichungen in LaTeX
4. gemeinsamer Export der Differentialgleichungen und der Startmarkierungen in LaTeX
5. gemeinsamer Export der Differentialgleichungen und der Reaktionsgleichungen in LaTeX
6. gemeinsamer Export der Startmarkierungen und der Reaktionsgleichungen in LaTeX
7. gemeinsamer Export der Differentialgleichungen, der Startmarkierungen und der Reaktionsgleichungen in LaTeX

Ferner kann die Ausgabe des Exports mit Hilfe mehrere Auswahlfelder gesteuert werden:

- Tabellarische Ausgabe
- Konvertierung griechischer Buchstaben in LaTeX-Kommandos
- Getrennte Speicherung von Plätzen, Parametern und Transitionen
- Markierung konstanter Plätze durch **(const.)**
- Hervorhebung der selektierten Plätze und Parameter durch Fettschreibung
- Sortierte Ausgabe
- Einrahmung von Platz- und Parameternamen in eckigen Klammern

A.2 Simulation

Die Simulation des vorliegenden kontinuierlichen Petrinetzes erfolgt in zwei Phasen. In der ersten Phase wird das Netz auf syntaktische Korrektheit geprüft. Dabei wird unter anderem sichergestellt, dass alle in einer Transition verwendeten Variablenbezeichner auch über eingehende Kanten mit gleichnamigen Plätzen verbunden sind oder ein Parameter mit dem entsprechenden Bezeichner existiert. Andernfalls wird eine Fehlermeldung mit der Angabe der betroffenen Transition und dem nicht verbundenen Bezeichner ausgegeben. In dieser Phase werden auch die kompletten internen Differentialgleichungen für jeden Platz konstruiert. Nach Abschluss dieser vorbereitenden Phase kann die Simulation mit den Voreinstellungen gestartet werden, oder es können mittels eines Optionendialogs folgende Einstellungen vorgenommen werden:

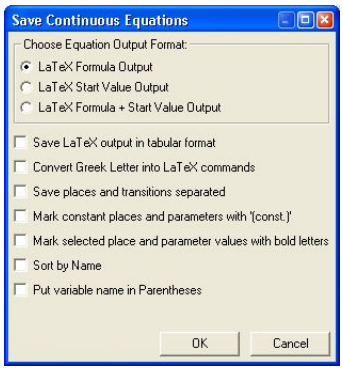
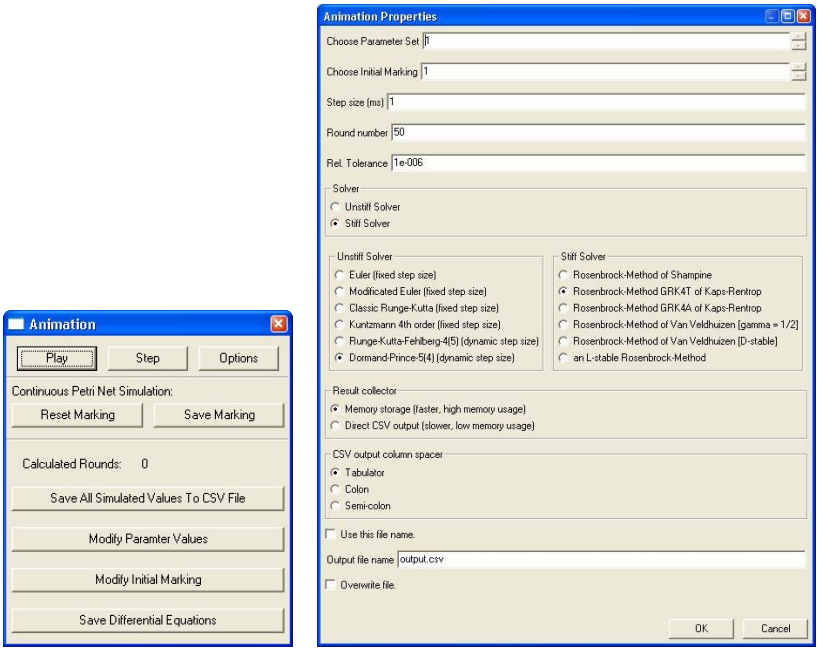


Abbildung A.3: Snoopy: Auswahldialog für Export in LaTeX



(a) Steuerungsdialog (b) Optionendialog

Abbildung A.4: Snoopy: Simulation

- Anfangsmarkierung
- Parametersatz
- Schrittweite
- Rundenanzahl
- Fehlertoleranz der Schrittweitensteuerung
- numerisches Lösungsverfahren
- Simulationsgeschwindigkeit in Millisekunden
- Auswertungsverfahren

An numerischen Lösungsverfahren stehen

- Explizite m-stufige Runge-Kutta-Verfahren
  - mit fester Schrittweite
    1. Euler-Verfahren
    2. Modifiziertes Euler-Verfahren
    3. Klassisches Runge-Kutta-Verfahren
    4. Verfahren von Kuntzmann 4. Ordnung
  - mit Schrittweitensteuerung
    5. Dynamisches Runge-Kutta-Fehlberg-Verfahren
    6. Dynamisches Dormand-Prince-Verfahren
- Rosenbrock-Verfahren für steife Differentialgleichungssysteme
  1. Koeffizienten von Shampine
  2. Koeffizienten GRK4T von Kaps und Rentrop
  3. Koeffizienten GRK4A von Kaps und Rentrop
  4. Koeffizienten von Van Veldhuizen
  5. Koeffizienten von Van Veldhuizen
  6. Koeffizienten für eine L-stabile Methode

zur Verfügung.

A.3 Auswertung der Simulationsergebnisse

Für die Auswertung der Simulationsergebnisse stehen zwei Varianten bereit. Zum einen existiert eine Speicherfunktion der berechneten Werte im CSV-Format. Dabei kann zwischen drei Trennzeichen gewählt werden. Dies sind das Tabulatorzeichen, welches die Standardeinstellung ist, das Komma und das Semikolon. Zusätzlich können, mittels eines in Abschnitt A.1.1 beschriebenen Plots, die berechneten Ergebnisse parallel zur Berechnung graphisch dargestellt werden.

Anhang B

Komplette kontinuierliche Petrinetze und Simulationsergebnisse der Beispielanwendungen

In diesem Anhang finden sich die kompletten kontinuierlichen Petrinetze der Beispielanwendungen sowie Auswertungen der durchgeführten Simulationsläufe zum Vergleich der implementierten Algorithmen. Die Simulationen wurden dabei mit den Standardwerten für relative Toleranz (10^{-6} Markierungseinheiten) und Simulationsverzögerung (1 ms) durchgeführt. Ferner wurden alle Daten während der Simulation im Arbeitsspeicher des Simulationsrechners gespeichert und erst nach Abschluss des Simulationslaufs auf der Festplatte gesichert.

Aufgrund der Tatsache, dass alle fünf Beispiele der Klasse der steifen Differentialgleichungssysteme zuzuordnen sind, haben die Algorithmen für nicht-steife Differentialgleichungen sehr schlechte Laufzeiteigenschaften. Deshalb wurden die vier Runge-Kutta-Algorithmen mit fester Schrittweite nur bei einem Beispiel exemplarisch simuliert. Die beiden Runge-Kutta-Verfahren mit dynamischer Schrittweitensteuerung haben bei allen Beispielen eine Schrittweite kleiner 10^{-8} eingestellt, um die Genauigkeitsanforderungen zu erfüllen. Diese Schrittweite wird in der numerischen Mathematik allgemein als zeitlich inakzeptabel angesehen und der Schrittweitenalgorithmus bricht die Simulation mit einer entsprechenden Fehlermeldung ab. Dadurch konnten für diese beiden Algorithmen keine Simulationsergebnisse ermittelt werden. Im Laufe der Tests wurden die nicht-steifen Algorithmen durch Vergrößerung der relativen Toleranz testbar. Die Beispiele wurden jeweils so viele Runden, im Folgenden Zeitschritte genannt, simuliert, wie es in den betreffenden Artikeln angegeben ist bzw. bis ein aussagekräftiger Zustand erreicht wurde.

Wenn nicht anders angegeben, wird in allen Diagrammen die in Abbildung B.1 dargestellte Legende verwendet.

Die Simulationen wurden auf einem Testsystem am Lehrstuhl durchgeführt. Dazu wurde ein Zwei-Prozessoren-PC mit zwei AMD Opteron 248 Prozessoren mit 2,2 GHz und 4 GB RAM verwendet. Auf dem Rechner lief Windows XP Professional mit Service Pack 2.

Abbildung B.1: Legende

B.1 RKIP-Beispiel

Die Hauptstruktur ist in Abbildung 5.11 auf Seite 44 und die dazugehörigen Parameter sind in Abbildung 5.12 auf Seite 44 abgebildet. Für die Simulationen wurden folgende resultierenden Differentialgleichungen mit den angegebenen Markierungs- und Parameterwerten verwendet.

$$
\begin{array}{llllllll}
m_1 = 2.5 & k_1 = 0.53 & m_7 = 2.5 & k_7 = 0.0075 \\
m_2 = 2.5 & k_2 = 0.0072 & m_8 = 10^{-5} & k_8 = 0.071 \\
m_3 = 0 & k_3 = 0.625 & m_9 = 2.5 & k_9 = 0.92 \\
m_4 = 0 & k_4 = 0.00245 & m_{10} = 3 & k_{10} = 0.00122 \\
m_5 = 0 & k_5 = 0.0315 & m_{11} = 0 & k_{11} = 0.87 \\
m_6 = 0 & k_6 = 0.6 &
\end{array}
$$

$$
\begin{array}{lll}
\frac{dm_1}{dt} = r_2 + r_5 - r_1 & \frac{dm_5}{dt} = r_5 + r_7 - r_6 & \frac{dm_9}{dt} = r_4 + r_8 - r_3 \\
\frac{dm_2}{dt} = r_2 + r_{11} - r_1 & \frac{dm_6}{dt} = r_5 + r_{10} - r_9 & \frac{dm_{10}}{dt} = r_{10} + r_{11} - r_9 \\
\frac{dm_3}{dt} = r_1 + r_4 - r_2 - r_3 & \frac{dm_7}{dt} = r_7 + r_8 - r_6 & \frac{dm_{11}}{dt} = r_9 - r_{10} - r_{11} \\
\frac{dm_4}{dt} = r_3 - r_4 - r_5 & \frac{dm_8}{dt} = r_6 - r_7 - r_8 &
\end{array}
$$

$$
\begin{array}{lll}
r_1 = k_1 * m_1 * m_2 & r_5 = k_5 * m_4 & r_9 = k_9 * m_6 * m_{10} \\
r_2 = k_2 * m_3 & r_6 = k_6 * m_5 * m_7 & r_{10} = k_{10} * m_{11} \\
r_3 = k_3 * m_3 * m_9 & r_7 = k_7 * m_8 & r_{11} = k_{11} * m_{11} \\
r_4 = k_4 * m_4 & r_8 = k_8 * m_8 &
\end{array}
$$

$$
\begin{array}{lrcl}
r_1: & m_1 + m_2 & \rightarrow & m_3 \\
r_2: & m_3 & \rightarrow & m_2 + m_1 \\
r_3: & m_9 + m_3 & \rightarrow & m_4 \\
r_4: & m_4 & \rightarrow & m_3 + m_9 \\
r_5: & m_4 & \rightarrow & m_5 + m_6 + m_1 \\
r_6: & m_7 + m_5 & \rightarrow & m_8
\end{array}
\qquad
\begin{array}{lrcl}
r_7: & m_8 & \rightarrow & m_7 + m_5 \\
r_8: & m_8 & \rightarrow & m_9 + m_7 \\
r_9: & m_6 + m_{10} & \rightarrow & m_{11} \\
r_{10}: & m_{11} & \rightarrow & m_{10} + m_6 \\
r_{11}: & m_{11} & \rightarrow & m_2 + m_{10}
\end{array}
$$

Bei der Simulation dieses Beispieles wurden die in Tabelle B.1 dargestellten Berechnungszeiten benötigt.

Verfahren	Zeit	Schrittweite	Schritte
Copasi	1 s	1	50
Euler	79 s	0.01	5000
Modifizierter Euler	78 s	0.01	5000
Klassischer Runge-Kutta 4. Ordnung	79 s	0.01	5000
Kuntzmann 4. Ordnung	78 s	0.01	5000
Runge-Kutta-Fehlberg-4(5)	-	1	50
Dormand-Prince-5(4)	-	1	50
Rosenbrockmethode von Shampine	5 s	1	50
Rosenbrockmethode GRK4T von Kaps-Rentrop	5 s	1	50
Rosenbrockmethode GRK4A von Kaps-Rentrop	5 s	1	50
Rosenbrockmethode von Van Veldhuizen [$\gamma = \frac{1}{2}$]	6 s	1	50
Rosenbrockmethode von Van Veldhuizen [D-stabil]	4 s	1	50
eine L-stabile Rosenbrockmethode	6 s	1	50

Tabelle B.1: RKIP: Berechnungszeiten

Die Ergebnisse lassen sich am anschaulichsten graphisch darstellen. Dazu wurde für jede Variable ein eigenes Diagramm (Abb. B.4 und B.5) erstellt und darin für jedes Verfahren ein Kurvenzug eingetragen. Für die bessere Verdeutlichung der Rechengenauigkeit am Ende des Simulationslaufs wurden die absoluten und relativen Fehler in den Tabellen B.2 und B.3 aufgeführt und in Abbildung B.3 Ausschnittvergrößerungen vom Zeitschritt 50 erstellt. Aufgrund der zusätzlichen nicht-steifen numerischen Verfahren wird für dieses Beispiel die in Abbildung B.2 dargestellte Legende verwendet.

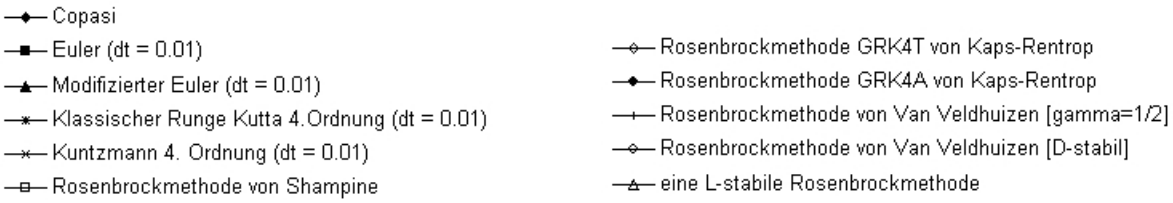


Abbildung B.2: RKIP: Legende

Verfahren	m_1	m_2	m_3
Copasi	0.36044	0.284049	0.544415
Euler	$-2.55118*10^{-6}$	$-1.86099*10^{-6}$	$2.20883*10^{-5}$
Modifizierter Euler	$1.14489*10^{-7}$	$-2.56486*10^{-7}$	$1.41271*10^{-8}$
Klassischer Runge-Kutta	$1.1131*10^{-7}$	$-2.58399*10^{-7}$	$4.04616*10^{-8}$
Kuntzmann	$3.89031*10^{-7}$	$-9.12388*10^{-8}$	$-2.2594*10^{-8}$
Shampine	$1.11174*10^{-7}$	$-2.58456*10^{-7}$	$4.14172*10^{-8}$
GRK4T von Kaps-Rentrop	$1.11383*10^{-7}$	$-2.58362*10^{-7}$	$3.96812*10^{-8}$
GRK4A von Kaps-Rentrop	$1.11293*10^{-7}$	$-2.58404*10^{-7}$	$4.0451*10^{-8}$
Van Veldhuizen [$\gamma=\frac{1}{2}$]	$1.11183*10^{-7}$	$-2.58449*10^{-7}$	$4.13151*10^{-8}$
Van Veldhuizen [D-stabil]	$1.11377*10^{-7}$	$-2.58365*10^{-7}$	$3.97315*10^{-8}$
eine L-stabile Methode	$1.11095*10^{-7}$	$-2.58496*10^{-7}$	$4.21624*10^{-8}$

Verfahren	m_4	m_5	m_6
Copasi	1.59514	0.0512542	0.0185923
Euler	$-1.45371*10^{-5}$	$2.53927*10^{-7}$	$-2.05171*10^{-7}$
Mod. Euler	$4.87138*10^{-6}$	$2.23369*10^{-8}$	$3.487*10^{-8}$
Klassischer Runge-Kutta	$4.84823*10^{-6}$	$2.26133*10^{-8}$	$3.45836*10^{-8}$
Kuntzmann	$6.87037*10^{-6}$	$-1.51626*10^{-9}$	$5.95931*10^{-8}$
Shampine	$4.84741*10^{-6}$	$2.25791*10^{-8}$	$3.45651*10^{-8}$
GRK4T von Kaps-Rentrop	$4.84893*10^{-6}$	$2.25998*10^{-8}$	$3.45903*10^{-8}$
GRK4A von Kaps-Rentrop	$4.84825*10^{-6}$	$2.25944*10^{-8}$	$3.45806*10^{-8}$
Van Veldhuizen [$\gamma=\frac{1}{2}$]	$4.8475*10^{-6}$	$2.25764*10^{-8}$	$3.45659*10^{-8}$
Van Veldhuizen [D-stabil]	$4.84889*10^{-6}$	$2.25968*10^{-8}$	$3.45884*10^{-8}$
eine L-stabile Methode	$4.84674*10^{-6}$	$2.25865*10^{-8}$	$3.4557*10^{-8}$

Verfahren	m_7	m_8	m_9
Copasi	1.80351	0.6965	0.157111
Euler	$-2.17525*10^{-5}$	$2.17525*10^{-5}$	$-2.66935*10^{-6}$
Modifizierter Euler	$1.85464*10^{-7}$	$-1.85463*10^{-7}$	$9.17428*10^{-8}$
Klassischer Runge-Kutta	$1.59287*10^{-7}$	$-1.59287*10^{-7}$	$8.84447*10^{-8}$
Kuntzmann	$2.445*10^{-6}$	$-2.445*10^{-6}$	$3.76142*10^{-7}$
Shampine	$1.57683*10^{-7}$	$-1.57683*10^{-7}$	$8.76939*10^{-8}$
GRK4T von Kaps-Rentrop	$1.60217*10^{-7}$	$-1.60217*10^{-7}$	$8.86865*10^{-8}$
GRK4A von Kaps-Rentrop	$1.59217*10^{-7}$	$-1.59217*10^{-7}$	$8.8351*10^{-8}$
Van Veldhuizen [$\gamma=\frac{1}{2}$]	$1.57722*10^{-7}$	$-1.57722*10^{-7}$	$8.76438*10^{-8}$
Van Veldhuizen [D-stabil]	$1.60128*10^{-7}$	$-1.60128*10^{-7}$	$8.86413*10^{-8}$
eine L-stabile Methode	$1.56745*10^{-7}$	$-1.56745*10^{-7}$	$8.74167*10^{-8}$

Verfahren	m_{10}	m_{11}
Copasi	2.9422	0.057799
Euler	$1.78502*10^{-6}$	$-7.85023*10^{-7}$
Modifizierter Euler	$9.63895*10^{-7}$	$3.61049*10^{-8}$
Klassischer Runge-Kutta	$9.64875*10^{-7}$	$3.51253*10^{-8}$
Kuntzmann	$8.79323*10^{-7}$	$1.20677*10^{-7}$
Shampine	$9.64936*10^{-7}$	$3.50639*10^{-8}$
GRK4T von Kaps-Rentrop	$9.64844*10^{-7}$	$3.51559*10^{-8}$
GRK4A von Kaps-Rentrop	$9.64883*10^{-7}$	$3.51168*10^{-8}$
Van Veldhuizen [$\gamma=\frac{1}{2}$]	$9.64934*10^{-7}$	$3.50662*10^{-8}$
Van Veldhuizen [D-stabil]	$9.64847*10^{-7}$	$3.51534*10^{-8}$
eine L-stabile Methode	$9.64967*10^{-7}$	$3.50326*10^{-8}$

Tabelle B.2: RKIP: Absoluter Fehler bei Zeitschritt 50

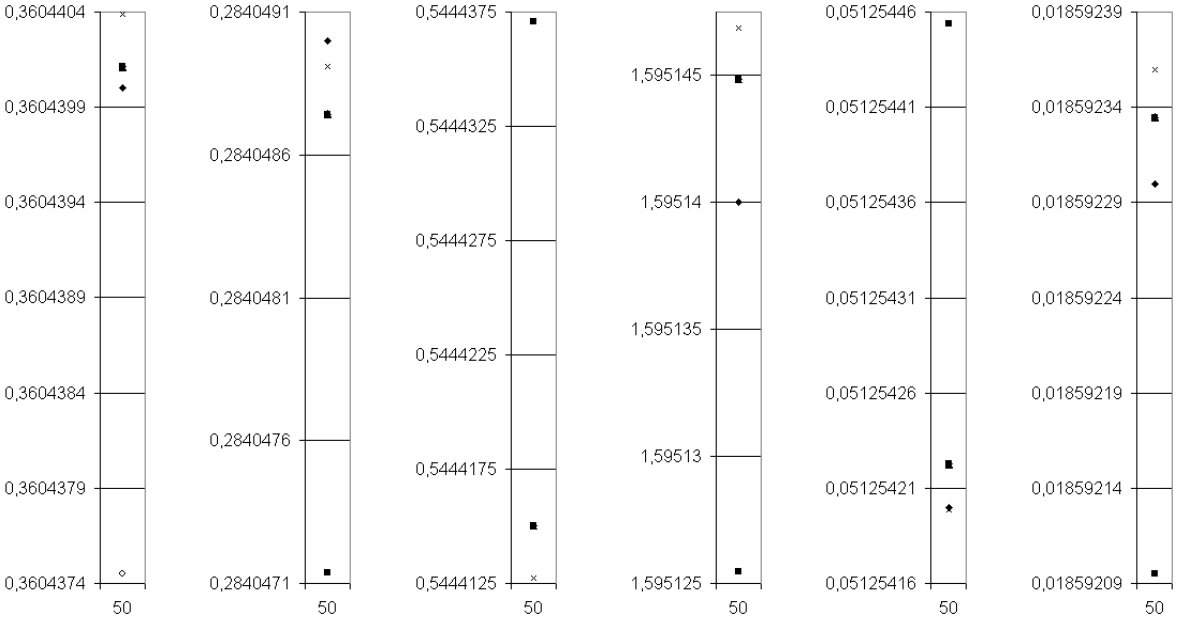
Verfahren	m_1	m_2	m_3
Euler	$-7.07796 * 10^{-6}$	$-6.55164 * 10^{-6}$	$4.05725 * 10^{-5}$
Modifizierter Euler	$3.17637 * 10^{-7}$	$-9.02963 * 10^{-7}$	$2.59491 * 10^{-8}$
Klassischer Runge-Kutta	$3.08816 * 10^{-7}$	$-9.097 * 10^{-7}$	$7.43213 * 10^{-8}$
Kuntzmann	$1.07932 * 10^{-7}$	$-3.21208 * 10^{-7}$	$-4.15015 * 10^{-6}$
Shampine	$3.0844 * 10^{-7}$	$-9.099 * 10^{-7}$	$7.60766 * 10^{-8}$
GRK4T von Kaps-Rentrop	$3.09019 * 10^{-7}$	$-9.09567 * 10^{-7}$	$7.28878 * 10^{-8}$
GRK4A von Kaps-Rentrop	$3.0877 * 10^{-7}$	$-9.09718 * 10^{-7}$	$7.43017 * 10^{-8}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$3.08464 * 10^{-7}$	$-9.09875 * 10^{-7}$	$7.5889 * 10^{-8}$
Van Veldhuizen [D-stabil]	$3.09003 * 10^{-7}$	$-9.09578 * 10^{-7}$	$7.29802 * 10^{-8}$
eine L-stabile Methode	$3.0822 * 10^{-7}$	$-9.10041 * 10^{-7}$	$7.74453 * 10^{-8}$

Verfahren	m_4	m_5	m_6
Euler	$-9.11338 * 10^{-6}$	$4.95427 * 10^{-6}$	$-1.10353 * 10^{-5}$
Mod. Euler	$3.05389 * 10^{-6}$	$4.35807 * 10^{-7}$	$1.87551 * 10^{-6}$
Klassischer Runge-Kutta	$3.03938 * 10^{-6}$	$4.41199 * 10^{-7}$	$1.8601 * 10^{-6}$
Kuntzmann	$4.30707 * 10^{-6}$	$-2.95831 * 10^{-8}$	$3.20526 * 10^{-6}$
Shampine	$3.03886 * 10^{-6}$	$4.40533 * 10^{-7}$	$1.85911 * 10^{-6}$
GRK4T von Kaps-Rentrop	$3.03982 * 10^{-6}$	$4.40935 * 10^{-7}$	$1.86046 * 10^{-6}$
GRK4A von Kaps-Rentrop	$3.03939 * 10^{-6}$	$4.40829 * 10^{-7}$	$1.85994 * 10^{-6}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	3.0389210^{-6}	$4.4048 * 10^{-7}$	$1.85915 * 10^{-6}$
Van Veldhuizen [D-stabil]	$3.03979 * 10^{-6}$	$4.40877 * 10^{-7}$	$1.86036 * 10^{-6}$
eine L-stabile Methode	$3.03844 * 10^{-6}$	$4.40677 * 10^{-7}$	$1.85867 * 10^{-6}$

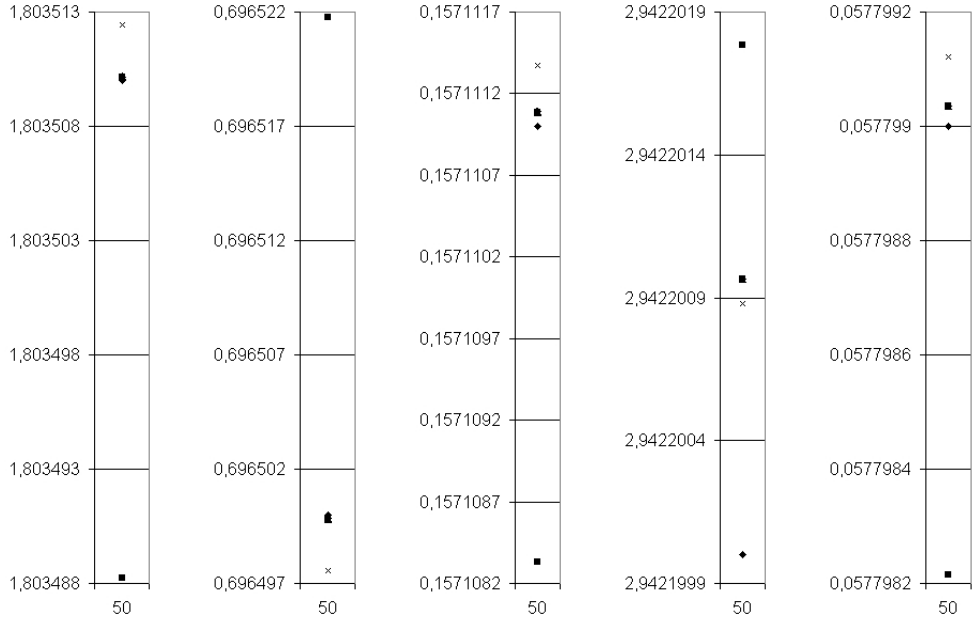
Verfahren	m_7	m_8	m_9
Euler	$-1.20612 * 10^{-5}$	$3.12312 * 10^{-5}$	$-1.66902 * 10^{-5}$
Modifizierter Euler	$1.02835 * 10^{-7}$	$-2.66279 * 10^{-7}$	$5.83936 * 10^{-7}$
Klassischer Runge-Kutta	$8.83206 * 10^{-8}$	$-2.28696 * 10^{-7}$	$5.62944 * 10^{-7}$
Kuntzmann	$1.35569 * 10^{-6}$	$-3.51041 * 10^{-6}$	$2.29411 * 10^{-6}$
Shampine	$8.7431 * 10^{-8}$	$-2.26393 * 10^{-7}$	$5.58165 * 10^{-7}$
GRK4T von Kaps-Rentrop	$8.88364 * 10^{-8}$	$-2.30032 * 10^{-7}$	$5.64483 * 10^{-7}$
GRK4A von Kaps-Rentrop	$8.82731 * 10^{-8}$	$-2.28574 * 10^{-7}$	$5.62348 * 10^{-7}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$8.7453 * 10^{-8}$	$-2.2645 * 10^{-7}$	$5.57847 * 10^{-7}$
Van Veldhuizen [D-stabil]	$8.87867 * 10^{-8}$	$-2.29903 * 10^{-7}$	$5.64195 * 10^{-7}$
eine L-stabile Methode	$8.69109 * 10^{-8}$	$-2.25046 * 10^{-7}$	$5.56401 * 10^{-7}$

Verfahren	m_{10}	m_{11}
Euler	$6.06697 * 10^{-6}$	$-1.35819 * 10^{-5}$
Modifizierter Euler	$3.2761 * 10^{-7}$	$6.24663 * 10^{-7}$
Klassischer Runge-Kutta	$3.27943 * 10^{-7}$	$6.07715 * 10^{-7}$
Kuntzmann	$2.98866 * 10^{-7}$	$2.08787 * 10^{-6}$
Shampine	$3.27964 * 10^{-7}$	$6.06652 * 10^{-7}$
GRK4T von Kaps-Rentrop	$3.27933 * 10^{-7}$	$6.08244 * 10^{-7}$
GRK4A von Kaps-Rentrop	$3.27946 * 10^{-7}$	$6.07567 * 10^{-7}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$3.27963 * 10^{-7}$	$6.06691 * 10^{-7}$
Van Veldhuizen [D-stabil]	$3.27934 * 10^{-7}$	$6.08201 * 10^{-7}$
eine L-stabile Methode	$3.27975 * 10^{-7}$	$6.06111 * 10^{-7}$

Tabelle B.3: RKIP: Relativer Fehler bei Zeitschritt 50

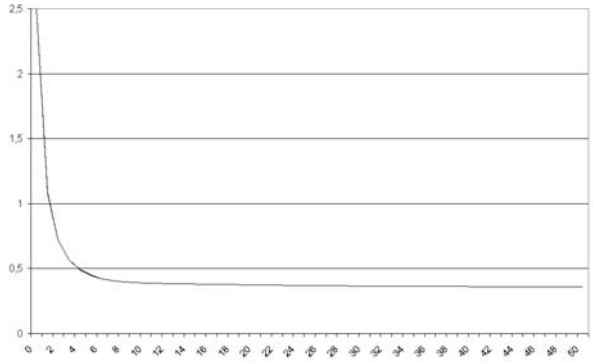


(a) m_1 (b) m_2 (c) m_3 (d) m_4 (e) m_5 (f) m_6

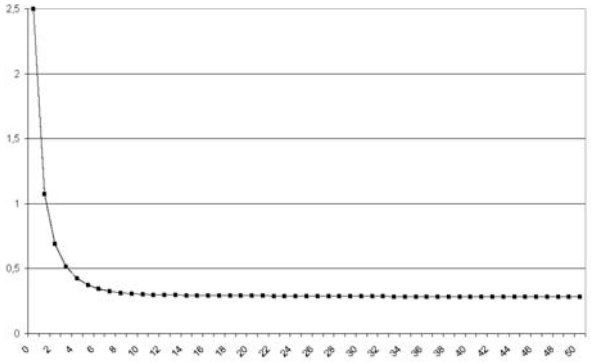


(g) m_7 (h) m_8 (i) m_9 (j) m_{10} (k) m_{11}

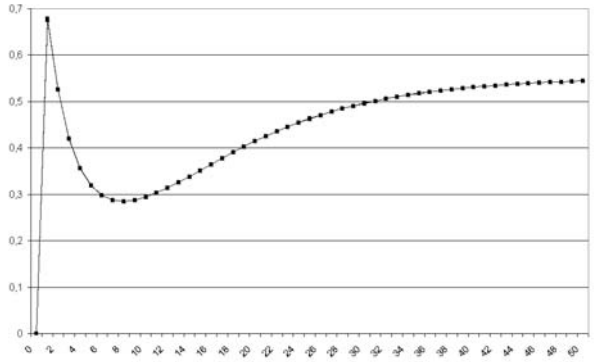
Abbildung B.3: RKIP: Absoluter Fehler bei Zeitschritt 50



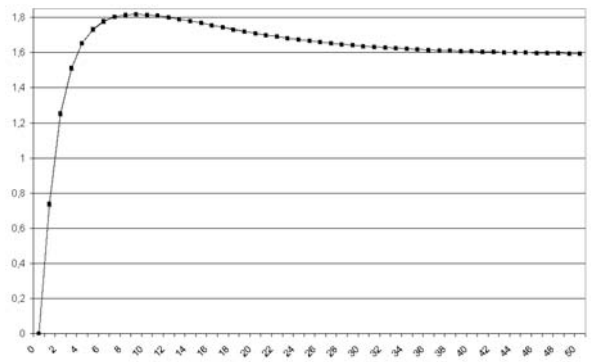
(a) m_1



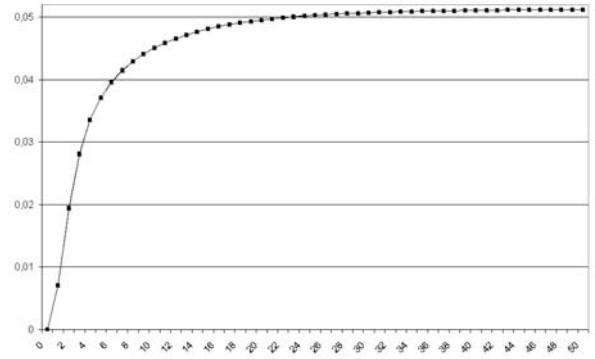
(b) m_2



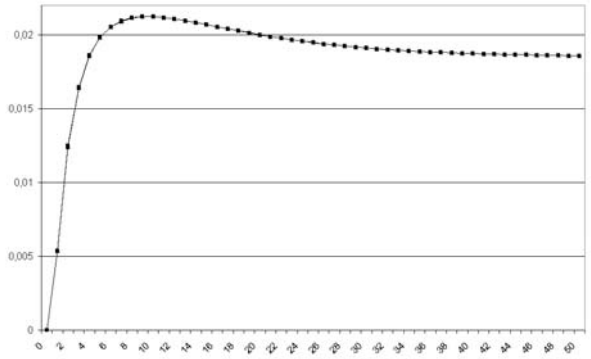
(c) m_3



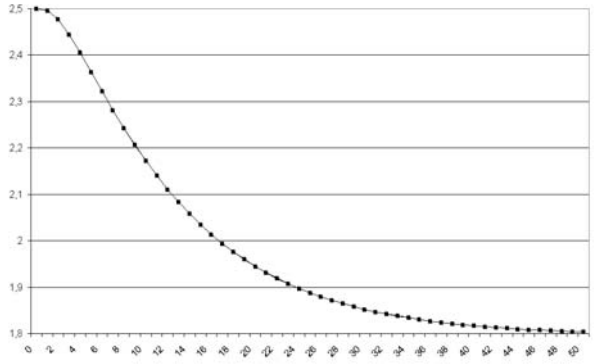
(d) m_4



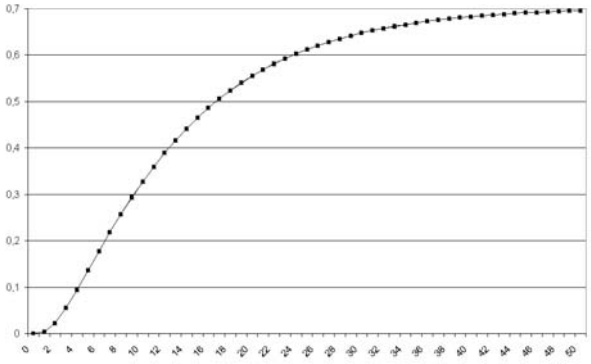
(e) m_5



(f) m_6

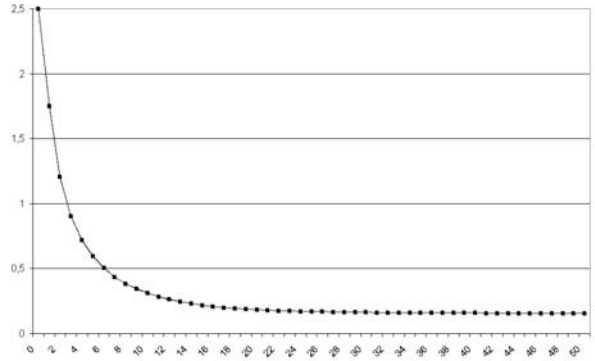


(g) m_7

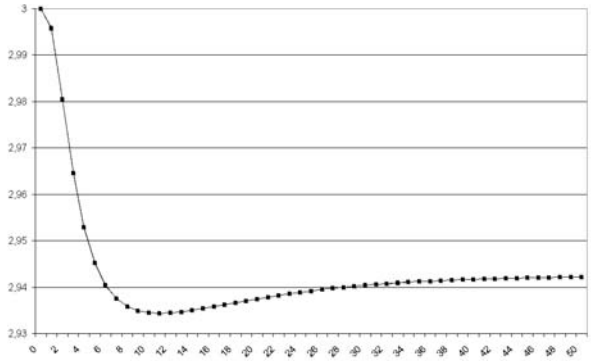


(h) m_8

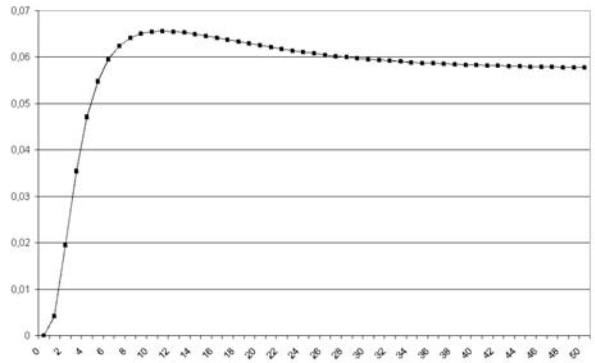
Abbildung B.4: RKIP: Kurvenverlaufsdiagramme



(a) m_9



(b) m_{10}



(c) m_{11}

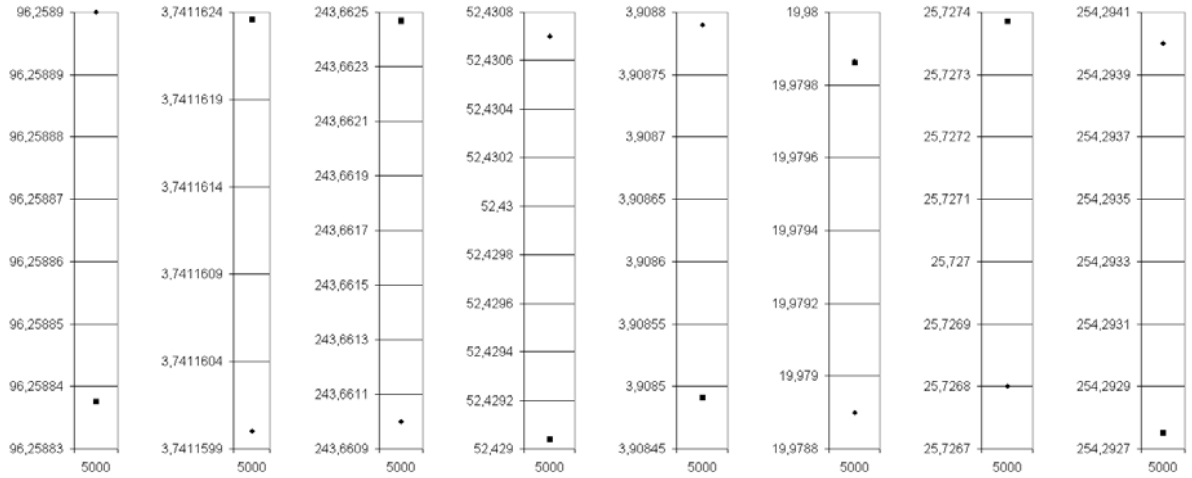
Abbildung B.5: RKIP: Kurvenverlaufsdiagramme (Forts.)

B.2 MAPK-Kaskaden-Beispiel von Kholodenko

Die Hauptstruktur ist in Abbildung 5.14 auf Seite 45 und die dazugehörigen Parameter sind in Abbildung 5.13 auf Seite 44 abgebildet. Zusätzlich zu den Simulationsergebnissen folgen die verwendeten Differentialgleichungssysteme mit den Parameter- und Markierungswerten.

Verfahren	Zeit	Schrittweite	Schritte
Copasi	1 s	1	5000
Rosenbrockmethode von Shampine	126 s	1	5000
Rosenbrockmethode GRK4T von Kaps-Rentrop	91 s	1	5000
Rosenbrockmethode GRK4A von Kaps-Rentrop	111 s	1	5000
Rosenbrockmethode von Van Veldhuizen [$\gamma = \frac{1}{2}$]	127 s	1	5000
Rosenbrockmethode von Van Veldhuizen [D-stabil]	91 s	1	5000
eine L-stabile Rosenbrockmethode	130 s	1	5000

Tabelle B.4: MAPK-Kaskade: Berechnungszeiten



(a) MKKK (b) $MKKK_P$ (c) MKK (d) MKK_P (e) MKK_{PP} (f) MAPK (g) $MAPK_P$ (h) $MAPK_{PP}$

Abbildung B.6: MAPK-Kaskade: Absoluter Fehler bei Zeitschritt 5000

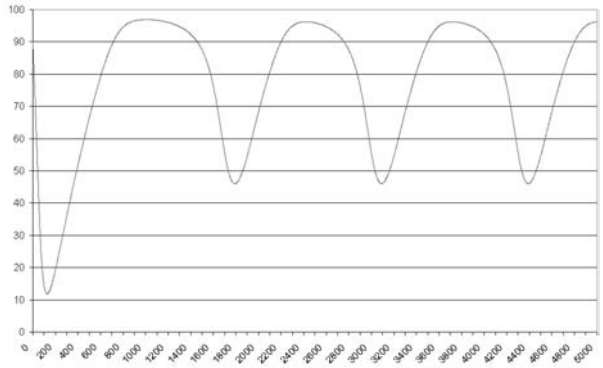
MKK	= 280	V_1	= 2.5	k_4	= 0.025	k_8	= 0.025
MKK_P	= 10	K_1	= 9	K_4	= 15	K_8	= 15.0
MKK_{PP}	= 10	K_{1_1}	= 10.0	V_5	= 0.75	V_9	= 0.5
MKKK	= 90	n	= 1.0	K_5	= 15	K_9	= 15.0
$MKKK_P$	= 10	V_2	= 0.25	V_6	= 0.75	V_{10}	= 0.5
MAPK	= 280	K_2	= 8	K_6	= 15	K_{10}	= 15.0
$MAPK_P$	= 10	k_3	= 0.025	k_7	= 0.025		
$MAPK_{PP}$	= 10	K_3	= 15.0	K_7	= 15		

Verfahren	MKKK	$MKKK_P$	MKK
Copasi	96.2589	3.74116	243.661
Shampine	$-6.23576*10^{-5}$	$2.35763*10^{-6}$	$1.46919*10^{-3}$
GRK4T von Kaps-Rentrop	$-6.23576*10^{-5}$	$2.35756*10^{-6}$	$1.46957*10^{-3}$
GRK4A von Kaps-Rentrop	$-6.23569*10^{-5}$	$2.35688*10^{-6}$	$1.46952*10^{-3}$
Van Veldhuizen [$\gamma=\frac{1}{2}$]	$-6.23579*10^{-5}$	$2.35789*10^{-6}$	$1.46913*10^{-3}$
Van Veldhuizen [D-stabil]	$-6.23576*10^{-5}$	$2.35757*10^{-6}$	$1.46957*10^{-3}$
eine L-stabile Methode	$-6.23581*10^{-5}$	$2.3581*10^{-6}$	$1.46886*10^{-3}$

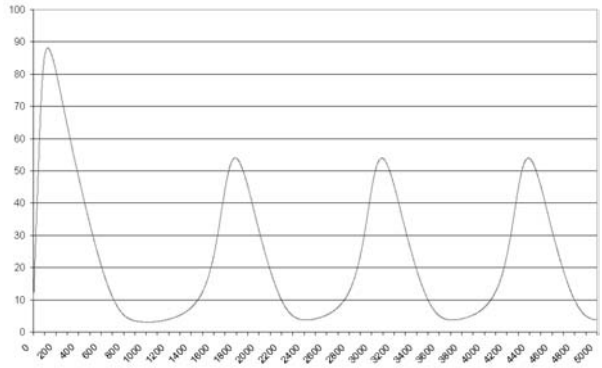
Verfahren	MKK_P	MKK_{PP}	MAPK
Copasi	52.4307	3.90879	19.9789
Shampine	$-1.66011*10^{-3}$	$-2.9908*10^{-4}$	$9.6195*10^{-4}$
GRK4T von Kaps-Rentrop	$-1.66046*10^{-3}$	$-2.9911*10^{-4}$	$9.6219*10^{-4}$
GRK4A von Kaps-Rentrop	$-1.66043*10^{-3}$	$-2.9909*10^{-4}$	$9.6213*10^{-4}$
Van Veldhuizen [$\gamma=\frac{1}{2}$]	$-1.66005*10^{-3}$	$-2.9908*10^{-4}$	$9.6192*10^{-4}$
Van Veldhuizen [D-stabil]	$-1.66046*10^{-3}$	$-2.9912*10^{-4}$	$9.622*10^{-4}$
eine L-stabile Methode	$-1.65981*10^{-3}$	$-2.9906*10^{-4}$	$9.6278*10^{-4}$

Verfahren	$MAPK_P$	$MAPK_{PP}$
Copasi	25.7268	254.294
Shampine	$5.857*10^{-4}$	$-1.24766*10^{-3}$
GRK4T von Kaps-Rentrop	$5.8579*10^{-4}$	$-1.24799*10^{-3}$
GRK4A von Kaps-Rentrop	$5.8579*10^{-4}$	$-1.24792*10^{-3}$
Van Veldhuizen [$\gamma=\frac{1}{2}$]	$5.8568*10^{-4}$	$-1.2476*10^{-3}$
Van Veldhuizen [D-stabil]	$5.8579*10^{-4}$	$-1.24799*10^{-3}$
eine L-stabile Methode	$5.8561*10^{-4}$	$-1.24739*10^{-3}$

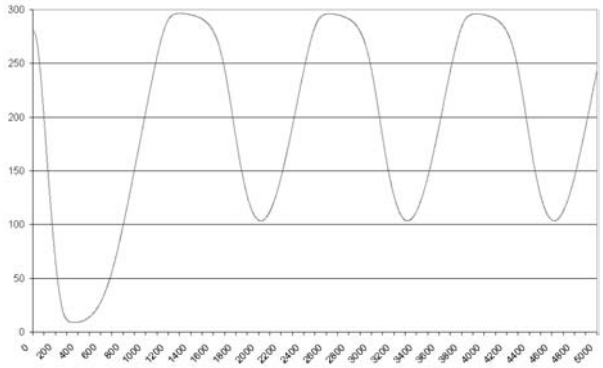
Tabelle B.5: MAPK-Kaskade: Absoluter Fehler bei Zeitschritt 5000



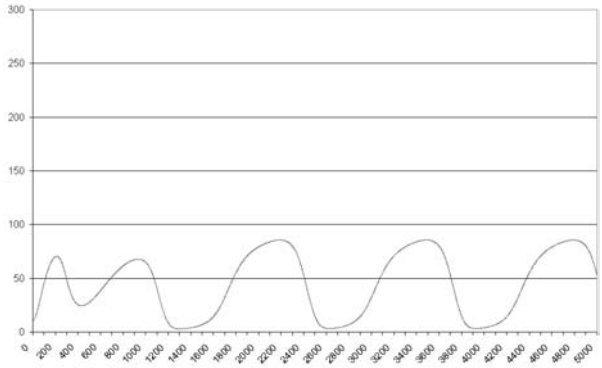
(a) MKKK



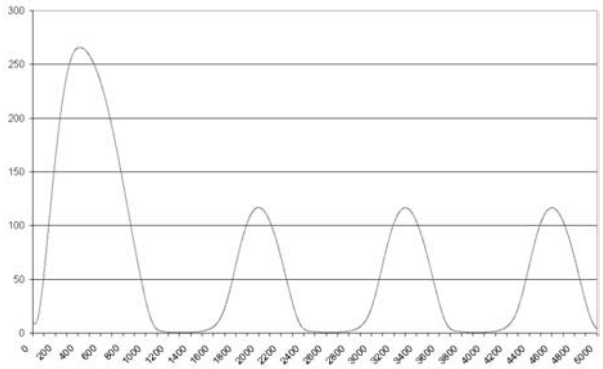
(b) $MKKK_P$



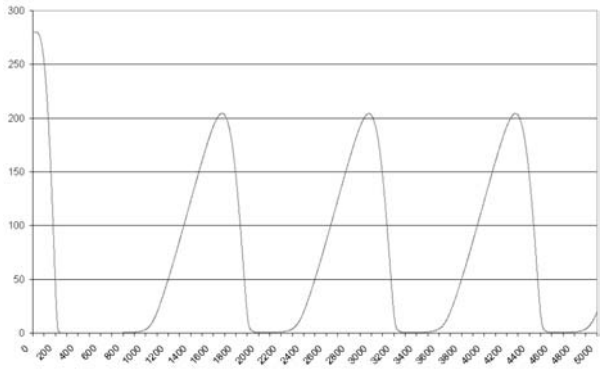
(c) MKK



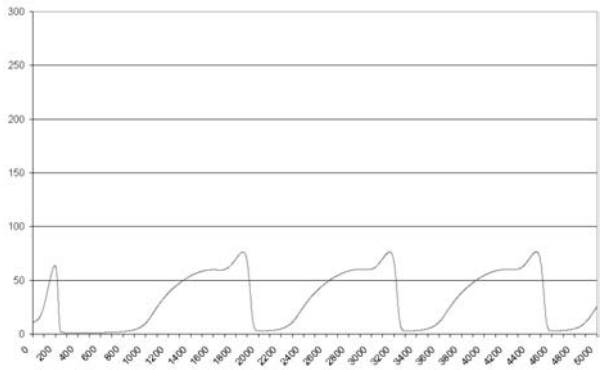
(d) MKK_P



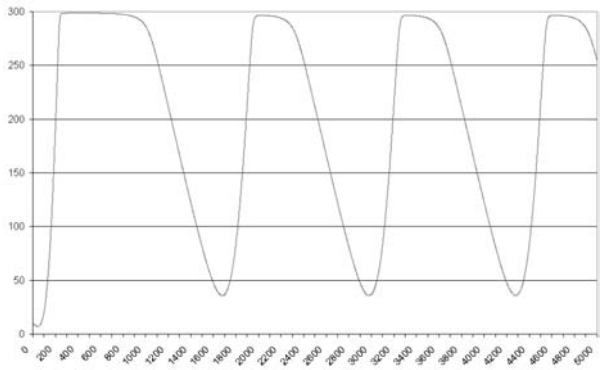
(e) MKK_{PP}



(f) MAPK



(g) $MAPK_P$



(h) $MAPK_{PP}$

Abbildung B.7: MAPK-Kaskade: Kurvenverlaufsdiagramme

Verfahren	MKKK	$MKKK_P$	MKK
Shampine	$-6.47812*10^{-7}$	$6.30186*10^{-7}$	$6.0297*10^{-6}$
GRK4T von Kaps-Rentrop	$-6.47811*10^{-7}$	$6.30168*10^{-7}$	$6.0312*10^{-6}$
GRK4A von Kaps-Rentrop	$-6.47804*10^{-7}$	$6.29987*10^{-7}$	$6.031*10^{-6}$
Van Veldhuizen [$\gamma=\frac{1}{2}$]	$-6.47814*10^{-7}$	$6.30256*10^{-7}$	$6.0294*10^{-6}$
Van Veldhuizen [D-stabil]	$-6.47811*10^{-7}$	$6.30171*10^{-7}$	$6.0312*10^{-6}$
eine L-stabile Methode	$-6.47816*10^{-7}$	$6.30313*10^{-7}$	$6.0283*10^{-6}$

Verfahren	MKK_P	MKK_{PP}	MAPK
Shampine	$-3.1663*10^{-5}$	$-7.6516*10^{-5}$	$4.81485*10^{-5}$
GRK4T von Kaps-Rentrop	$-3.167*10^{-5}$	$-7.6523*10^{-5}$	$4.81605*10^{-5}$
GRK4A von Kaps-Rentrop	$-3.1669*10^{-5}$	$-7.6517*10^{-5}$	$4.81573*10^{-5}$
Van Veldhuizen [$\gamma=\frac{1}{2}$]	$-3.1662*10^{-5}$	$-7.6514*10^{-5}$	$4.81466*10^{-5}$
Van Veldhuizen [D-stabil]	$-3.167*10^{-5}$	$-7.6524*10^{-5}$	$4.81606*10^{-5}$
eine L-stabile Methode	$-3.1657*10^{-5}$	$-7.6509*10^{-5}$	$4.81398*10^{-5}$

Verfahren	$MAPK_P$	$MAPK_{PP}$
Shampine	$2.2766*10^{-5}$	$-4.9064*10^{-6}$
GRK4T von Kaps-Rentrop	$2.277*10^{-5}$	$-4.9077*10^{-6}$
GRK4A von Kaps-Rentrop	$2.277*10^{-5}$	$-4.9074*10^{-6}$
Van Veldhuizen [$\gamma=\frac{1}{2}$]	$2.2765*10^{-5}$	$-4.9061*10^{-6}$
Van Veldhuizen [D-stabil]	$2.277*10^{-5}$	$-4.9077*10^{-6}$
eine L-stabile Methode	$2.2763*10^{-5}$	$-4.9053*10^{-6}$

Tabelle B.6: MAPK-Kaskade: Relativer Fehler bei Zeitschritt 5000

$$\frac{d\mathrm{MKK}}{d\mathrm{t}} = \mathrm{b_4} - \mathrm{b_1} \qquad \frac{d\mathrm{MKKK_P}}{d\mathrm{t}} = \mathrm{a_1} - \mathrm{a_2}$$
$$\frac{d\mathrm{MKK_P}}{d\mathrm{t}} = \mathrm{b_1} + \mathrm{b_3} - \mathrm{b_4} - \mathrm{b_2} \qquad \frac{d\mathrm{MAPK}}{d\mathrm{t}} = \mathrm{c_4} - \mathrm{c_1}$$
$$\frac{d\mathrm{MKK_{PP}}}{d\mathrm{t}} = \mathrm{b_2} - \mathrm{b_3} \qquad \frac{d\mathrm{MAPK_P}}{d\mathrm{t}} = \mathrm{c_1} + \mathrm{c_3} - \mathrm{c_2} - \mathrm{c_4}$$
$$\frac{d\mathrm{MKKK}}{d\mathrm{t}} = \mathrm{a_2} - \mathrm{a_1} \qquad \frac{d\mathrm{MAPK_{PP}}}{d\mathrm{t}} = \mathrm{c_2} - \mathrm{c_3}$$

$$\mathrm{a_1} = \frac{\mathrm{V_1} * \mathrm{MKKK}}{\left(1 + \left(\frac{\mathrm{MAPK_{PP}}}{\mathrm{K_1}}\right)^{\mathrm{n}}\right) * (\mathrm{K_{1_1}} + \mathrm{MKKK})} \qquad \mathrm{b_4} = \frac{\mathrm{V_6} * \mathrm{MKK_P}}{\mathrm{K_6} + \mathrm{MKK_P}}$$
$$\mathrm{a_2} = \frac{\mathrm{V_2} * \mathrm{MKKK_P}}{\mathrm{K_2} + \mathrm{MKKK_P}} \qquad \mathrm{c_1} = \frac{\mathrm{k_7} * \mathrm{MKK_{PP}} * \mathrm{MAPK}}{\mathrm{K_7} + \mathrm{MAPK}}$$
$$\mathrm{b_1} = \frac{\mathrm{k_3} * \mathrm{MKKK_P} * \mathrm{MKK}}{\mathrm{K_3} + \mathrm{MKK}} \qquad \mathrm{c_2} = \frac{\mathrm{k_8} * \mathrm{MKK_{PP}} * \mathrm{MAPK_P}}{\mathrm{K_8} + \mathrm{MAPK_P}}$$
$$\mathrm{b_2} = \frac{\mathrm{k_4} * \mathrm{MKKK_P} * \mathrm{MKK_P}}{\mathrm{K_4} + \mathrm{MKK_P}} \qquad \mathrm{c_3} = \frac{\mathrm{V_9} * \mathrm{MAPK_{PP}}}{\mathrm{K_9} + \mathrm{MAPK_{PP}}}$$
$$\mathrm{b_3} = \frac{\mathrm{V_5} * \mathrm{MKK_{PP}}}{\mathrm{K_5} + \mathrm{MKK_{PP}}} \qquad \mathrm{c_4} = \frac{\mathrm{V_{10}} * \mathrm{MAPK_P}}{\mathrm{K_{10}} + \mathrm{MAPK_P}}$$

a_1 : MKKK $\rightarrow$ $MKKK_P$
a_2 : $MKKK_P$ $\rightarrow$ MKKK
b_1 : MKK $\rightarrow$ MKK_P
b_2 : MKK_P $\rightarrow$ MKK_{PP}
b_3 : MKK_{PP} $\rightarrow$ MKK_P
b_4 : MKK_P $\rightarrow$ MKK
c_1 : MAPK $\rightarrow$ $MAPK_P$
c_2 : $MAPK_P$ $\rightarrow$ $MAPK_{PP}$
c_3 : $MAPK_{PP}$ $\rightarrow$ $MAPK_P$
c_4 : $MAPK_P$ $\rightarrow$ MAPK

B.3 Leber-Beispiel von Reich

In der folgenden Abbildung B.11 sind die Unterstrukturen zum als Petrinetz umgesetzten Leber-Beispiel [KSLR00] dargestellt. Die Hauptstruktur ist in Abbildung 5.15 auf Seite 45 und die dazugehörigen Parameter in Abbildung 5.16 auf Seite 45 abgebildet. Außerdem folgen die Differentialgleichungen und die verwendeten Startwerte.

Verfahren	Zeit	Schrittweite	Schritte
Copasi	1 s	1	15
Rosenbrockmethode von Shampine	22 s	1	15
Rosenbrockmethode GRK4T von Kaps-Rentrop	15 s	1	15
Rosenbrockmethode GRK4A von Kaps-Rentrop	18 s	1	15
Rosenbrockmethode von Van Veldhuizen [$\gamma = \frac{1}{2}$]	22 s	1	15
Rosenbrockmethode von Van Veldhuizen [D-stabil]	16 s	1	15
eine L-stabile Rosenbrockmethode	23 s	1	15

Tabelle B.7: Leber: Berechnungszeiten

$CHYL = 1.65$
$CR = 10$
$DISC = 7.2$
$HDL2 = 25$
$HDL3 = 2.4$
$IDL = 2$
$LL = 2.2$
$LDLR = 0$
$LRP = 0$
$VLDL = 0.46$
$K_{11} = 2.0$
$K_{22} = 0.04$
$K_{33} = 4.0$
$K_{43} = 8.0$
$K_{45} = 2.5$
$K_{46} = 10.0$
$K_{47} = 0.3$
$K_{48} = 7.0$
$K_{56} = 200.0$
$K_{68} = 2.8$
$K_{74} = 0.05$
$K_{75} = 30.0$
$K_{76} = 2.0$
$K_{77} = 0.07$
$K_{78} = 0.4$
$K_{87} = 0.007$
$K_{88} = 3.0$
$K_{97} = 30.0$
$K_a = 0.0$
$k_{Chol} = 1.5$
$k_i = 2.5$
$\lambda_1 = 0.05$
$\lambda_2 = 0.1$
$\lambda_3 = 0.006$
$\lambda_4 = 0.05$
$\lambda_5 = 0.25$
$\lambda_6 = 2$
$V_{CETP} = 10$
$V_{HL} = 1.0$
$V_{LCAT} = 1.4$
$V_{LDLR} = 5$
$V_{LPL} = 25$
$V_{LRP} = 4.0$
$V_{SynCHYL} = 5.0$
$V_{SynDISC} = 1.4$
$V_{SynVLDL} = 20.0$

Verfahren	CHYL	CR	DISC
Copasi	1.94669	11.2005	39.1136
Shampine	$-2.28917*10^{-6}$	$-4.38141*10^{-5}$	$-6.08713*10^{-6}$
GRK4T von Kaps-Rentrop	$-2.28299*10^{-6}$	$-4.38138*10^{-5}$	$-6.09152*10^{-6}$
GRK4A von Kaps-Rentrop	$-2.2854*10^{-6}$	$-4.38137*10^{-5}$	$-6.08936*10^{-6}$
Van Veldhuizen [$\gamma=\frac{1}{2}$]	$-2.28411*10^{-6}$	$-4.38145*10^{-5}$	$-6.09141*10^{-6}$
Van Veldhuizen [D-stabil]	$-2.283*10^{-6}$	$-4.38139*10^{-5}$	$-6.09141*10^{-6}$
eine L-stabile Methode	$-2.28666*10^{-6}$	$-4.38145*10^{-5}$	$-6.08626*10^{-6}$

Verfahren	HDL2	HDL3	IDL
Copasi	-0.133216	3.32747	2.63598
Shampine	$1.25791*10^{-5}$	$-4.1989*10^{-6}$	$3.0682*10^{-6}$
GRK4T von Kaps-Rentrop	$1.25781*10^{-5}$	$-4.20196*10^{-6}$	$3.06336*10^{-6}$
GRK4A von Kaps-Rentrop	$1.25802*10^{-5}$	$-4.20652*10^{-6}$	$3.06715*10^{-6}$
Van Veldhuizen [$\gamma=\frac{1}{2}$]	$1.2571*10^{-5}$	$-4.18553*10^{-6}$	$3.06096*10^{-6}$
Van Veldhuizen [D-stabil]	$1.25787*10^{-5}$	$-4.20166*10^{-6}$	$3.06332*10^{-6}$
eine L-stabile Methode	$1.25777*10^{-5}$	$-4.19472*10^{-6}$	$3.06869*10^{-6}$

Verfahren	LDL	LDLR	LRP
Copasi	2.97415	5.71154	-1.22602
Shampine	$-3.45167*10^{-6}$	$-1.22055*10^{-7}$	$-3.54626*10^{-6}$
GRK4T von Kaps-Rentrop	$-3.44922*10^{-6}$	$-1.20753*10^{-7}$	$-3.54648*10^{-6}$
GRK4A von Kaps-Rentrop	$-3.45236*10^{-6}$	$-1.20803*10^{-7}$	$-3.54664*10^{-6}$
Van Veldhuizen [$\gamma=\frac{1}{2}$]	$-3.44597*10^{-6}$	$-1.21662*10^{-7}$	$-3.54587*10^{-6}$
Van Veldhuizen [D-stabil]	$-3.44916*10^{-6}$	$-1.20807*10^{-7}$	$-3.54639*10^{-6}$
eine L-stabile Methode	$-3.45145*10^{-6}$	$-1.22505*10^{-7}$	$-3.54584*10^{-6}$

Verfahren	VLDL
Copasi	0.529106
Shampine	$5.25507*10^{-7}$
GRK4T von Kaps-Rentrop	$5.26008*10^{-7}$
GRK4A von Kaps-Rentrop	$5.25444*10^{-7}$
Van Veldhuizen [$\gamma=\frac{1}{2}$]	$5.25843*10^{-7}$
Van Veldhuizen [D-stabil]	$5.26004*10^{-7}$
eine L-stabile Methode	$5.25493*10^{-7}$

Tabelle B.8: Leber: Absoluter Fehler bei Zeitschritt 15

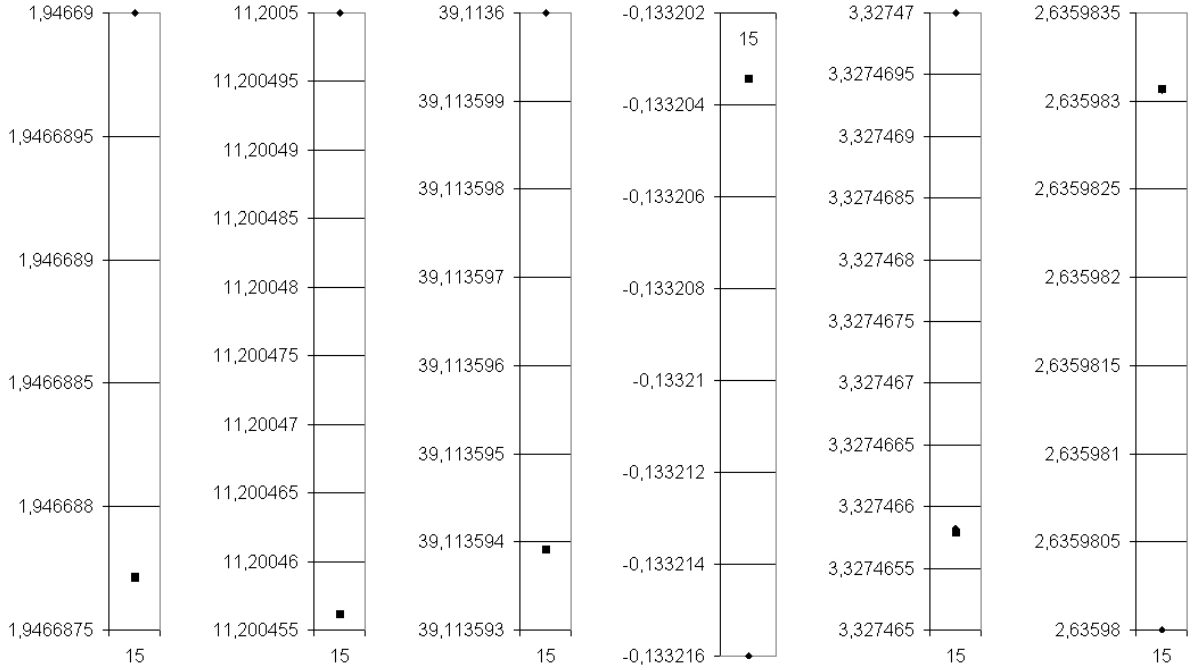
Verfahren	CHYL	CR	DISC
Shampine	$-1.17439*10^{-6}$	$-3.9118*10^{-6}$	$-1.55627*10^{-7}$
GRK4T von Kaps-Rentrop	$-1.17276*10^{-6}$	$-3.91178*10^{-6}$	$-1.55739*10^{-7}$
GRK4A von Kaps-Rentrop	$-1.17399*10^{-6}$	$-3.91176*10^{-6}$	$-1.55684*10^{-7}$
Van Veldhuizen [$\gamma=\frac{1}{2}$]	$-1.17333*10^{-6}$	$-3.91183*10^{-6}$	$-1.55736*10^{-7}$
Van Veldhuizen [D-stabil]	$-1.17276*10^{-6}$	$-3.91178*10^{-6}$	$-1.55736*10^{-7}$
eine L-stabile Methode	$-1.17464*10^{-6}$	$-3.91183*10^{-6}$	$-1.55605*10^{-7}$

Verfahren	HDL2	HDL3	IDL
Shampine	$-9.44263*10^{-5}$	$-1.26189*10^{-6}$	$1.16397*10^{-6}$
GRK4T von Kaps-Rentrop	$-9.4419*10^{-5}$	$-1.26281*10^{-6}$	$1.16213*10^{-6}$
GRK4A von Kaps-Rentrop	$-9.44346*10^{-5}$	$-1.26418*10^{-6}$	$1.16357*10^{-6}$
Van Veldhuizen [$\gamma=\frac{1}{2}$]	$-9.44655*10^{-5}$	$-1.26787*10^{-6}$	$1.16122*10^{-6}$
Van Veldhuizen [D-stabil]	$-9.44237*10^{-5}$	$-1.26272*10^{-6}$	$1.16212*10^{-6}$
eine L-stabile Methode	$-9.44157*10^{-5}$	$-1.26063*10^{-6}$	$1.16416*10^{-6}$

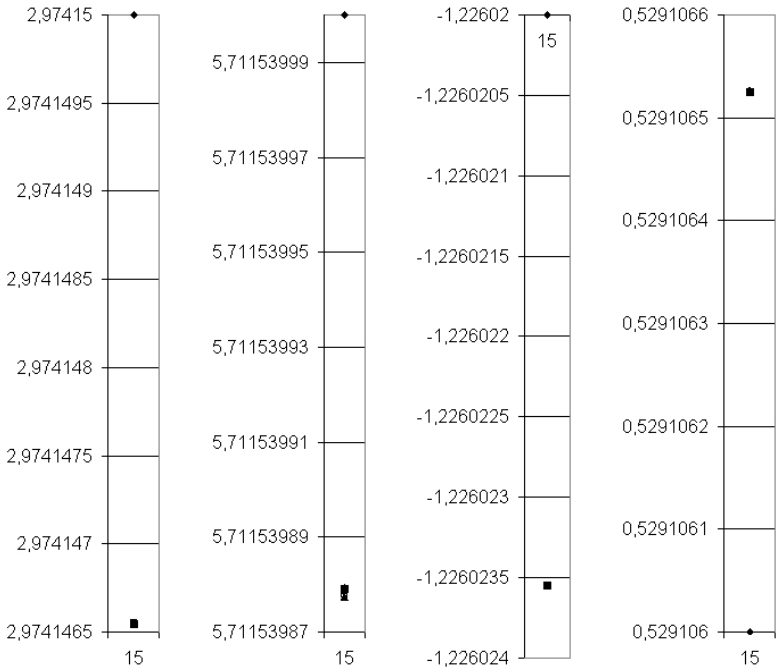
Verfahren	LDL	LDLR	LRP
Shampine	$-1.16056*10^{-6}$	$-2.13699*10^{-8}$	$2.8925*10^{-6}$
GRK4T von Kaps-Rentrop	$-1.15973*10^{-6}$	$-2.11419*10^{-8}$	$2.89268*10^{-6}$
GRK4A von Kaps-Rentrop	$-1.16079*10^{-6}$	$-2.11508*10^{-8}$	$2.8928*10^{-6}$
Van Veldhuizen [$\gamma=\frac{1}{2}$]	$-1.15864*10^{-6}$	$-2.13011*10^{-8}$	$2.89218*10^{-6}$
Van Veldhuizen [D-stabil]	$-1.15971*10^{-6}$	$-2.11515*10^{-8}$	$2.8926*10^{-6}$
eine L-stabile Methode	$-1.16048*10^{-6}$	$-2.14486*10^{-8}$	$2.89216*10^{-6}$

Verfahren	VLDL
Shampine	$9.93197*10^{-7}$
GRK4T von Kaps-Rentrop	$9.94144*10^{-7}$
GRK4A von Kaps-Rentrop	$9.93079*10^{-7}$
Van Veldhuizen [$\gamma=\frac{1}{2}$]	$9.93834*10^{-7}$
Van Veldhuizen [D-stabil]	$9.94137*10^{-7}$
eine L-stabile Methode	$9.93172*10^{-7}$

Tabelle B.9: Leber: Relativer Fehler bei Zeitschritt 15

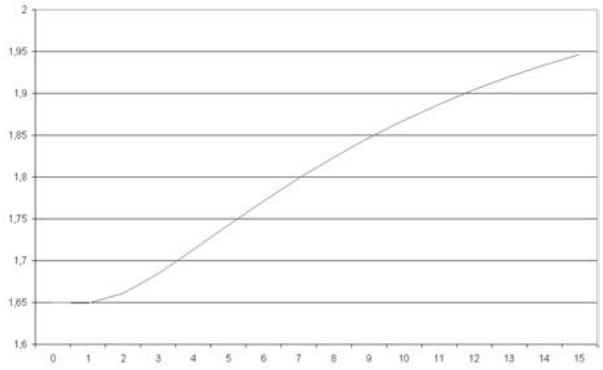


(a) CHYL (b) CR (c) DISC (d) HDL2 (e) HDL3 (f) IDL

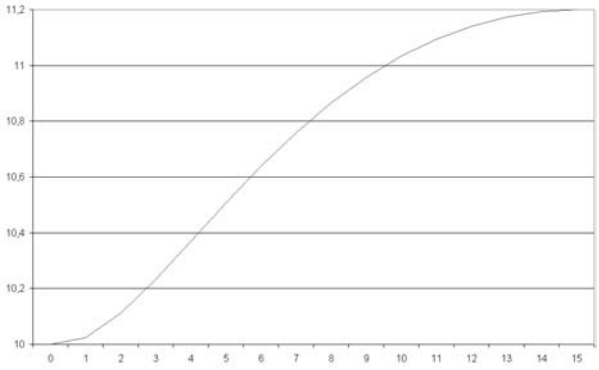


(g) LDL (h) LDLR (i) LRP (j) VLDL

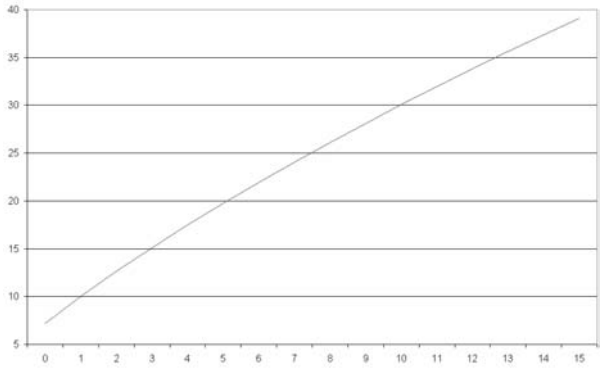
Abbildung B.8: Leber: Absoluter Fehler bei Zeitschritt 15



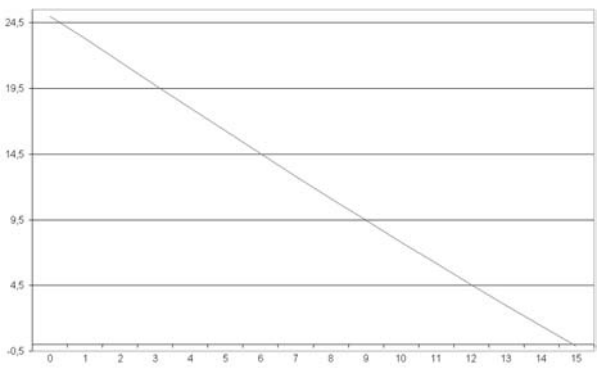
(a) CHYL



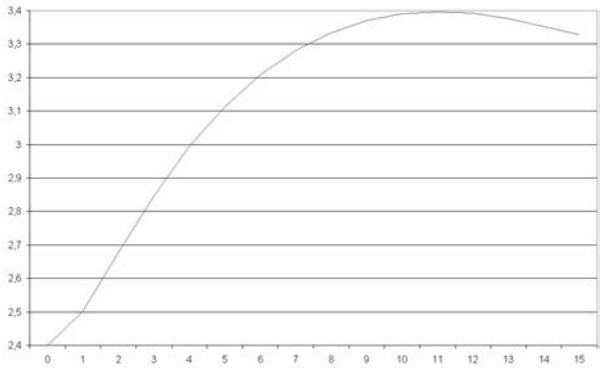
(b) CR



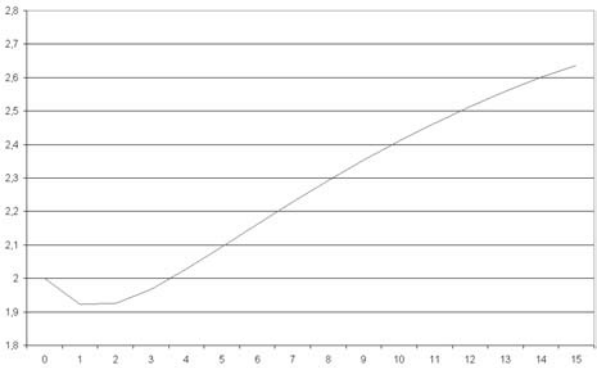
(c) DISC



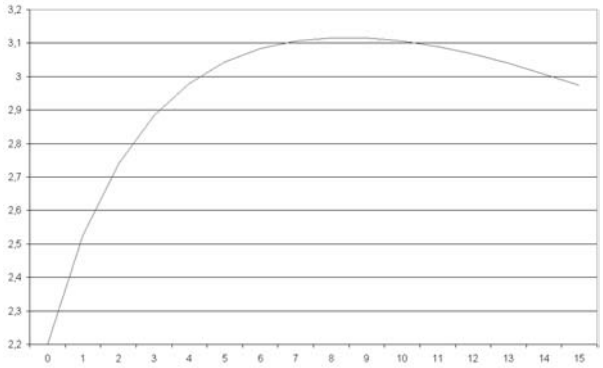
(d) HDL2



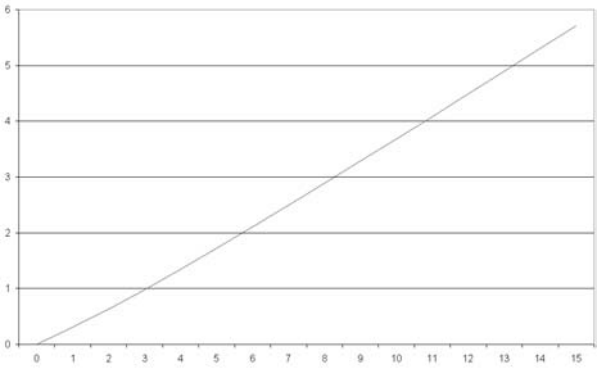
(e) HDL3



(f) IDL

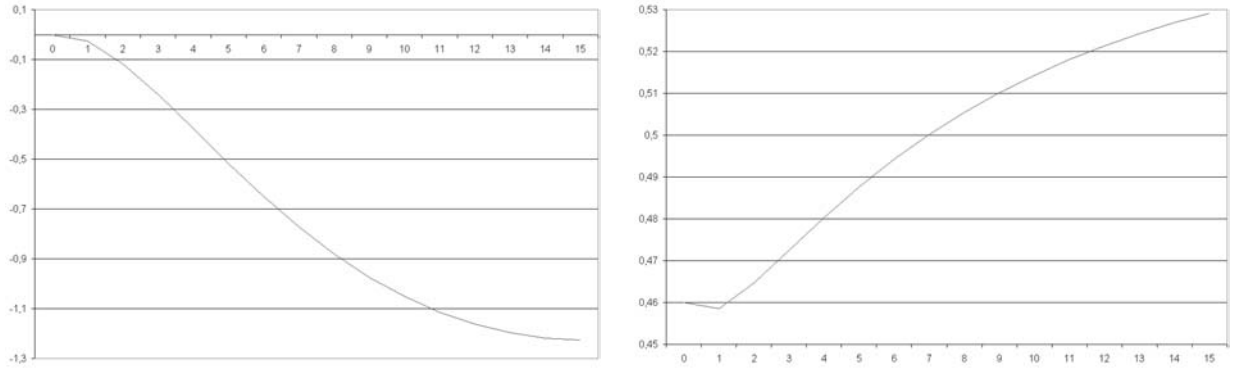


(g) LDL



(h) LDLR

Abbildung B.9: Leber: Kurvenverlaufsdiagramme



(a) LRP (b) VLDL

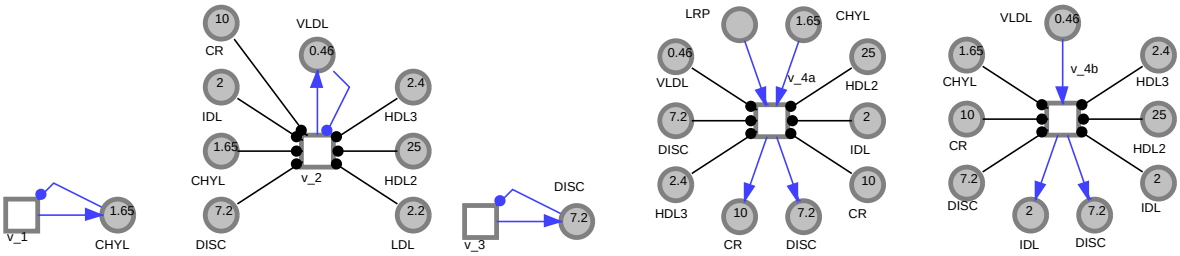
Abbildung B.10: Leber: Kurvenverlaufsdiagramme (Forts.)

$$
\begin{array}{rrcl}
v_1: & & \rightarrow & \mathrm{CHYL} \\
v_2: & & \rightarrow & \mathrm{VLDL} \\
v_3: & & \rightarrow & \mathrm{DISC} \\
v_{4a}: & \mathrm{CHYL}+\mathrm{LRP} & \rightarrow & \mathrm{DISC}+\mathrm{CR} \\
v_{4b}: & \mathrm{VLDL} & \rightarrow & \mathrm{IDL}+\mathrm{DISC} \\
v_5: & \mathrm{DISC} & \rightarrow & \mathrm{HDL3} \\
v_{6a}: & \mathrm{HDL3} & \rightarrow & \mathrm{HDL2} \\
v_{6b}: & \mathrm{HDL3} & \rightarrow & \mathrm{HDL2} \\
v_{7a}: & \mathrm{HDL2} & \rightarrow & \mathrm{HDL3} \\
v_{7b}: & \mathrm{HDL2}+\mathrm{IDL} & \rightarrow & \mathrm{LL}+\mathrm{HDL3} \\
v_{7c}: & \mathrm{IDL} & \rightarrow & \mathrm{LL}
\end{array}
$$

$$
\begin{array}{rrcl}
v_8: & \mathrm{CR} & \rightarrow & \mathrm{LRP} \\
v_9: & \mathrm{IDL} & \rightarrow & \\
v_{10}: & \mathrm{IDL} & \rightarrow & \mathrm{LDLR} \\
v_{11}: & \mathrm{LL} & \rightarrow & \mathrm{LDLR} \\
v_{12}: & \mathrm{CR} & \rightarrow & \\
v_{13}: & \mathrm{CHYL} & \rightarrow & \\
v_{14}: & \mathrm{VLDL} & \rightarrow & \\
v_{15}: & \mathrm{DISC} & \rightarrow & \\
v_{16}: & \mathrm{LL} & \rightarrow & \\
v_{17}: & \mathrm{HDL3} & \rightarrow & \\
v_{18}: & \mathrm{HDL2} & \rightarrow &
\end{array}
$$

$$
\begin{aligned}
\frac{d\mathrm{CHYL}}{dt} &= v_1 - v_{4a} - v_{13} \\
\frac{d\mathrm{CR}}{dt} &= v_{4a} - v_8 - v_{12} \\
\frac{d\mathrm{DISC}}{dt} &= v_{4a} + v_3 + v_{4b} - v_5 - v_{15}
\end{aligned}
\qquad
\begin{aligned}
\frac{d\mathrm{LRP}}{dt} &= v_8 - v_{4a} \\
\frac{d\mathrm{VLDL}}{dt} &= v_2 - v_{4b} - v_{14} \\
\frac{d\mathrm{LDLR}}{dt} &= v_{10} + v_{11}
\end{aligned}
$$

$$
\begin{aligned}
\frac{d\mathrm{HDL2}}{dt} &= v_{6a} + v_{6b} - v_{7a} - v_{7b} - v_{18} \\
\frac{d\mathrm{HDL3}}{dt} &= v_5 + v_{7a} + v_{7b} - v_{6a} - v_{6b} - v_{17} \\
\frac{d\mathrm{IDL}}{dt} &= v_{4b} - v_{7b} - v_{7c} - v_9 - v_{10} \\
\frac{d\mathrm{LL}}{dt} &= v_{7b} + v_{7c} - v_{11} - v_{16}
\end{aligned}
$$



(a) r_1 (b) r_2 (c) r_3 (d) r_{4a} (e) r_{4b}

(f) r_5 (g) r_{6a} (h) r_{6b} (i) r_{7a} (j) r_{7b}

(k) r_{7c} (l) r_8 (m) r_9 (n) r_{10}

(o) r_{11} (p) r_{12} (q) r_{13} (r) r_{14} (s) r_{15} (t) r_{16} (u) r_{17} (v) r_{18}

Abbildung B.11: Leber: Unterstrukturen

$$v_1 = \frac{V_{SynCHYL}}{1+\frac{CHYL}{K_{11}}}$$

$$v_2 = \frac{V_{SynVLDL}}{1+\frac{VLDL}{K_{22}}} * \left(\frac{k_i}{k_i + \frac{V_{LRP}*CR}{1+CR+\frac{IDL}{K_{97}}} * \left(1+\frac{V_{LPL}}{1+CHYL+VLDL+\frac{DISC}{K_{43}}+\frac{HDL3}{K_{45}}+\frac{HDL2}{K_{46}}+\frac{IDL}{K_{47}}+\frac{CR}{K_{48}}} * \frac{CR}{K_{48}}\right) * \frac{V_{HL}}{1+\frac{HDL2}{K_{76}}+\frac{CR}{K_{78}}+\frac{LL}{K_{74}}+\frac{IDL}{K_{75}}+\frac{IDL}{K_{77}}} * \frac{CR}{K_{78}}} \right)$$

$$v_3 = \frac{V_{SynDISC}}{1+\frac{DISC}{K_{33}}}$$

$$v_{4a} = \frac{V_{LPL} * CHYL}{1+CHYL+VLDL+\frac{DISC}{K_{43}}+\frac{HDL3}{K_{45}}+\frac{HDL2}{K_{46}}+\frac{IDL}{K_{47}}+\frac{CR}{K_{48}}}$$

$$v_{4b} = \frac{V_{LPL} * VLDL}{1+CHYL+VLDL+\frac{DISC}{K_{43}}+\frac{HDL3}{K_{45}}+\frac{HDL2}{K_{46}}+\frac{IDL}{K_{47}}+\frac{CR}{K_{48}}}$$

$$v_5 = \frac{V_{LCAT} * DISC}{1+DISC+HDL3+\frac{HDL2}{K_{56}}}$$

$$v_{6a} = \frac{V_{LCAT} * HDL3}{1+DISC+HDL3+\frac{HDL2}{K_{56}}} * \left(1+\frac{V_{LPL}}{1+CHYL+VLDL+\frac{DISC}{K_{43}}+\frac{HDL3}{K_{45}}+\frac{HDL2}{K_{46}}+\frac{IDL}{K_{47}}+\frac{CR}{K_{48}}} * \frac{HDL3}{K_{45}}\right)$$

$$v_{6b} = \frac{V_{LCAT} * HDL3}{1+DISC+HDL3+\frac{HDL2}{K_{56}}} * k_{Chol}$$

$$v_{7a} = \frac{V_{CETP} * HDL2 * \frac{CR}{K_{68}}}{(1+HDL2) * \left(1+IDL+\frac{CR}{K_{68}}\right)} * \frac{V_{HL} * \frac{HDL2}{K_{76}}}{1+\frac{HDL2}{K_{76}}+\frac{CR}{K_{78}}+\frac{LL}{K_{74}}+\frac{IDL}{K_{75}}+\frac{IDL}{K_{77}}}$$

$$v_{7b} = \frac{V_{CETP} * HDL2 * IDL}{(1+HDL2) * \left(1+IDL+\frac{CR}{K_{68}}\right)} * \frac{V_{HL}}{1+\frac{HDL2}{K_{76}}+\frac{CR}{K_{78}}+\frac{LL}{K_{74}}+\frac{IDL}{K_{75}}+\frac{IDL}{K_{77}}} * \frac{HDL2}{K_{76}}$$

$$v_{7c} = \frac{V_{HL}}{1+\frac{HDL2}{K_{76}}+\frac{CR}{K_{78}}+\frac{LL}{K_{74}}+\frac{IDL}{K_{75}}+\frac{IDL}{K_{77}}} * \frac{IDL}{K_{77}}$$

$$v_8 = \frac{V_{LRP} * CR}{1+CR+\frac{IDL}{K_{97}}} * \left(1+\frac{V_{LPL}}{1+CHYL+VLDL+\frac{DISC}{K_{43}}+\frac{HDL3}{K_{45}}+\frac{HDL2}{K_{46}}+\frac{IDL}{K_{47}}+\frac{CR}{K_{48}}} * \frac{CR}{K_{48}}\right) * \frac{V_{HL}}{1+\frac{HDL2}{K_{76}}+\frac{CR}{K_{78}}+\frac{LL}{K_{74}}+\frac{IDL}{K_{75}}+\frac{IDL}{K_{77}}} * \frac{CR}{K_{78}}$$

$$v_9 = \frac{V_{LRP} * \frac{IDL}{K_{97}}}{1+CR+\frac{IDL}{K_{97}}} * \left(1+\frac{V_{LPL}}{1+CHYL+VLDL+\frac{DISC}{K_{43}}+\frac{HDL3}{K_{45}}+\frac{HDL2}{K_{46}}+\frac{IDL}{K_{47}}+\frac{CR}{K_{48}}} * \frac{IDL}{K_{47}}\right) * \frac{V_{HL}}{1+\frac{HDL2}{K_{76}}+\frac{CR}{K_{78}}+\frac{LL}{K_{74}}+\frac{IDL}{K_{75}}+\frac{IDL}{K_{77}}} * \frac{IDL}{K_{77}}$$

$$v_{10} = \frac{V_{LDLR} * \frac{IDL}{K_{87}}}{1+(1+LL)^4+\frac{CR}{K_{88}}+\frac{IDL}{K_{87}}} * \frac{V_{HL}}{1+\frac{HDL2}{K_{76}}+\frac{CR}{K_{78}}+\frac{LL}{K_{74}}+\frac{IDL}{K_{75}}+\frac{IDL}{K_{77}}} * \frac{IDL}{K_{77}} * \left(\frac{k_i}{k_i+\frac{V_{LRP}*CR}{1+CR+\frac{IDL}{K_{97}}} * \left(1+\frac{V_{LPL}}{1+CHYL+VLDL+\frac{DISC}{K_{43}}+\frac{HDL3}{K_{45}}+\frac{HDL2}{K_{46}}+\frac{IDL}{K_{47}}+\frac{CR}{K_{48}}} * \frac{CR}{K_{48}}\right) * \frac{V_{HL}}{1+\frac{HDL2}{K_{76}}+\frac{CR}{K_{78}}+\frac{LL}{K_{74}}+\frac{IDL}{K_{75}}+\frac{IDL}{K_{77}}} * \frac{CR}{K_{78}}}\right)^2$$

$$v_{11} = \frac{V_{LDLR} * LL * (1+LL)^3}{1+(1+LL)^4+\frac{CR}{K_{88}}+\frac{IDL}{K_{87}}} * \frac{V_{HL}}{1+\frac{HDL2}{K_{76}}+\frac{CR}{K_{78}}+\frac{LL}{K_{74}}+\frac{IDL}{K_{75}}+\frac{IDL}{K_{77}}} * \frac{LL}{K_{74}} * \left(\frac{k_i}{k_i+\frac{V_{LRP}*CR}{1+CR+\frac{IDL}{K_{97}}} * \left(1+\frac{V_{LPL}}{1+CHYL+VLDL+\frac{DISC}{K_{43}}+\frac{HDL3}{K_{45}}+\frac{HDL2}{K_{46}}+\frac{IDL}{K_{47}}+\frac{CR}{K_{48}}} * \frac{CR}{K_{48}}\right) * \frac{V_{HL}}{1+\frac{HDL2}{K_{76}}+\frac{CR}{K_{78}}+\frac{LL}{K_{74}}+\frac{IDL}{K_{75}}+\frac{IDL}{K_{77}}} * \frac{CR}{K_{78}}}\right)^2$$

$$v_{12} = \frac{V_{LDLR} * \frac{CR}{K_{88}}}{1+(1+LL)^4+\frac{CR}{K_{88}}+\frac{IDL}{K_{87}}} * \frac{V_{HL}}{1+\frac{HDL2}{K_{76}}+\frac{CR}{K_{78}}+\frac{LL}{K_{74}}+\frac{IDL}{K_{75}}+\frac{IDL}{K_{77}}} * \frac{CR}{K_{78}} * \left(\frac{k_i}{k_i+\frac{V_{LRP}*CR}{1+CR+\frac{IDL}{K_{97}}} * \left(1+\frac{V_{LPL}}{1+CHYL+VLDL+\frac{DISC}{K_{43}}+\frac{HDL3}{K_{45}}+\frac{HDL2}{K_{46}}+\frac{IDL}{K_{47}}+\frac{CR}{K_{48}}} * \frac{CR}{K_{48}}\right) * \frac{V_{HL}}{1+\frac{HDL2}{K_{76}}+\frac{CR}{K_{78}}+\frac{LL}{K_{74}}+\frac{IDL}{K_{75}}+\frac{IDL}{K_{77}}} * \frac{CR}{K_{78}}}\right)^2$$

$$v_{13} = \lambda_1 * CHYL$$

$$v_{14} = \lambda_2 * VLDL$$

$$v_{15} = \lambda_3 * DISC$$

$$v_{16} = \lambda_4 * LL$$

$$v_{17} = \lambda_5 * HDL3$$

$$v_{18} = \lambda_6 * \left(\frac{HDL2}{HDL2+K_a}\right)^3$$

B.4 Kartoffel-Beispiel von Junker

Die Hauptstruktur ist in Abbildung 5.17 auf Seite 46 und die dazugehörigen Parameter sind in Abbildung 5.18 auf Seite 46 abgebildet. Ergänzend dazu folgen die Differentialgleichungen, Startwerte und Reaktionsgleichungen.

Verfahren	Zeit	Schrittweite	Schritte
Copasi	1 s	50	110
Rosenbrockmethode von Shampine	440 s	50	110
Rosenbrockmethode GRK4T von Kaps-Rentrop	346 s	50	110
Rosenbrockmethode GRK4A von Kaps-Rentrop	416 s	50	110
Rosenbrockmethode von Van Veldhuizen [$\gamma = \frac{1}{2}$]	449 s	50	110
Rosenbrockmethode von Van Veldhuizen [D-stabil]	344 s	50	110
eine L-stabile Rosenbrockmethode	446 s	50	110

Tabelle B.10: Kartoffel: Berechnungszeiten

$$\begin{aligned}
\frac{d\mathrm{ADP}}{d\mathrm{t}} &= \mathrm{R}_4 + \mathrm{R}_6 + \mathrm{R}_{11} + \mathrm{R}_{12} + \mathrm{R}_{13} - 29 * \mathrm{R}_8 - 2 * \mathrm{R}_{14} \\
\frac{d\mathrm{ATP}}{d\mathrm{t}} &= 29 * \mathrm{R}_8 + \mathrm{R}_{14} - \mathrm{R}_6 - \mathrm{R}_4 - \mathrm{R}_{11} - \mathrm{R}_{12} - \mathrm{R}_{13}
\end{aligned}$$

$$\begin{aligned}
\frac{d\mathrm{AMP}}{d\mathrm{t}} &= \mathrm{R}_{14} \\
\frac{d\mathrm{F6P}}{d\mathrm{t}} &= \mathrm{R}_4 + \mathrm{R}_5 - \mathrm{R}_8 - \mathrm{R}_9 \\
\frac{d\mathrm{Frc}}{d\mathrm{t}} &= \mathrm{R}_1 + \mathrm{R}_7 - \mathrm{R}_4 \\
\frac{d\mathrm{G1P}}{d\mathrm{t}} &= \mathrm{R}_2 - \mathrm{R}_3 \\
\frac{d\mathrm{G6P}}{d\mathrm{t}} &= \mathrm{R}_3 + \mathrm{R}_6 - \mathrm{R}_5 - \mathrm{R}_{13} \\
\frac{d\mathrm{Glc}}{d\mathrm{t}} &= \mathrm{R}_7 - \mathrm{R}_6 \\
\frac{d\mathrm{P}}{d\mathrm{t}} &= \mathrm{R}_{10}
\end{aligned}$$

$$\begin{aligned}
\frac{d\mathrm{PP}}{d\mathrm{t}} &= \mathrm{R}_{13} - \mathrm{R}_2 \\
\frac{d\mathrm{S6P}}{d\mathrm{t}} &= \mathrm{R}_9 - \mathrm{R}_{10} \\
\frac{d\mathrm{Suc}}{d\mathrm{t}} &= \mathrm{R}_{10} - \mathrm{R}_1 - \mathrm{R}_7 \\
\frac{d\mathrm{UDP}}{d\mathrm{t}} &= \mathrm{R}_9 - \mathrm{R}_1 - \mathrm{R}_{11} \\
\frac{d\mathrm{UDPglc}}{d\mathrm{t}} &= \mathrm{R}_1 - \mathrm{R}_2 - \mathrm{R}_9 \\
\frac{d\mathrm{UTP}}{d\mathrm{t}} &= \mathrm{R}_2 + \mathrm{R}_{11}
\end{aligned}$$

Verfahren	ADP	AMP	ATP
Copasi	0.0707587	0.0390604	0.128181
Shampine	$3.98797 * 10^{-8}$	$4.03777 * 10^{-8}$	$-1.8019 * 10^{-7}$
GRK4T von Kaps-Rentrop	$3.98529 * 10^{-8}$	$4.03784 * 10^{-8}$	$-1.80289 * 10^{-7}$
GRK4A von Kaps-Rentrop	$3.98803 * 10^{-8}$	$4.03787 * 10^{-8}$	$-1.80191 * 10^{-7}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$3.9865 * 10^{-8}$	$4.03774 * 10^{-8}$	$-1.80242 * 10^{-7}$
Van Veldhuizen [D-stabil]	$3.98681 * 10^{-8}$	$4.03786 * 10^{-8}$	$-1.80235 * 10^{-7}$
eine L-stabile Methode	$3.98839 * 10^{-8}$	$4.03769 * 10^{-8}$	$-1.80173 * 10^{-7}$

Verfahren	F6P	Frc	G1P
Copasi	0.19354	0.0129586	0.0509233
Shampine	$2.58697 * 10^{-7}$	$2.70659 * 10^{-8}$	$-4.14069 * 10^{-8}$
GRK4T von Kaps-Rentrop	$2.58661 * 10^{-7}$	$2.70668 * 10^{-8}$	$-4.142 * 10^{-8}$
GRK4A von Kaps-Rentrop	$2.58696 * 10^{-7}$	$2.70661 * 10^{-8}$	$-4.14072 * 10^{-8}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$2.58678 * 10^{-7}$	$2.70663 * 10^{-8}$	$-4.14137 * 10^{-8}$
Van Veldhuizen [D-stabil]	$2.5868 * 10^{-7}$	$2.70664 * 10^{-8}$	$-4.14129 * 10^{-8}$
eine L-stabile Methode	$2.58704 * 10^{-7}$	$2.70657 * 10^{-8}$	$-4.14044 * 10^{-8}$

Verfahren	G6P	Glc	S6P
Copasi	0.351954	0.0044038	0.000850018
Shampine	$-9.33575 * 10^{-8}$	$-4.47347 * 10^{-9}$	$5.88366 * 10^{-11}$
GRK4T von Kaps-Rentrop	$-9.34006 * 10^{-8}$	$-4.47156 * 10^{-9}$	$5.8583 * 10^{-11}$
GRK4A von Kaps-Rentrop	$-9.33601 * 10^{-8}$	$-4.47349 * 10^{-9}$	$5.88282 * 10^{-11}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$-9.33788 * 10^{-8}$	$-4.47244 * 10^{-9}$	$5.87068 * 10^{-11}$
Van Veldhuizen [D-stabil]	$-9.33784 * 10^{-8}$	$-4.47262 * 10^{-9}$	$5.87179 * 10^{-11}$
eine L-stabile Methode	$-9.33484 * 10^{-8}$	$-4.47379 * 10^{-9}$	$5.8886 * 10^{-11}$

Verfahren	UDP	UDPglc	UTP
Copasi	0.028547	0.526104	0.733349
Shampine	$4.13837 * 10^{-8}$	$-3.5764 * 10^{-7}$	$3.16256 * 10^{-7}$
GRK4T von Kaps-Rentrop	$4.13753 * 10^{-8}$	$-3.57718 * 10^{-7}$	$3.16343 * 10^{-7}$
GRK4A von Kaps-Rentrop	$4.13842 * 10^{-8}$	$-3.57642 * 10^{-7}$	$3.16258 * 10^{-7}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$4.13789 * 10^{-8}$	$-3.5768 * 10^{-7}$	$3.16301 * 10^{-7}$
Van Veldhuizen [D-stabil]	$4.13803 * 10^{-8}$	$-3.57676 * 10^{-7}$	$3.16296 * 10^{-7}$
eine L-stabile Methode	$4.13848 * 10^{-8}$	$-3.57625 * 10^{-7}$	$3.16241 * 10^{-7}$

Tabelle B.11: Kartoffel: Absoluter Fehler bei Zeitschritt 5500

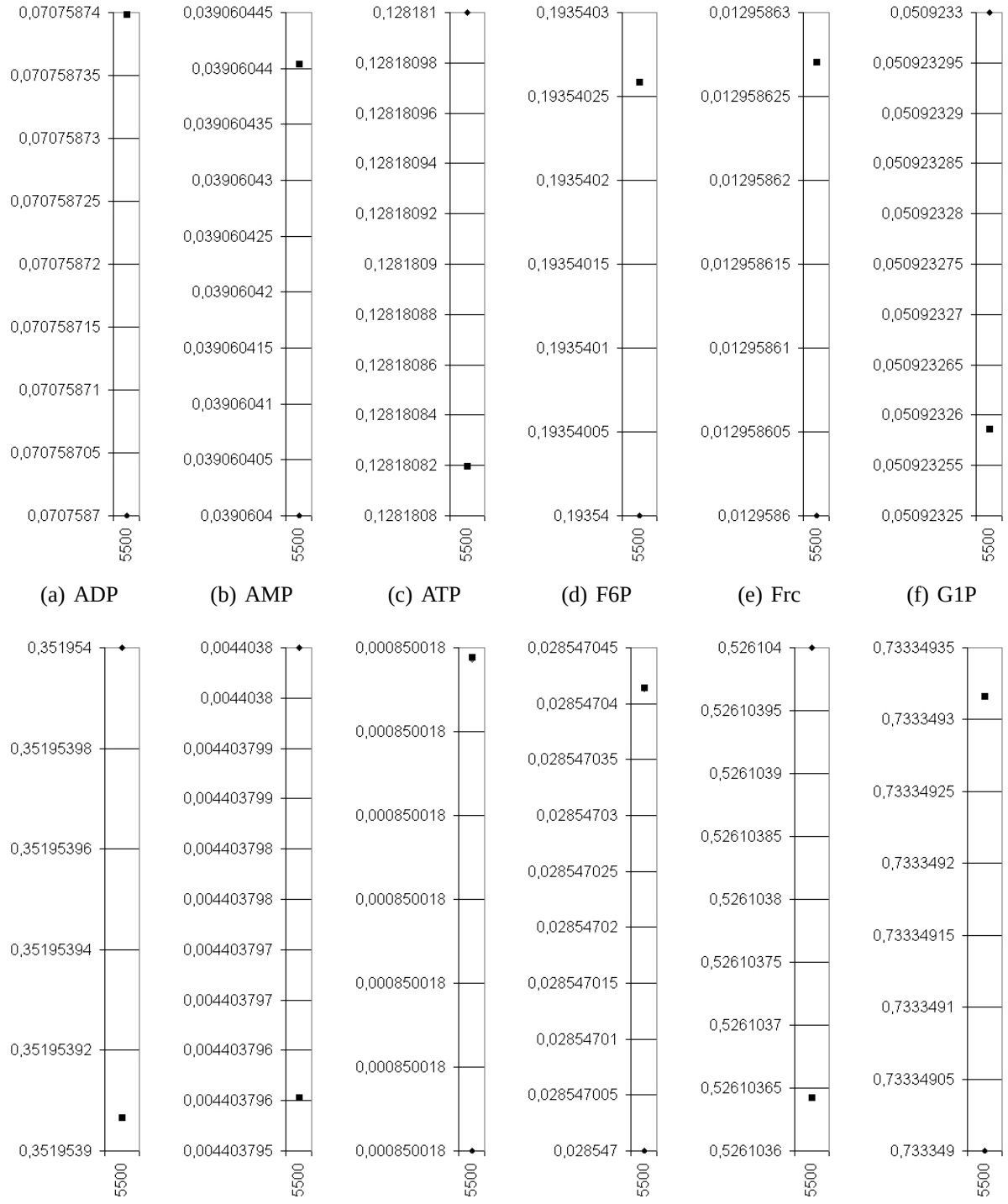
Verfahren	ADP	AMP	ATP
Shampine	$5.63601 * 10^{-7}$	$1.03372 * 10^{-6}$	$-1.40575 * 10^{-6}$
GRK4T von Kaps-Rentrop	$5.63222 * 10^{-7}$	$1.03374 * 10^{-6}$	$-1.40652 * 10^{-6}$
GRK4A von Kaps-Rentrop	$5.6361 * 10^{-7}$	$1.03375 * 10^{-6}$	$-1.40575 * 10^{-6}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$5.63394 * 10^{-7}$	$1.03372 * 10^{-6}$	$-1.40616 * 10^{-6}$
Van Veldhuizen [D-stabil]	$5.63437 * 10^{-7}$	$1.03375 * 10^{-6}$	$-1.40609 * 10^{-6}$
eine L-stabile Methode	$5.63661 * 10^{-7}$	$1.0337 * 10^{-6}$	$-1.40561 * 10^{-6}$

Verfahren	F6P	Frc	G1P
Shampine	$1.33666 * 10^{-6}$	$2.08865 * 10^{-6}$	$-8.13123 * 10^{-7}$
GRK4T von Kaps-Rentrop	$1.33647 * 10^{-6}$	$2.08871 * 10^{-6}$	$-8.1338 * 10^{-7}$
GRK4A von Kaps-Rentrop	$1.33665 * 10^{-6}$	$2.08866 * 10^{-6}$	$-8.13128 * 10^{-7}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$1.33656 * 10^{-6}$	$2.08867 * 10^{-6}$	$-8.13256 * 10^{-7}$
Van Veldhuizen [D-stabil]	$1.33657 * 10^{-6}$	$2.08868 * 10^{-6}$	$-8.13241 * 10^{-7}$
eine L-stabile Methode	$1.33669 * 10^{-6}$	$2.08863 * 10^{-6}$	$-8.13074 * 10^{-7}$

Verfahren	G6P	Glc	S6P
Shampine	$-2.65255 * 10^{-7}$	$-1.01582 * 10^{-6}$	$6.9218 * 10^{-8}$
GRK4T von Kaps-Rentrop	$-2.65377 * 10^{-7}$	$-1.01539 * 10^{-6}$	$6.89197 * 10^{-8}$
GRK4A von Kaps-Rentrop	$-2.65262 * 10^{-7}$	$-1.01582 * 10^{-6}$	$6.92082 * 10^{-8}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$-2.65315 * 10^{-7}$	$-1.01559 * 10^{-6}$	$6.90653 * 10^{-8}$
Van Veldhuizen [D-stabil]	$-2.65314 * 10^{-7}$	$-1.01563 * 10^{-6}$	$6.90784 * 10^{-8}$
eine L-stabile Methode	$-2.65229 * 10^{-7}$	$-1.01589 * 10^{-6}$	$6.92762 * 10^{-8}$

Verfahren	UDP	UDPglc	UTP
Shampine	$1.44967 * 10^{-6}$	$-6.7979 * 10^{-7}$	$4.3125 * 10^{-7}$
GRK4T von Kaps-Rentrop	$1.44938 * 10^{-6}$	$-6.79938 * 10^{-7}$	$4.31367 * 10^{-7}$
GRK4A von Kaps-Rentrop	$1.44969 * 10^{-6}$	$-6.79793 * 10^{-7}$	$4.31251 * 10^{-7}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$1.4495 * 10^{-6}$	$-6.79866 * 10^{-7}$	$4.31311 * 10^{-7}$
Van Veldhuizen [D-stabil]	$1.44955 * 10^{-6}$	$-6.79858 * 10^{-7}$	$4.31303 * 10^{-7}$
eine L-stabile Methode	$1.44971 * 10^{-6}$	$-6.79761 * 10^{-7}$	$4.31228 * 10^{-7}$

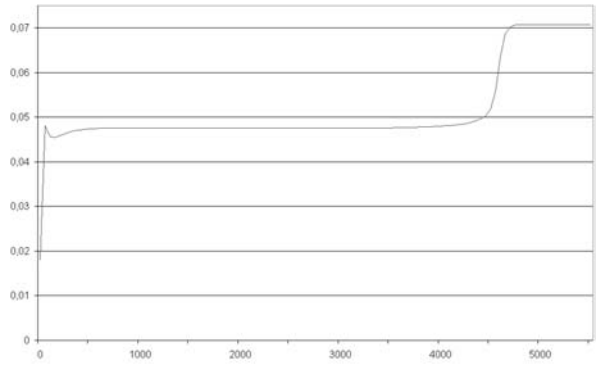
Tabelle B.12: Kartoffel: Relativer Fehler bei Zeitschritt 5500



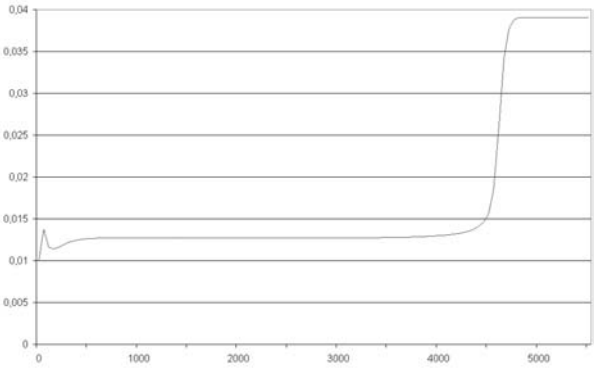
(a) ADP (b) AMP (c) ATP (d) F6P (e) Frc (f) G1P

(g) G6P (h) Glc (i) S6P (j) UDP (k) UDPglc (l) UTP

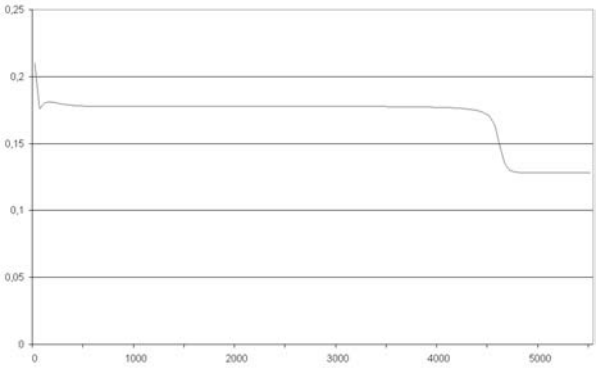
Abbildung B.12: Kartoffel: Absoluter Fehler bei Zeitschritt 5500



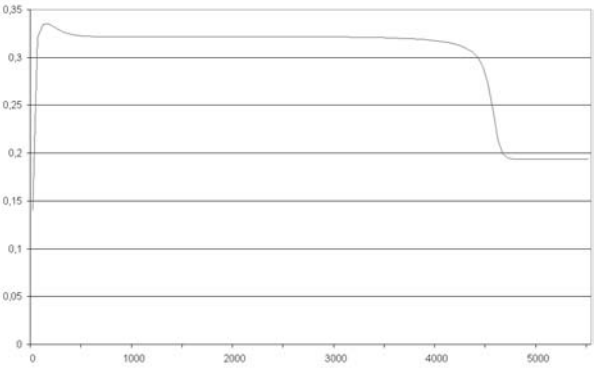
(a) ADP



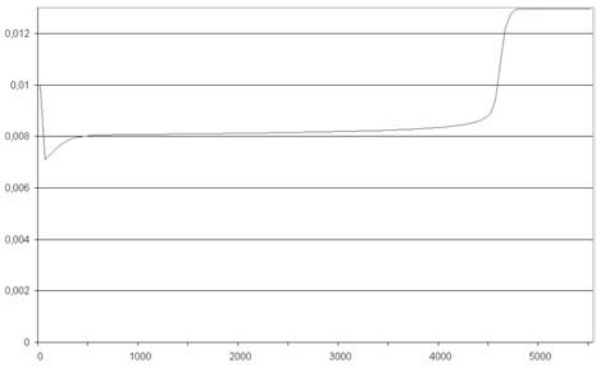
(b) AMP



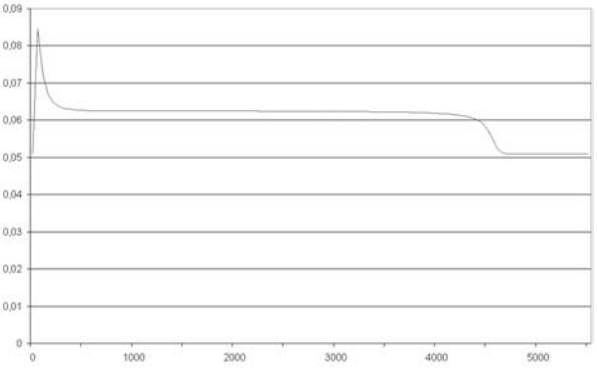
(c) ATP



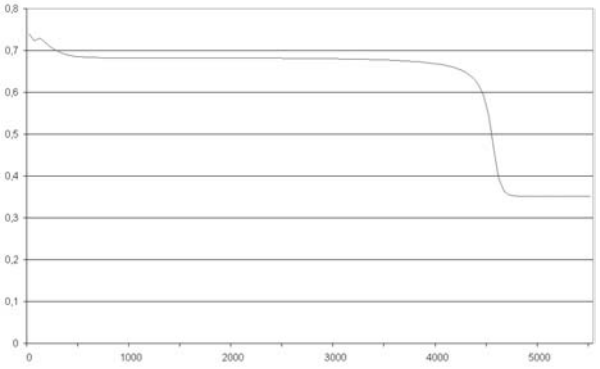
(d) F6P



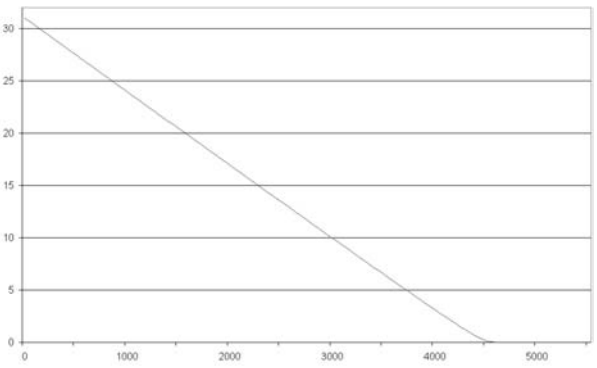
(e) Frc



(f) G1P

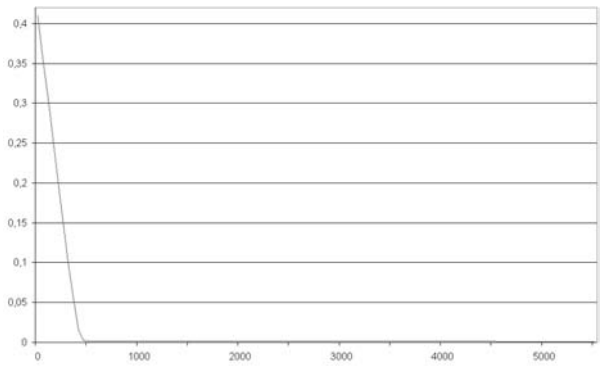


(g) G6P

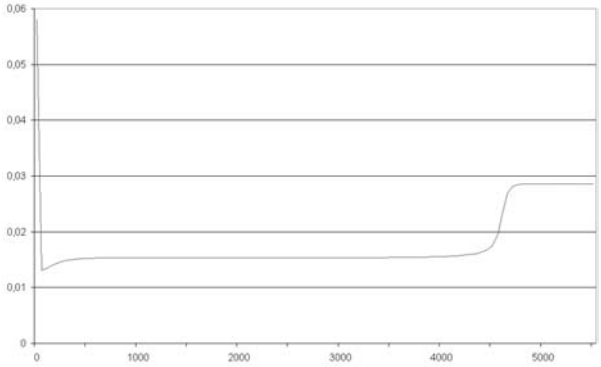


(h) Glc

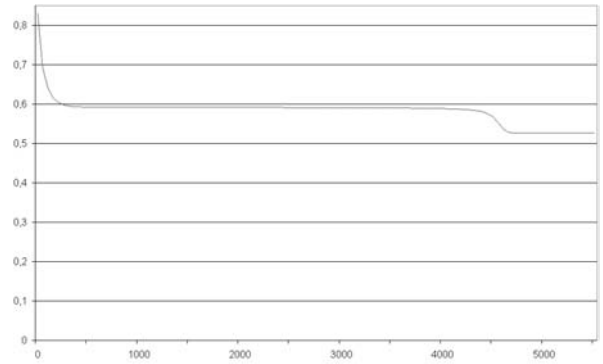
Abbildung B.13: Kartoffel: Kurvenverlaufsdiagramme



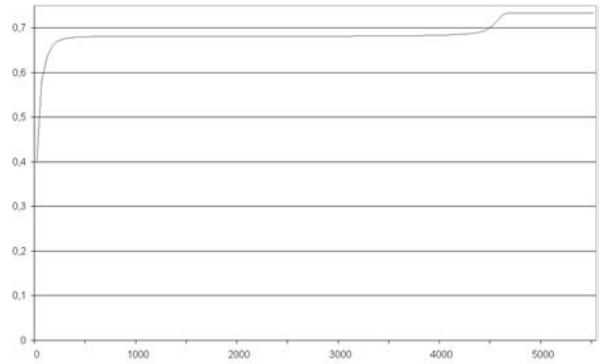
(a) S6P



(b) UDP



(c) UDPglc



(d) UTP

Abbildung B.14: Kartoffel: Kurvenverlaufsdiagramme (Forts.)

ADP	=	0.018	Kip_2	=	0.1	Vm_7	=	0.0006
AMP	=	0.01	Kma_2	=	0.137	$Kiatp_8$	=	0.21
ATP	=	0.21	Kmb_2	=	0.127	$Kmadp_8$	=	0.018
F6P	=	0.14	Kmp_2	=	0.16	$Kmf6p_8$	=	0.14
Frc	=	0.01	Kmq_2	=	0.142	Vm_8	=	0.01
G1P	=	0.051	Vf_2	=	1.306	Keq_9	=	10.0
G6P	=	0.74	Vr_2	=	0.782	$Kif6p_9$	=	0.14
Glc	=	31	Keq_3	=	19	Kip_9	=	3.0
P	=	33.4	Kmp_3	=	0.5	$Kis6p_9$	=	0.41
PP	=	0.023	Kms_3	=	0.06	$Kiudpg_9$	=	1.4
S6P	=	0.41	Vf_3	=	0.0228	$Kmf6p_9$	=	0.3
Suc	=	41	$Kiadp_4$	=	0.078	$Kms6p_9$	=	0.41
UDP	=	0.058	$Kifrc_4$	=	5.9	$Kmudp_9$	=	0.3
UDPglc	=	0.83	$Kmatp_4$	=	0.026	$Kmudpg_9$	=	4.6
UTP	=	0.4	$Kmfrc_4$	=	0.077	Vf_9	=	0.0077
Keq_1	=	0.35	Vm_4	=	0.058	Vr_9	=	0.0041
$Kifrc_1$	=	7.8	Keq_5	=	2	$Kisuc_{10}$	=	41.0
$Kisuc_1$	=	40.0	Kmp_5	=	0.27	$Kms6p_{10}$	=	0.01
$Kiudp_1$	=	0.058	Kms_5	=	0.15	Vm_{10}	=	0.0025
$Kmfrc_1$	=	7.8	Vf_5	=	0.0406	$k1_{11}$	=	0.1
$Kmsuc_1$	=	50.0	$Kig6p_6$	=	4.1	$k2_{11}$	=	0.1
$Kmudp_1$	=	0.058	$Kmatp_6$	=	0.185	v_{12}	=	0.1
$Kmudpg_1$	=	0.076	$Kmglc_6$	=	0.13	$Kmatp_{13}$	=	0.21
Vf_1	=	0.0354	Vm_6	=	0.0147	$Kmg6p_{13}$	=	0.74
Vr_1	=	0.0802	$Kifrc_7$	=	0.01	Vm_{13}	=	0.03
Keq_2	=	3.2	$Kiglc_7$	=	31	$k1_{14}$	=	1
Kia_2	=	0.02	$Kmsuc_7$	=	41	$k2_{14}$	=	1
Kib_2	=	0.1						

R_1 :	UDP + Suc	$\rightleftharpoons$	UDPglc + Frc	R_8 :	F6P + 29 ∗ ADP	$\rightarrow$	29 ∗ ATP
R_2 :	UDPglc + PP	$\rightleftharpoons$	G1P + UTP	R_9 :	F6P + UDPglc	$\rightleftharpoons$	S6P + UDP
R_3 :	G1P	$\rightleftharpoons$	G6P	R_{10} :	S6P	$\rightarrow$	P + Suc
R_4 :	Frc + ATP	$\rightarrow$	ADP + F6P	R_{11} :	ATP + UDP	$\rightleftharpoons$	UTP + ADP
R_5 :	G6P	$\rightleftharpoons$	F6P	R_{12} :	ATP	$\rightarrow$	ADP
R_6 :	Glc + ATP	$\rightarrow$	G6P + ADP	R_{13} :	G6P + ATP	$\rightarrow$	PP + ADP
R_7 :	Suc	$\rightarrow$	Frc + Glc	R_{14} :	2 ∗ ADP	$\rightleftharpoons$	AMP + ATP

$$R_1 = \frac{Vf_1 * \left(Suc * UDP - \frac{UDPglc * Frc}{Keq_1}\right)}{Suc * UDP * \left(1 + \frac{UDPglc}{Kifrc_1}\right) + Kmsuc_1 * (UDP + Kiudp_1) + Kmudp_1 * Suc + \frac{Vf_1}{Vr_1 * Keq_1} * \left[UDPglc * Kmudpg_1 * \left(1 + \frac{UDP}{Kiudp_1}\right) + Frc * \left\{Kmfrc_1 * \left(1 + \frac{Kmudp_1 * Suc}{Kiudp_1 * Kmsuc_1}\right) + UDPglc * \left(1 + \frac{Suc}{Kisuc_1}\right)\right\}\right]}$$

$$R_2 = \frac{Vf_2 * \left(UDPglc * PP - \frac{G1P * UTP}{Keq_2}\right)}{UDPglc * PP * (1 + G1P * Kip_2) + Kmb_2 * (UDPglc + Kia_2) + Kma_2 * PP + \frac{Vf_2}{Vr_2 * Keq_2} * Kmq_2 * G1P * \left(1 + \frac{UDPglc}{Kia_2}\right) * UTP * \left(Kmp_2 * \left(1 + \frac{Kma_2 * PP}{Kia_2 * Kmb_2}\right) + G1P * \left(1 + \frac{PP}{Kib_2}\right)\right)}$$

$$R_3 = \frac{Vf_3 * \left(G1P - \frac{G6P}{Keq_3}\right)}{G1P + Kms_3 * \left(1 + \frac{G6P}{Kmp_3}\right)}$$

$$R_4 = \frac{Vm_4}{1 + \frac{Frc}{Kifrc_4}} * \frac{\frac{Frc * ATP}{Kmfrc_4 * Kmatp_4}}{1 + \frac{Frc}{Kmfrc_4} + \frac{ATP}{Kmatp_4} + \frac{Frc * ATP}{Kmfrc_4 * Kmatp_4} + \frac{F6P}{Kiadp_4}}$$

$$R_5 = \frac{Vf_5 * \left(F6P - \frac{G6P}{Keq_5}\right)}{F6P + Kms_5 * \left(1 + \frac{G6P}{Kmp_5}\right)}$$

$$R_6 = Vm_6 * \frac{\frac{Glc * ATP}{Kmglc_6 * Kmatp_6}}{\left(1 + \frac{ATP}{Kmatp_6}\right) * \left(1 + \frac{Glc}{Kmglc_6} + \frac{G6P}{Kig6p_6}\right)}$$

$$R_7 = \frac{Vm_7}{1 + \frac{Glc}{Kiglc_7}} * \frac{Suc}{Kmsuc_7 * \left(1 + \frac{Frc}{Kifrc_7}\right) + Suc}$$

$$R_8 = Vm_8 * \frac{\frac{F6P * ADP}{Kmf6p_8 * Kmadp_8}}{\left(1 + \frac{ADP}{Kmadp_8}\right) * \left(1 + \frac{F6P}{Kmf6p_8} + \frac{ATP}{Kiatp_8}\right)}$$

$$R_9 = \frac{Vf_9 * \left(F6P * UDPglc - \frac{S6P * UDP}{Keq_9}\right)}{F6P * UDPglc * \left(1 + \frac{S6P}{Kis6p_9}\right) + Kmf6p_9 * \left(1 + \frac{P}{Kip_9}\right) * (UDPglc + Kiudpg_9) + Kmudpg_9 * F6P + \frac{Vf_9}{Vr_9 * Keq_9} * \left[S6P * Kmudp_9 * \left(1 + \frac{UDPglc}{Kiudpg_9}\right) + UDP * \left\{Kms6p_9 * \left(1 + \frac{Kmudpg_9 * F6P}{Kiudpg_9 * Kmf6p_9 * \left(1 + \frac{P}{Kip_9}\right)}\right) + S6P * \left(1 + \frac{F6P}{Kif6p_9}\right)\right\}\right]}$$

$$R_{10} = \frac{Vm_{10} * S6P}{Kms6p_{10} * \left(1 + \frac{Suc}{Kisuc_{10}}\right) + S6P * \left(1 + \frac{Suc}{Kisuc_{10}}\right)}$$

$$R_{11} = k1_{11} * UDP * ATP - k2_{11} * UTP * ADP$$

$$R_{12} = v_{12}$$

$$R_{13} = Vm_{13} * \frac{G6P * ATP}{G6P * ATP + Kmg6p_{13} * ATP + Kmatp_{13} * G6P}$$

$$R_{14} = k1_{14} * ADP * ADP - k2_{14} * AMP * ATP$$

B.5 Hefe-Beispiel von Klipp/Kofahl

In der folgenden Abbildung B.23 sind die Unterstrukturen zum als Petrinetz umgesetzten Hefe-Beispiel [KK04] dargestellt. Die Hauptstruktur ist in Abbildung 5.20 auf Seite 48 und die dazugehörigen Parameter in Abbildung 5.19 auf Seite 47 abgebildet. Die Differentialgleichungen, Reaktionsgleichungen und Startwerte sind ab Seite 105 aufgeführt.

Verfahren	Zeit	Schrittweite	Schritte
Copasi	2 s	1	200
Rosenbrockmethode von Shampine	8002 s	1	200
Rosenbrockmethode GRK4T von Kaps-Rentrop	8880 s	1	200
Rosenbrockmethode GRK4A von Kaps-Rentrop	8873 s	1	200
Rosenbrockmethode von Van Veldhuizen [$\gamma = \frac{1}{2}$]	7697 s	1	200
Rosenbrockmethode von Van Veldhuizen [D-stabil]	8573 s	1	200
eine L-stabile Rosenbrockmethode	7072 s	1	200

Tabelle B.13: Hefe: Berechnungszeiten

Verfahren	α	A	B
Copasi	$1.83 * 10^{-184}$	105.297	75.6775
Shampine	$-1.8266 * 10^{-184}$	$5.86929 * 10^{-5}$	$-5.00287 * 10^{-6}$
GRK4T von Kaps-Rentrop	$-1.8266 * 10^{-184}$	$5.86937 * 10^{-5}$	$-5.00402 * 10^{-6}$
GRK4A von Kaps-Rentrop	$-1.8266 * 10^{-184}$	$5.86941 * 10^{-5}$	$-5.00449 * 10^{-6}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$-1.8266 * 10^{-184}$	$5.86942 * 10^{-5}$	$-5.00442 * 10^{-6}$
Van Veldhuizen [D-stabil]	$-1.8266 * 10^{-184}$	$5.86942 * 10^{-5}$	$-5.00448 * 10^{-6}$
eine L-stabile Methode	$-1.8266 * 10^{-184}$	$5.86952 * 10^{-5}$	$-5.00616 * 10^{-6}$
Verfahren	Bar1	$\text{Bar1}_{\text{active}}$	C
Copasi	22.3152	15.1223	239.059
Shampine	$3.58019 * 10^{-5}$	$-2.19984 * 10^{-5}$	$-4.72979 * 10^{-4}$
GRK4T von Kaps-Rentrop	$3.58036 * 10^{-5}$	$-2.1998 * 10^{-5}$	$-4.72981 * 10^{-4}$
GRK4A von Kaps-Rentrop	$3.58037 * 10^{-5}$	$-2.19982 * 10^{-5}$	$-4.72982 * 10^{-4}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$3.58027 * 10^{-5}$	$-2.19987 * 10^{-5}$	$-4.72981 * 10^{-4}$
Van Veldhuizen [D-stabil]	$3.58036 * 10^{-5}$	$-2.1998 * 10^{-5}$	$-4.72981 * 10^{-4}$
eine L-stabile Methode	$3.58032 * 10^{-5}$	$-2.19996 * 10^{-5}$	$-4.72984 * 10^{-4}$

Tabelle B.14: Hefe: Absoluter Fehler bei Zeitschritt 200

Verfahren	Cdc28	D	E
Copasi	64.8127	$2.55 * 10^{-5}$	$7.96 * 10^{-6}$
Shampine	$-1.26531 * 10^{-5}$	$1.89754 * 10^{-13}$	$-1.65522 * 10^{-12}$
GRK4T von Kaps-Rentrop	$-1.26484 * 10^{-5}$	$1.89758 * 10^{-13}$	$-1.65551 * 10^{-12}$
GRK4A von Kaps-Rentrop	$-1.26483 * 10^{-5}$	$1.89749 * 10^{-13}$	$-1.65559 * 10^{-12}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$-1.26531 * 10^{-5}$	$1.89754 * 10^{-13}$	$-1.65533 * 10^{-12}$
Van Veldhuizen [D-stabil]	$-1.26488 * 10^{-5}$	$1.89755 * 10^{-13}$	$-1.65556 * 10^{-12}$
eine L-stabile Methode	$-1.26565 * 10^{-5}$	$1.89749 * 10^{-13}$	$-1.65537 * 10^{-12}$

Verfahren	F	Far1	$\text{Far1}_{\text{Ubiquitin}}$
Copasi	$1.53 * 10^{-6}$	2.7047	252.622
Shampine	$-4.86516 * 10^{-12}$	$-3.93667 * 10^{-6}$	$2.64744 * 10^{-4}$
GRK4T von Kaps-Rentrop	$-4.86526 * 10^{-12}$	$-3.93721 * 10^{-6}$	$2.64744 * 10^{-4}$
GRK4A von Kaps-Rentrop	$-4.86525 * 10^{-12}$	$-3.93718 * 10^{-6}$	$2.64745 * 10^{-4}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$-4.86517 * 10^{-12}$	$-3.93669 * 10^{-6}$	$2.64744 * 10^{-4}$
Van Veldhuizen [D-stabil]	$-4.86526 * 10^{-12}$	$-3.93715 * 10^{-6}$	$2.64744 * 10^{-4}$
eine L-stabile Methode	$-4.86517 * 10^{-12}$	$-3.93633 * 10^{-6}$	$2.64744 * 10^{-4}$

Verfahren	Far1PP	Fus3	Fus3PP
Copasi	0.326806	684.564	$1.91396 * 10^{-3}$
Shampine	$-4.75971 * 10^{-7}$	$8.72503 * 10^{-6}$	$-1.54013 * 10^{-9}$
GRK4T von Kaps-Rentrop	$-4.7073 * 10^{-7}$	$8.69854 * 10^{-6}$	$-1.54114 * 10^{-9}$
GRK4A von Kaps-Rentrop	$-4.7136 * 10^{-7}$	$8.70065 * 10^{-6}$	$-1.53896 * 10^{-9}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$-4.75973 * 10^{-7}$	$8.70404 * 10^{-6}$	$-1.54026 * 10^{-9}$
Van Veldhuizen [D-stabil]	$-4.71148 * 10^{-7}$	$8.70337 * 10^{-6}$	$-1.54034 * 10^{-9}$
eine L-stabile Methode	$-4.79945 * 10^{-7}$	$8.69707 * 10^{-6}$	$-1.53934 * 10^{-9}$

Verfahren	G	GαGDP	GαGTP
Copasi	$2.05546 * 10^{-7}$	9.15998	$-5.85 * 10^{-39}$
Shampine	$4.62332 * 10^{-13}$	$-3.65635 * 10^{-6}$	$-3.43648 * 10^{-41}$
GRK4T von Kaps-Rentrop	$4.62317 * 10^{-13}$	$-3.65667 * 10^{-6}$	$-3.43614 * 10^{-41}$
GRK4A von Kaps-Rentrop	$4.62319 * 10^{-13}$	$-3.65689 * 10^{-6}$	$-3.43622 * 10^{-41}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$4.62331 * 10^{-13}$	$-3.65624 * 10^{-6}$	$-3.43648 * 10^{-41}$
Van Veldhuizen [D-stabil]	$4.62318 * 10^{-13}$	$-3.65646 * 10^{-6}$	$-3.43613 * 10^{-41}$
eine L-stabile Methode	$4.62331 * 10^{-13}$	$-3.65598 * 10^{-6}$	$-3.4368 * 10^{-41}$

Verfahren	G$\alpha\beta\gamma$	G$\beta\gamma$	H
Copasi	1657.51	$5.00 * 10^{-6}$	$1.29 * 10^{-6}$
Shampine	$2.36565 * 10^{-5}$	$-6.27199 * 10^{-13}$	$-3.04532 * 10^{-12}$
GRK4T von Kaps-Rentrop	$2.36566 * 10^{-5}$	$-6.26374 * 10^{-13}$	$-3.04545 * 10^{-12}$
GRK4A von Kaps-Rentrop	$2.36569 * 10^{-5}$	$-6.27643 * 10^{-13}$	$-3.04541 * 10^{-12}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$2.36563 * 10^{-5}$	$-6.27233 * 10^{-13}$	$-3.04533 * 10^{-12}$
Van Veldhuizen [D-stabil]	$2.36565 * 10^{-5}$	$-6.27139 * 10^{-13}$	$-3.04544 * 10^{-12}$
eine L-stabile Methode	$2.36562 * 10^{-5}$	$-6.28216 * 10^{-13}$	$-3.04533 * 10^{-12}$

Tabelle B.15: Hefe: Absoluter Fehler bei Zeitschritt 200 (Forts.)

Verfahren	I	K	L
Copasi	$6.57749 * 10^{-4}$	$3.68074 * 10^{-4}$	$1.25 * 10^{-5}$
Shampine	$-2.57594 * 10^{-10}$	$-1.92361 * 10^{-10}$	$1.80415 * 10^{-11}$
GRK4T von Kaps-Rentrop	$-2.57606 * 10^{-10}$	$-1.92367 * 10^{-10}$	$1.80406 * 10^{-11}$
GRK4A von Kaps-Rentrop	$-2.57616 * 10^{-10}$	$-1.92374 * 10^{-10}$	$1.8043 * 10^{-11}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$-2.57631 * 10^{-10}$	$-1.92382 * 10^{-10}$	$1.80411 * 10^{-11}$
Van Veldhuizen [D-stabil]	$-2.57608 * 10^{-10}$	$-1.92369 * 10^{-10}$	$1.80415 * 10^{-11}$
eine L-stabile Methode	$-2.57686 * 10^{-10}$	$-1.92413 * 10^{-10}$	$1.80414 * 10^{-11}$

Verfahren	M	N	$Sst2_{active}$
Copasi	9.15892	235.187	$4.63 * 10^{-5}$
Shampine	$1.01563 * 10^{-6}$	$3.12653 * 10^{-4}$	$2.23653 * 10^{-12}$
GRK4T von Kaps-Rentrop	$1.01529 * 10^{-6}$	$3.12648 * 10^{-4}$	$2.35594 * 10^{-12}$
GRK4A von Kaps-Rentrop	$1.01503 * 10^{-6}$	$3.12648 * 10^{-4}$	$2.32865 * 10^{-12}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$1.01564 * 10^{-6}$	$3.12653 * 10^{-4}$	$2.22622 * 10^{-12}$
Van Veldhuizen [D-stabil]	$1.01538 * 10^{-6}$	$3.12649 * 10^{-4}$	$2.35208 * 10^{-12}$
eine L-stabile Methode	$1.01568 * 10^{-6}$	$3.12656 * 10^{-4}$	$2.12487 * 10^{-12}$

Verfahren	Ste11	Ste12	$Ste12_{active}$
Copasi	155.633	199.313	0.687019
Shampine	$3.63959 * 10^{-4}$	$-1.84899 * 10^{-5}$	$-5.10089 * 10^{-7}$
GRK4T von Kaps-Rentrop	$3.63958 * 10^{-4}$	$-1.84899 * 10^{-5}$	$-5.10085 * 10^{-7}$
GRK4A von Kaps-Rentrop	$3.63958 * 10^{-4}$	$-1.84899 * 10^{-5}$	$-5.10103 * 10^{-7}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$3.63958 * 10^{-4}$	$-1.84899 * 10^{-5}$	$-5.1013 * 10^{-7}$
Van Veldhuizen [D-stabil]	$3.63958 * 10^{-4}$	$-1.84899 * 10^{-5}$	$-5.10086 * 10^{-7}$
eine L-stabile Methode	$3.63957 * 10^{-4}$	$-1.84898 * 10^{-5}$	$-5.10206 * 10^{-7}$

Verfahren	Ste2	$Ste2_{active}$	Ste20
Copasi	12.2953	$5.35 * 10^{-62}$	999.999
Shampine	$-4.53692 * 10^{-5}$	$-5.34648 * 10^{-62}$	$-5.03028 * 10^{-5}$
GRK4T von Kaps-Rentrop	$-4.53691 * 10^{-5}$	$-5.34648 * 10^{-62}$	$-5.0303 * 10^{-5}$
GRK4A von Kaps-Rentrop	$-4.53692 * 10^{-5}$	$-5.34648 * 10^{-62}$	$-5.0303 * 10^{-5}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$-4.53692 * 10^{-5}$	$-5.34648 * 10^{-62}$	$-5.03031 * 10^{-5}$
Van Veldhuizen [D-stabil]	$-4.53691 * 10^{-5}$	$-5.34648 * 10^{-62}$	$-5.03031 * 10^{-5}$
eine L-stabile Methode	$-4.53693 * 10^{-5}$	$-5.34648 * 10^{-62}$	$-5.03028 * 10^{-5}$

Verfahren	Ste5	Ste7
Copasi	155.633	35.2529
Shampine	$3.63959 * 10^{-4}$	$2.76561 * 10^{-5}$
GRK4T von Kaps-Rentrop	$3.63958 * 10^{-4}$	$2.76571 * 10^{-5}$
GRK4A von Kaps-Rentrop	$3.63958 * 10^{-4}$	$2.76569 * 10^{-5}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$3.63958 * 10^{-4}$	$2.76568 * 10^{-5}$
Van Veldhuizen [D-stabil]	$3.63958 * 10^{-4}$	$2.76568 * 10^{-5}$
eine L-stabile Methode	$3.63957 * 10^{-4}$	$2.76567 * 10^{-5}$

Tabelle B.16: Hefe: Absoluter Fehler bei Zeitschritt 200 (Forts.)

Verfahren	α	A	B
Shampine	-1	$5.57404 * 10^{-7}$	$-6.61077 * 10^{-8}$
GRK4T von Kaps-Rentrop	-1	$5.57411 * 10^{-7}$	$-6.61229 * 10^{-8}$
GRK4A von Kaps-Rentrop	-1	$5.57415 * 10^{-7}$	$-6.61292 * 10^{-8}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	-1	$5.57416 * 10^{-7}$	$-6.61283 * 10^{-8}$
Van Veldhuizen [D-stabil]	-1	$5.57415 * 10^{-7}$	$-6.6129 * 10^{-8}$
eine L-stabile Methode	-1	$5.57425 * 10^{-7}$	$-6.61513 * 10^{-8}$

Verfahren	Bar1	$\text{Bar1}_{\text{active}}$	C
Shampine	$1.60437 * 10^{-6}$	$-1.4547 * 10^{-6}$	$-1.9785 * 10^{-6}$
GRK4T von Kaps-Rentrop	$1.60445 * 10^{-6}$	$-1.45467 * 10^{-6}$	$-1.97851 * 10^{-6}$
GRK4A von Kaps-Rentrop	$1.60445 * 10^{-6}$	$-1.45468 * 10^{-6}$	$-1.97851 * 10^{-6}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$1.60441 * 10^{-6}$	$-1.45472 * 10^{-6}$	$-1.97851 * 10^{-6}$
Van Veldhuizen [D-stabil]	$1.60445 * 10^{-6}$	$-1.45467 * 10^{-6}$	$-1.97851 * 10^{-6}$
eine L-stabile Methode	$1.60443 * 10^{-6}$	$-1.45478 * 10^{-6}$	$-1.97853 * 10^{-6}$

Verfahren	Cdc28	D	E
Shampine	$-1.95226 * 10^{-7}$	$7.45007 * 10^{-6}$	$-2.07957 * 10^{-7}$
GRK4T von Kaps-Rentrop	$-1.95153 * 10^{-7}$	$7.45022 * 10^{-6}$	$-2.07993 * 10^{-7}$
GRK4A von Kaps-Rentrop	$-1.95151 * 10^{-7}$	$7.44987 * 10^{-6}$	$-2.08003 * 10^{-7}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$-1.95225 * 10^{-7}$	$7.45005 * 10^{-6}$	$-2.0797 * 10^{-7}$
Van Veldhuizen [D-stabil]	$-1.9516 * 10^{-7}$	$7.4501 * 10^{-6}$	$-2.08 * 10^{-7}$
eine L-stabile Methode	$-1.95277 * 10^{-7}$	$7.44986 * 10^{-6}$	$-2.07976 * 10^{-7}$

Verfahren	F	Far1	$\text{Far1}_{\text{Ubiquitin}}$
Shampine	$-3.17845 * 10^{-6}$	$-1.45549 * 10^{-6}$	$1.04798 * 10^{-6}$
GRK4T von Kaps-Rentrop	$-3.17852 * 10^{-6}$	$-1.45569 * 10^{-6}$	$1.04799 * 10^{-6}$
GRK4A von Kaps-Rentrop	$-3.17851 * 10^{-6}$	$-1.45568 * 10^{-6}$	$1.04799 * 10^{-6}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$-3.17846 * 10^{-6}$	$-1.4555 * 10^{-6}$	$1.04798 * 10^{-6}$
Van Veldhuizen [D-stabil]	$-3.17851 * 10^{-6}$	$-1.45567 * 10^{-6}$	$1.04798 * 10^{-6}$
eine L-stabile Methode	$-3.17846 * 10^{-6}$	$-1.45537 * 10^{-6}$	$1.04799 * 10^{-6}$

Verfahren	Far1PP	Fus3	Fus3PP
Shampine	$-1.45643 * 10^{-6}$	$1.27454 * 10^{-8}$	$-8.04681 * 10^{-7}$
GRK4T von Kaps-Rentrop	$-1.4404 * 10^{-6}$	$1.27067 * 10^{-8}$	$-8.05208 * 10^{-7}$
GRK4A von Kaps-Rentrop	$-1.44232 * 10^{-6}$	$1.27098 * 10^{-8}$	$-8.04073 * 10^{-7}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$-1.45644 * 10^{-6}$	$1.27147 * 10^{-8}$	$-8.04748 * 10^{-7}$
Van Veldhuizen [D-stabil]	$-1.44167 * 10^{-6}$	$1.27137 * 10^{-8}$	$-8.04782 * 10^{-7}$
eine L-stabile Methode	$-1.46859 * 10^{-6}$	$1.27045 * 10^{-8}$	$-8.04267 * 10^{-7}$

Verfahren	G	GαGDP	GαGTP
Shampine	$2.24929 * 10^{-6}$	$-3.99166 * 10^{-7}$	$-5.872464 * 10^{-3}$
GRK4T von Kaps-Rentrop	$2.24922 * 10^{-6}$	$-3.992 * 10^{-7}$	$-5.871882 * 10^{-3}$
GRK4A von Kaps-Rentrop	$2.24922 * 10^{-6}$	$-3.99224 * 10^{-7}$	$-5.872023 * 10^{-3}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$2.24928 * 10^{-6}$	$-3.99153 * 10^{-7}$	$-5.872462 * 10^{-3}$
Van Veldhuizen [D-stabil]	$2.24922 * 10^{-6}$	$-3.99178 * 10^{-7}$	$-5.871876 * 10^{-3}$
eine L-stabile Methode	$2.24928 * 10^{-6}$	$-3.99125 * 10^{-7}$	$-5.873018 * 10^{-3}$

Tabelle B.17: Hefe: Relativer Fehler bei Zeitschritt 200

Verfahren	G$\alpha\beta\gamma$	G$\beta\gamma$	H
Shampine	$1.42723 * 10^{-8}$	$-1.2545 * 10^{-7}$	$-2.36192 * 10^{-6}$
GRK4T von Kaps-Rentrop	$1.42724 * 10^{-8}$	$-1.25285 * 10^{-7}$	$-2.36202 * 10^{-6}$
GRK4A von Kaps-Rentrop	$1.42726 * 10^{-8}$	$-1.25539 * 10^{-7}$	$-2.36199 * 10^{-6}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$1.42722 * 10^{-8}$	$-1.25457 * 10^{-7}$	$-2.36193 * 10^{-6}$
Van Veldhuizen [D-stabil]	$1.42723 * 10^{-8}$	$-1.25438 * 10^{-7}$	$-2.36202 * 10^{-6}$
eine L-stabile Methode	$1.42722 * 10^{-8}$	$-1.25654 * 10^{-7}$	$-2.36193 * 10^{-6}$

Verfahren	I	K	L
Shampine	$-3.9163 * 10^{-7}$	$-5.22615 * 10^{-7}$	$1.33685 * 10^{-6}$
GRK4T von Kaps-Rentrop	$-3.91649 * 10^{-7}$	$-5.22631 * 10^{-7}$	$1.33679 * 10^{-6}$
GRK4A von Kaps-Rentrop	$-3.91663 * 10^{-7}$	$-5.22651 * 10^{-7}$	$1.33696 * 10^{-6}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$-3.91687 * 10^{-7}$	$-5.22671 * 10^{-7}$	$1.33682 * 10^{-6}$
Van Veldhuizen [D-stabil]	$-3.91651 * 10^{-7}$	$-5.22637 * 10^{-7}$	$1.33685 * 10^{-6}$
eine L-stabile Methode	$-3.9177 * 10^{-7}$	$-5.22756 * 10^{-7}$	$1.33684 * 10^{-6}$

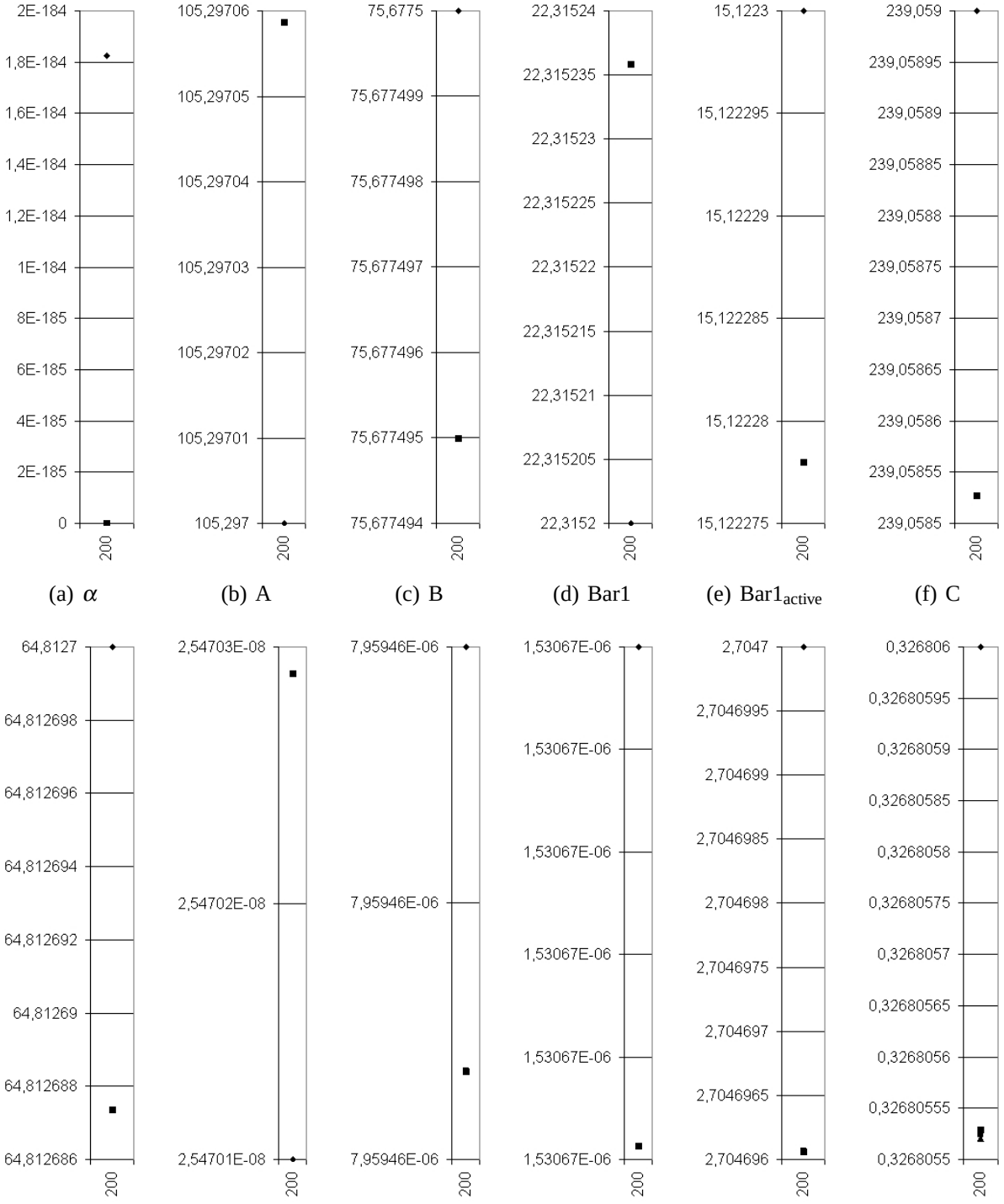
Verfahren	M	N	$Sst2_{active}$
Shampine	$1.1089 * 10^{-7}$	$1.32938 * 10^{-6}$	$4.8299 * 10^{-8}$
GRK4T von Kaps-Rentrop	$1.10853 * 10^{-7}$	$1.32936 * 10^{-6}$	$5.08778 * 10^{-8}$
GRK4A von Kaps-Rentrop	$1.10824 * 10^{-7}$	$1.32936 * 10^{-6}$	$5.02885 * 10^{-8}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$1.10891 * 10^{-7}$	$1.32938 * 10^{-6}$	$4.80766 * 10^{-8}$
Van Veldhuizen [D-stabil]	$1.10863 * 10^{-7}$	$1.32936 * 10^{-6}$	$5.07944 * 10^{-8}$
eine L-stabile Methode	$1.10896 * 10^{-7}$	$1.3294 * 10^{-6}$	$4.58877 * 10^{-8}$

Verfahren	Ste11	Ste12	$Ste12_{active}$
Shampine	$2.33857 * 10^{-6}$	$-9.27682 * 10^{-8}$	$-7.42467 * 10^{-7}$
GRK4T von Kaps-Rentrop	$2.33857 * 10^{-6}$	$-9.27682 * 10^{-8}$	$-7.42461 * 10^{-7}$
GRK4A von Kaps-Rentrop	$2.33857 * 10^{-6}$	$-9.27681 * 10^{-8}$	$-7.42487 * 10^{-7}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$2.33857 * 10^{-6}$	$-9.2768 * 10^{-8}$	$-7.42527 * 10^{-7}$
Van Veldhuizen [D-stabil]	$2.33857 * 10^{-6}$	$-9.27682 * 10^{-8}$	$-7.42463 * 10^{-7}$
eine L-stabile Methode	$2.33856 * 10^{-6}$	$-9.27676 * 10^{-8}$	$-7.42637 * 10^{-7}$

Verfahren	Ste2	$Ste2_{active}$	Ste20
Shampine	$-3.68996 * 10^{-6}$	-0.999999998	$-5.03028 * 10^{-8}$
GRK4T von Kaps-Rentrop	$-3.68996 * 10^{-6}$	-0.999999998	$-5.03031 * 10^{-8}$
GRK4A von Kaps-Rentrop	$-3.68996 * 10^{-6}$	-0.999999998	$-5.03031 * 10^{-8}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$-3.68996 * 10^{-6}$	-0.999999998	$-5.03031 * 10^{-8}$
Van Veldhuizen [D-stabil]	$-3.68996 * 10^{-6}$	-0.999999998	$-5.03032 * 10^{-8}$
eine L-stabile Methode	$-3.68997 * 10^{-6}$	-0.999999998	$-5.03028 * 10^{-8}$

Verfahren	Ste5	Ste7
Shampine	$2.33857 * 10^{-6}$	$7.84505 * 10^{-7}$
GRK4T von Kaps-Rentrop	$2.33857 * 10^{-6}$	$7.84535 * 10^{-7}$
GRK4A von Kaps-Rentrop	$2.33857 * 10^{-6}$	$7.8453 * 10^{-7}$
Van Veldhuizen [$\gamma = \frac{1}{2}$]	$2.33857 * 10^{-6}$	$7.84527 * 10^{-7}$
Van Veldhuizen [D-stabil]	$2.33857 * 10^{-6}$	$7.84526 * 10^{-7}$
eine L-stabile Methode	$2.33856 * 10^{-6}$	$7.84524 * 10^{-7}$

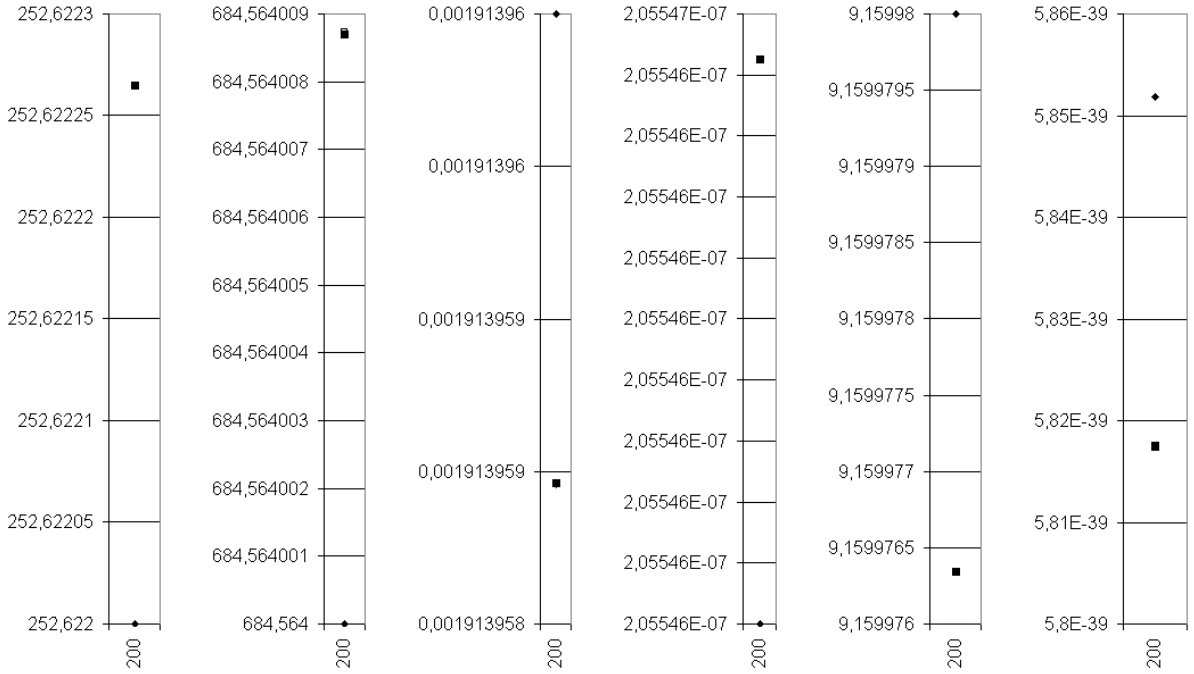
Tabelle B.18: Hefe: Relativer Fehler bei Zeitschritt 200 (Forts.)



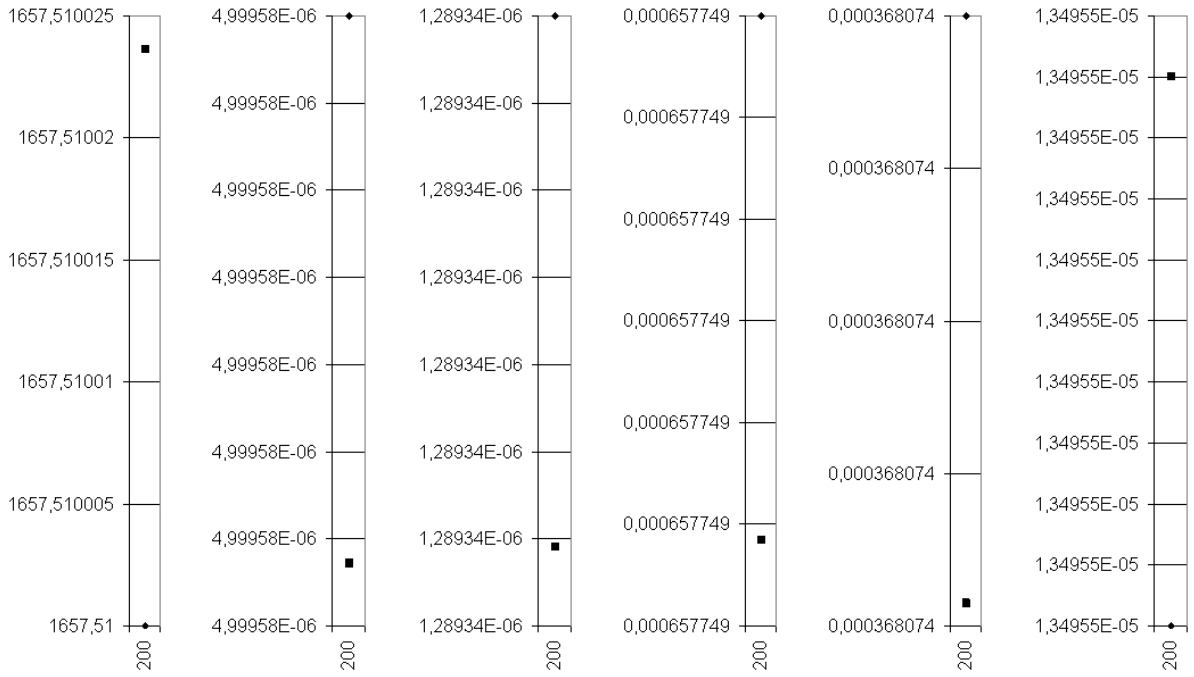
(a) α (b) A (c) B (d) Bar1 (e) $\text{Bar1}_{\text{active}}$ (f) C

(g) Cdc28 (h) D (i) E (j) F (k) Far1 (l) Far1PP

Abbildung B.15: Hefe: Absoluter Fehler bei Zeitschritt 200

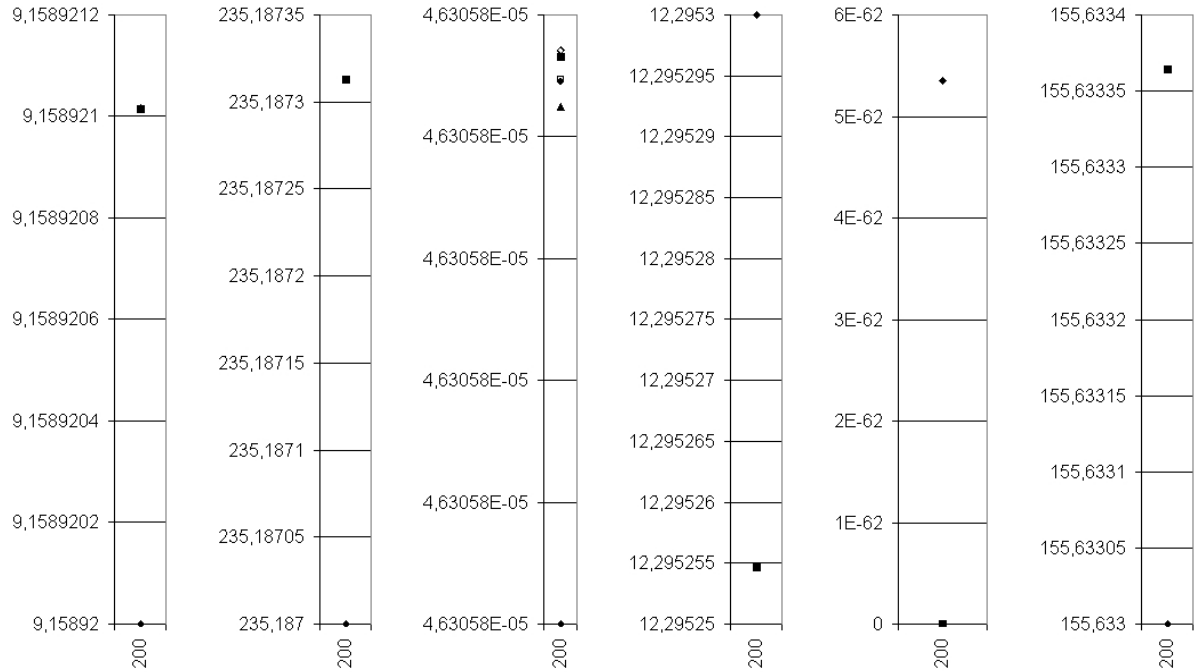


(a) Far1$_{\text{Ubiquitin}}$ (b) Fus3 (c) Fus3PP (d) G (e) GαGDP (f) GαGTP

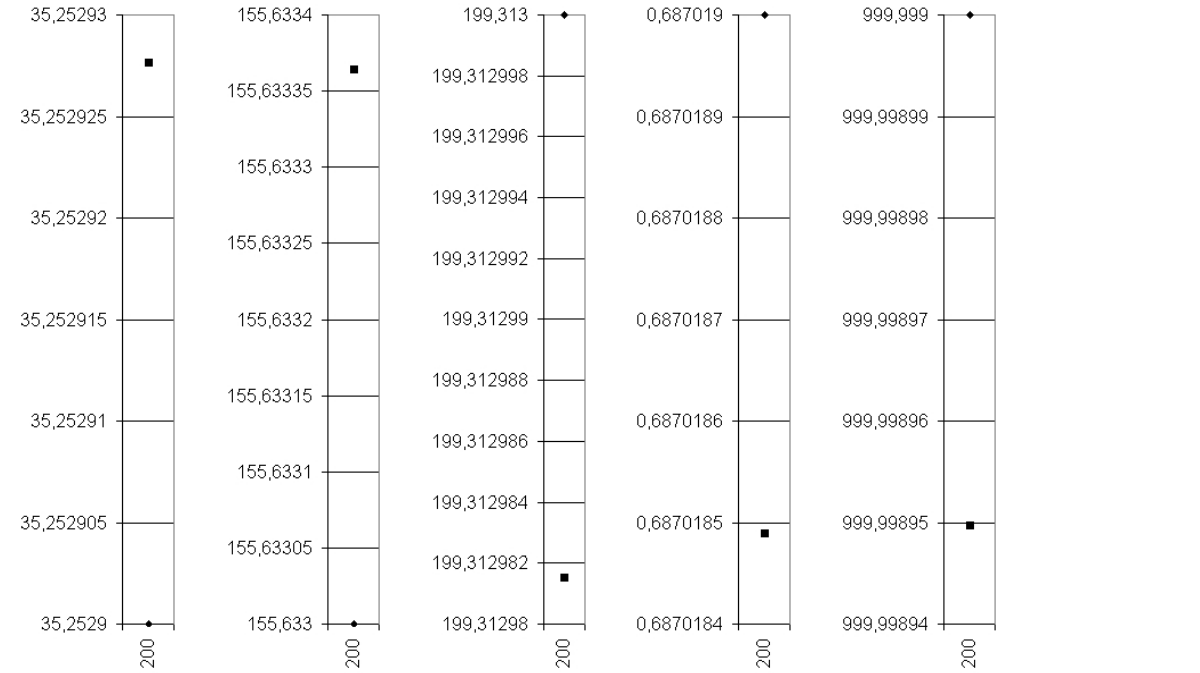


(g) G$\alpha\beta\gamma$ (h) G$\beta\gamma$ (i) H (j) I (k) K (l) L

Abbildung B.16: Hefe: Absoluter Fehler bei Zeitschritt 200 (Forts.)

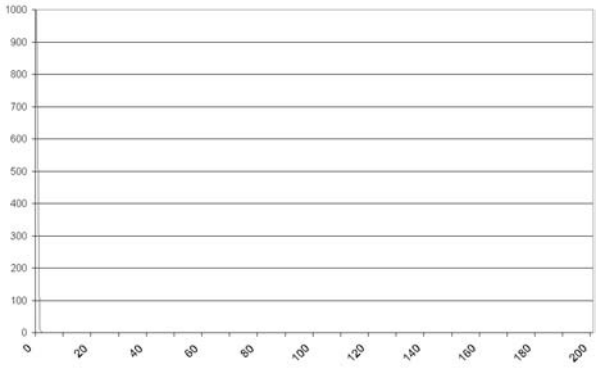


(a) M (b) N (c) $Sst2_{active}$ (d) Ste2 (e) $Ste2_{active}$ (f) Ste5

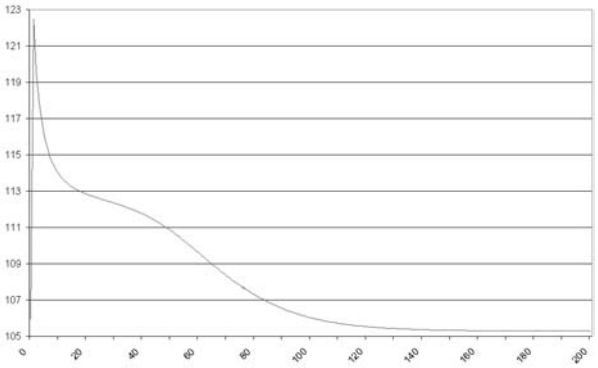


(g) Ste7 (h) Ste11 (i) Ste12 (j) $Ste12_{active}$ (k) Ste20

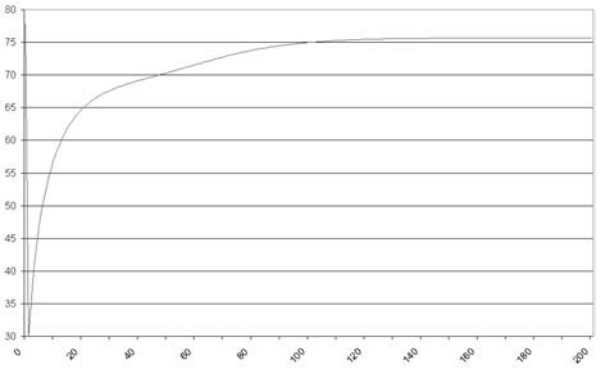
Abbildung B.17: Hefe: Absoluter Fehler bei Zeitschritt 200 (Forts.)



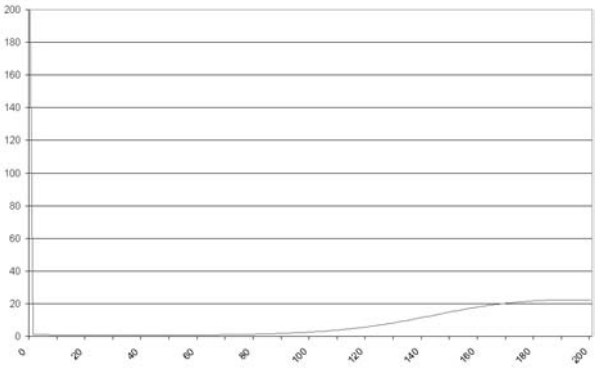
(a) α



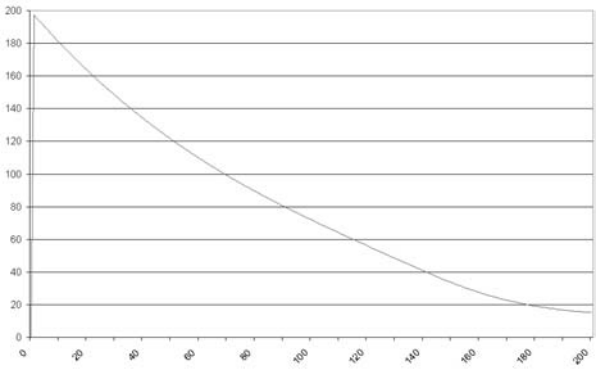
(b) A



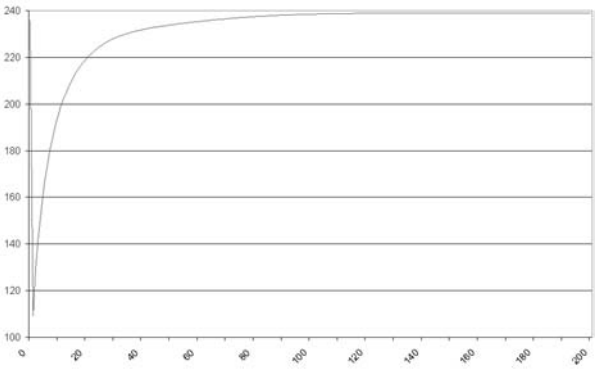
(c) B



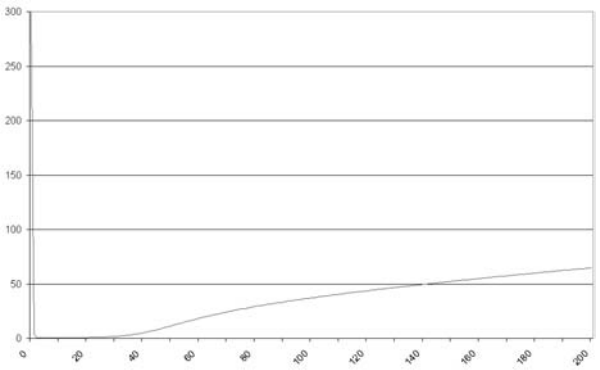
(d) Bar1



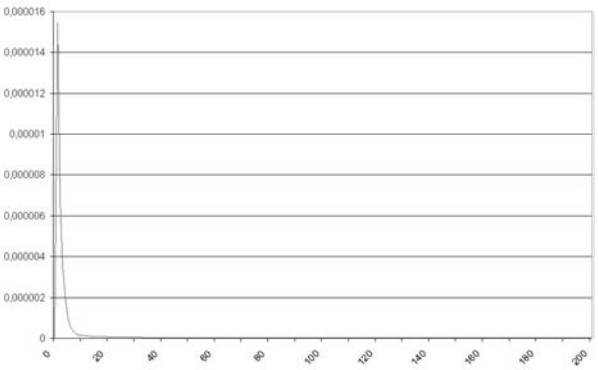
(e) $\text{Bar1}_{\text{active}}$



(f) C

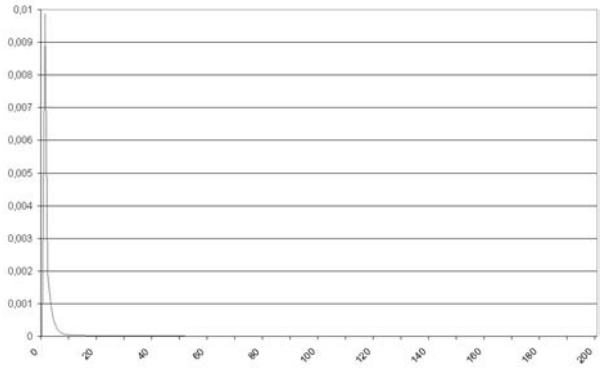


(g) Cdc28

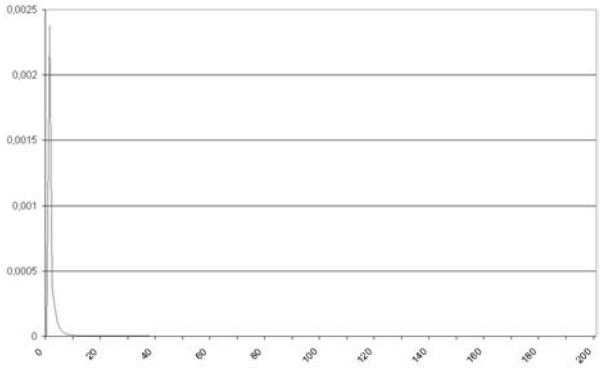


(h) D

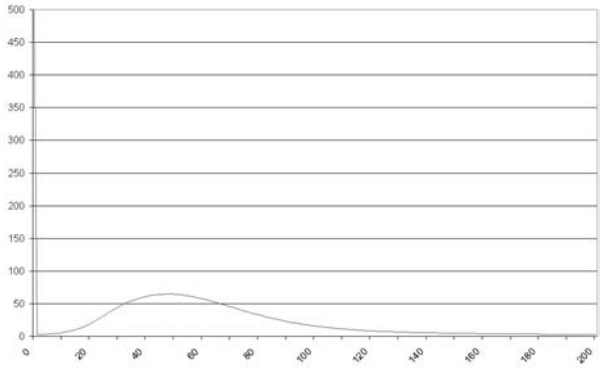
Abbildung B.18: Hefe: Kurvenverlaufsdiagramme



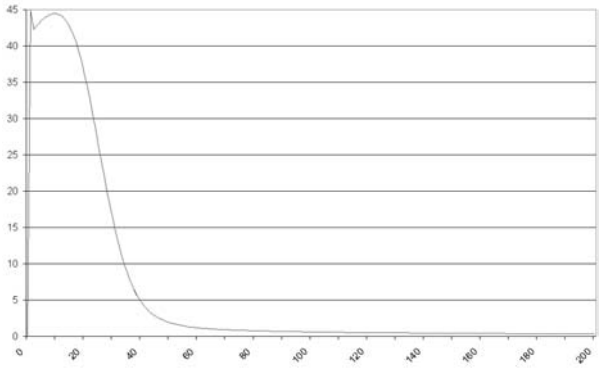
(a) E



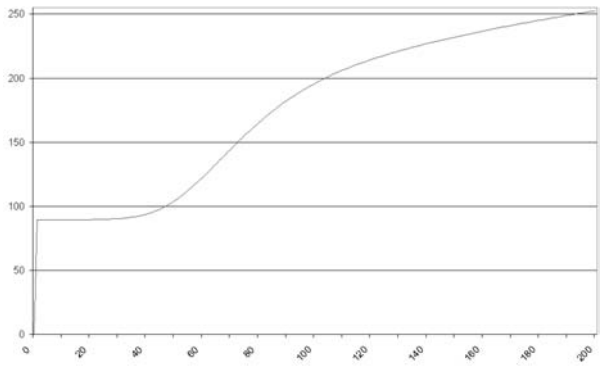
(b) F



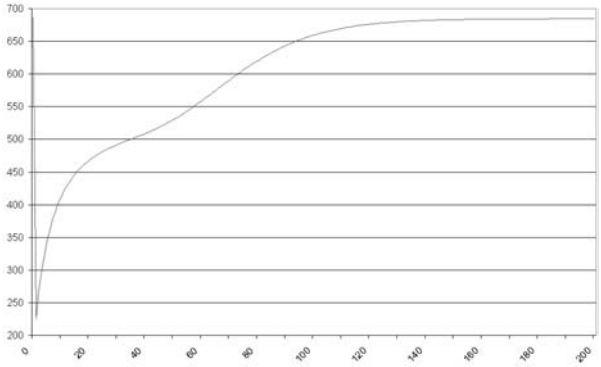
(c) Far1



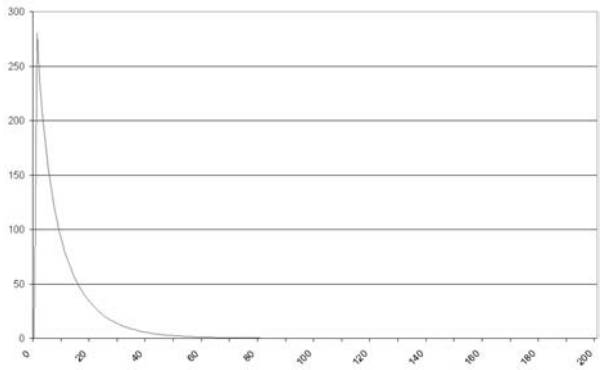
(d) Far1PP



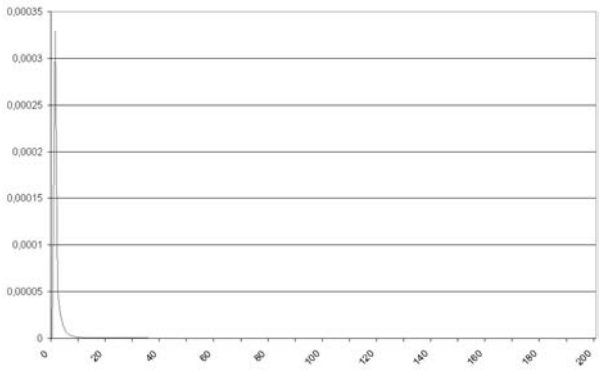
(e) $\text{Far1}_{\text{Ubiqiutin}}$



(f) Fus3

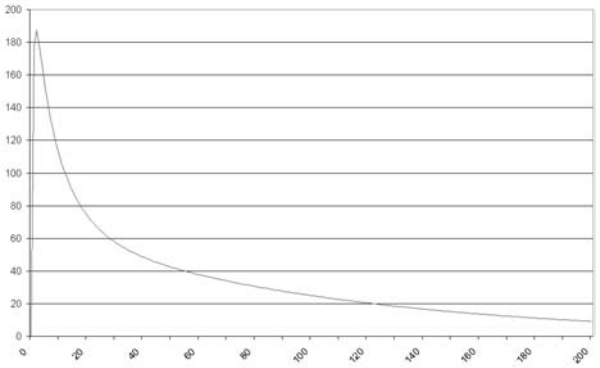


(g) Fus3PP

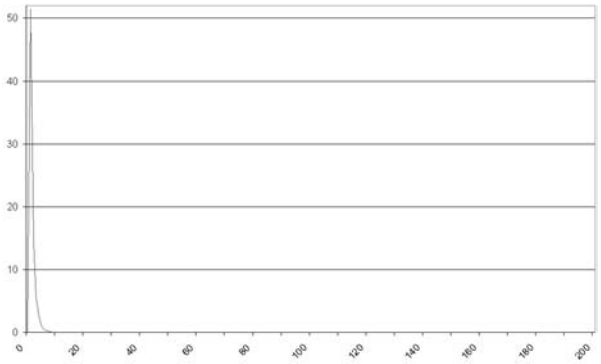


(h) G

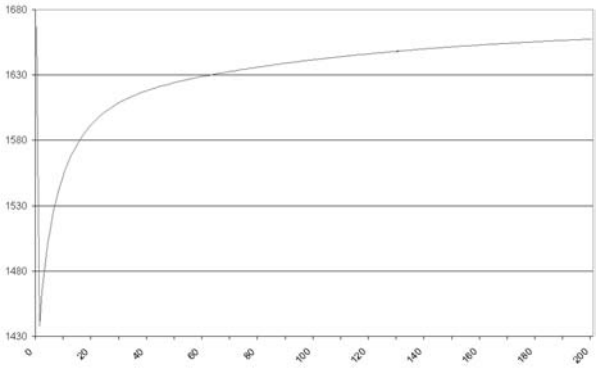
Abbildung B.19: Hefe: Kurvenverlaufsdiagramme (Forts.)



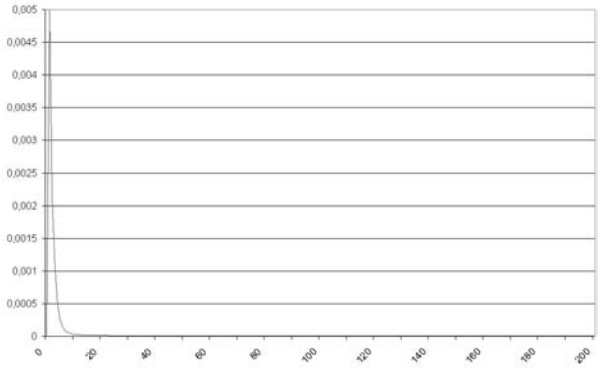
(a) GαGDP



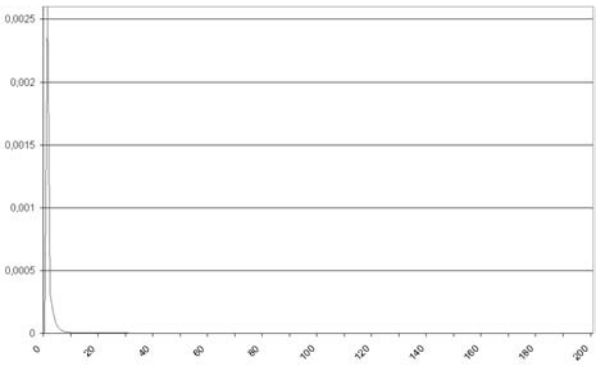
(b) GαGTP



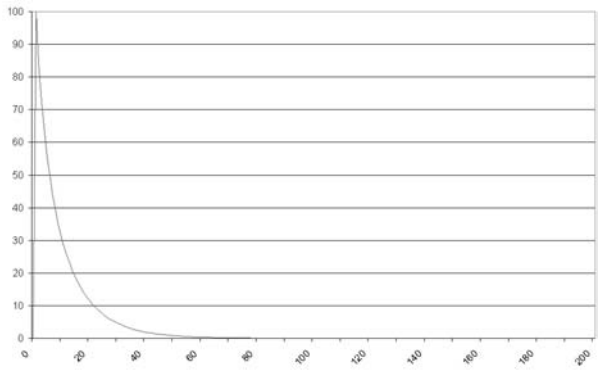
(c) G$\alpha\beta\gamma$



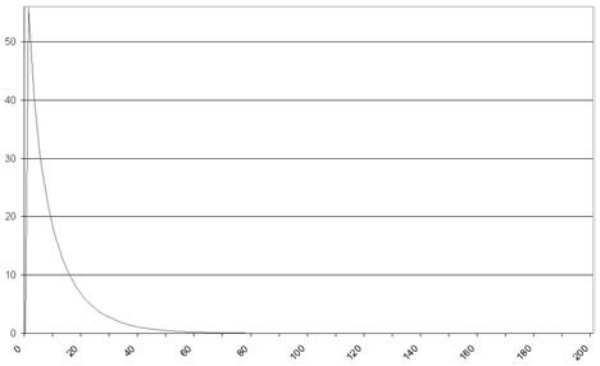
(d) G$\beta\gamma$



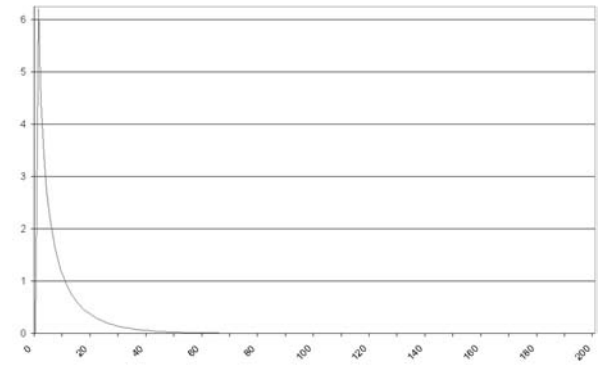
(e) H



(f) I

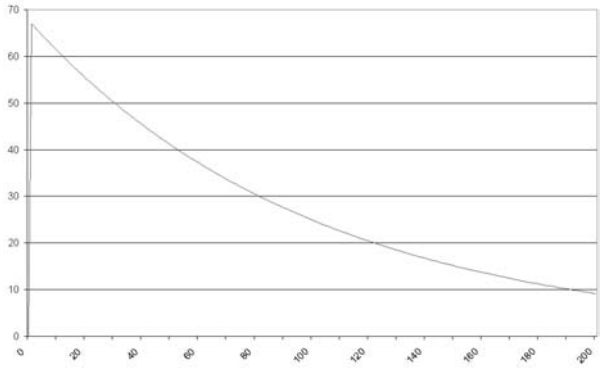


(g) K

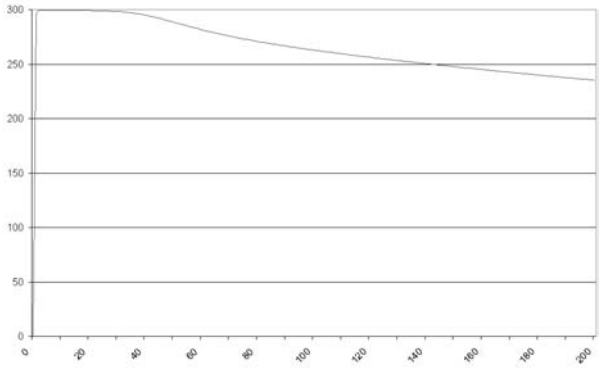


(h) L

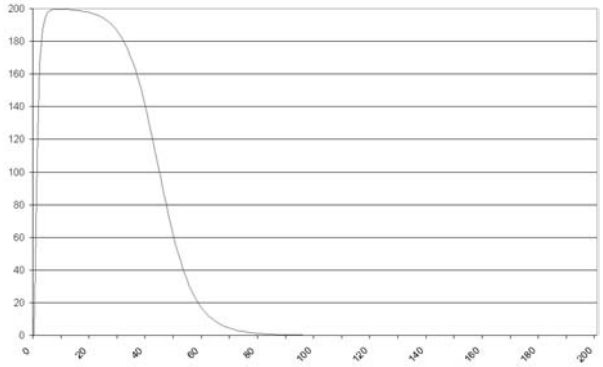
Abbildung B.20: Hefe: Kurvenverlaufsdiagramme (Forts.)



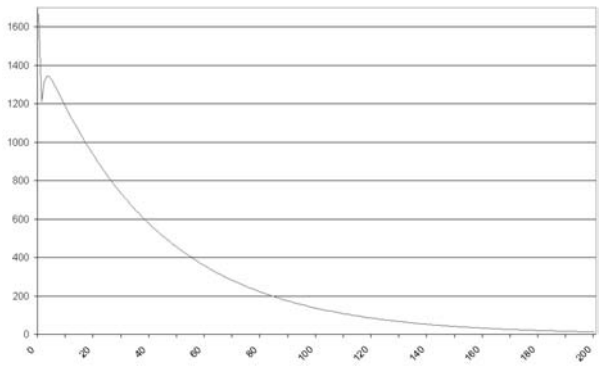
(a) M



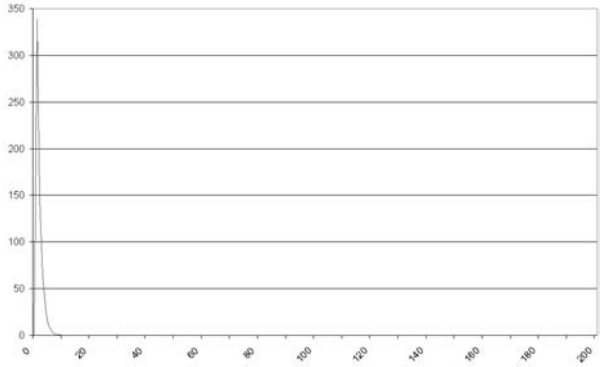
(b) N



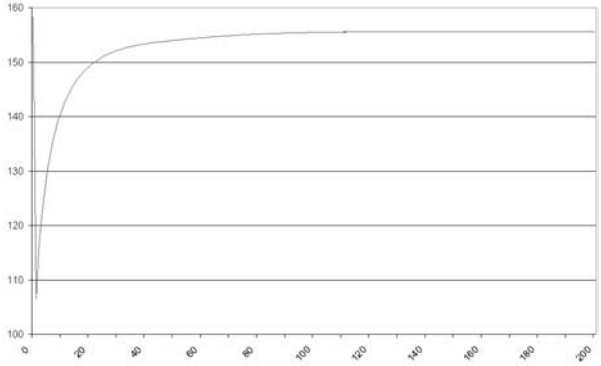
(c) $Sst2_{active}$



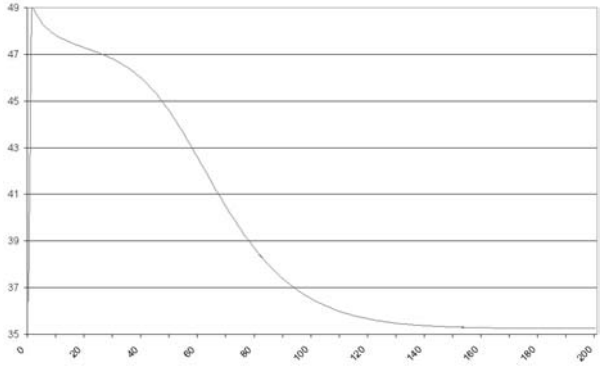
(d) Ste2



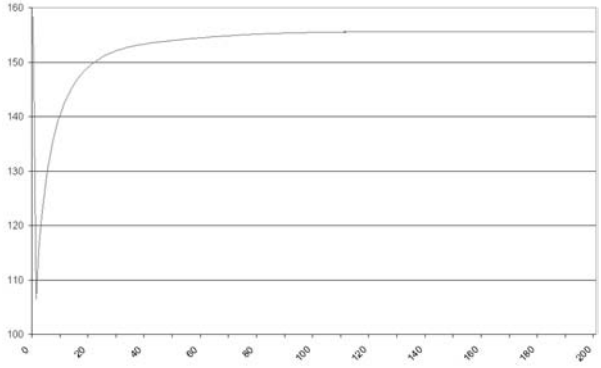
(e) $Ste2_{active}$



(f) Ste5

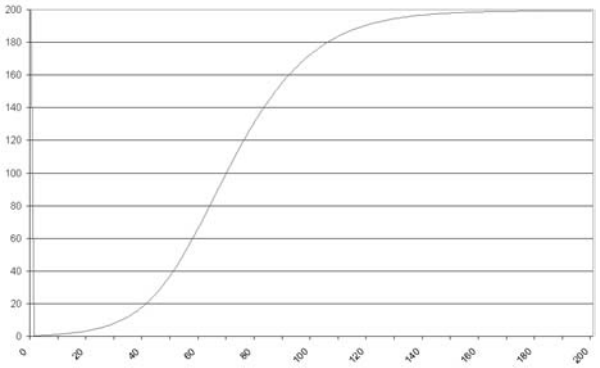


(g) Ste7

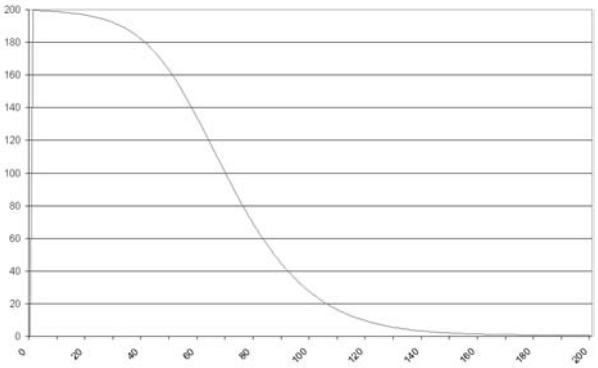


(h) Ste11

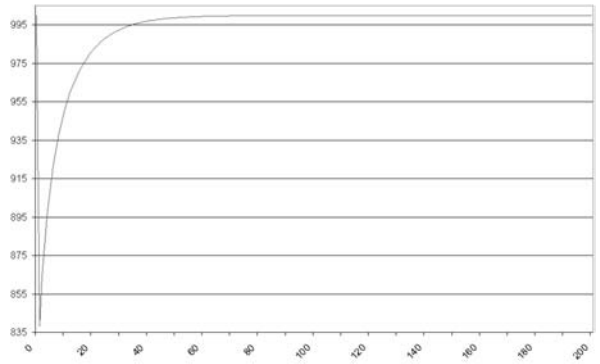
Abbildung B.21: Hefe: Kurvenverlaufsdiagramme (Forts.)



(a) Ste12

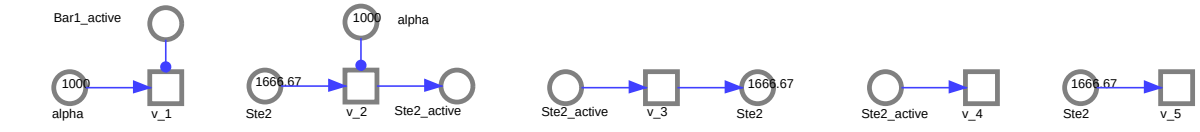


(b) $\text{Ste12}_{\text{active}}$

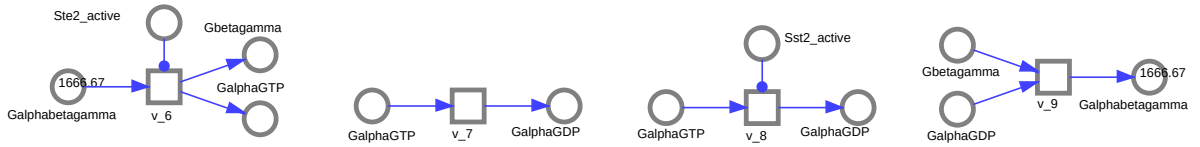


(c) Ste20

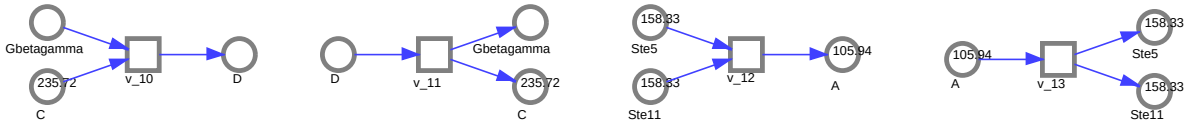
Abbildung B.22: Hefe: Kurvenverlaufsdiagramme (Forts.)



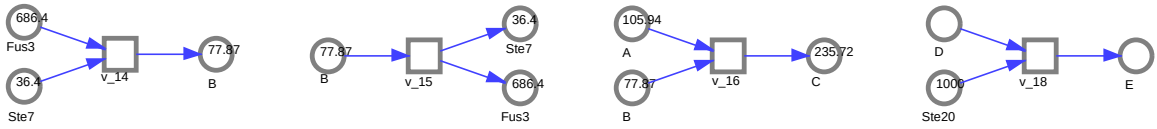
(a) r_1 (b) r_2 (c) r_3 (d) r_4 (e) r_5



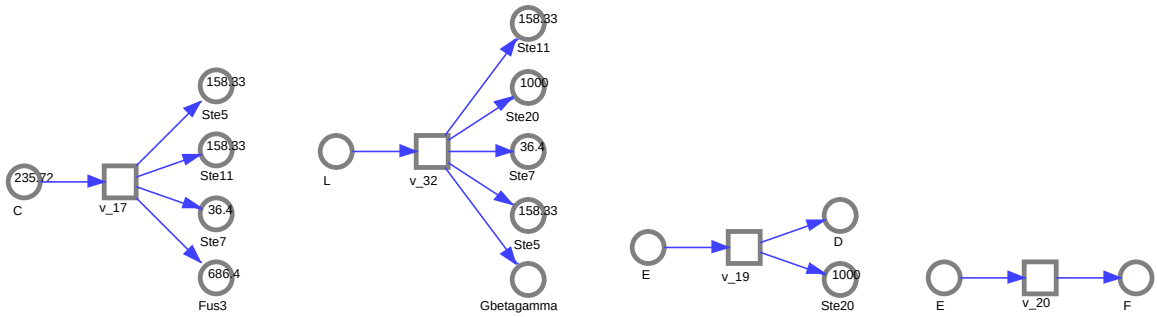
(f) r_6 (g) r_7 (h) r_8 (i) r_9



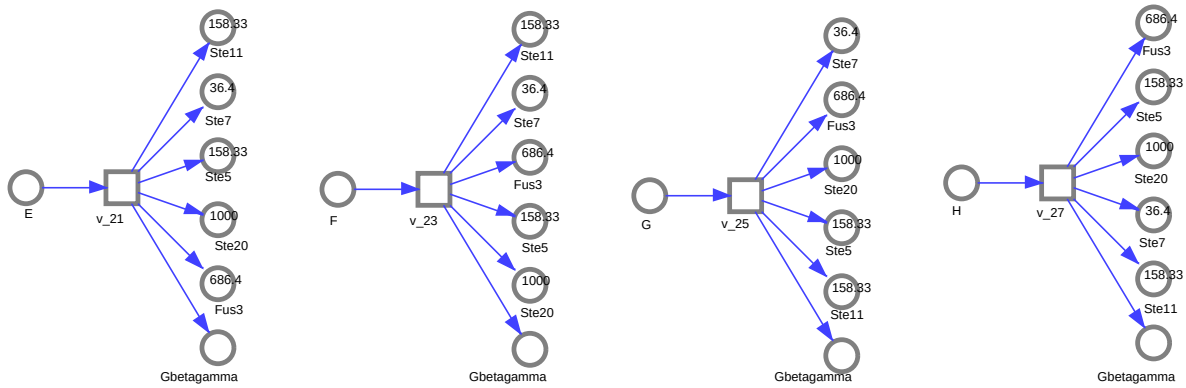
(j) r_{10} (k) r_{11} (l) r_{12} (m) r_{13}



(n) r_{14} (o) r_{15} (p) r_{16} (q) r_{18}



(r) r_{17} (s) r_{32} (t) r_{19} (u) r_{20}



(v) r_{21} (w) r_{23} (x) r_{25} (y) r_{27}

Abbildung B.23: Hefe: Unterstrukturen

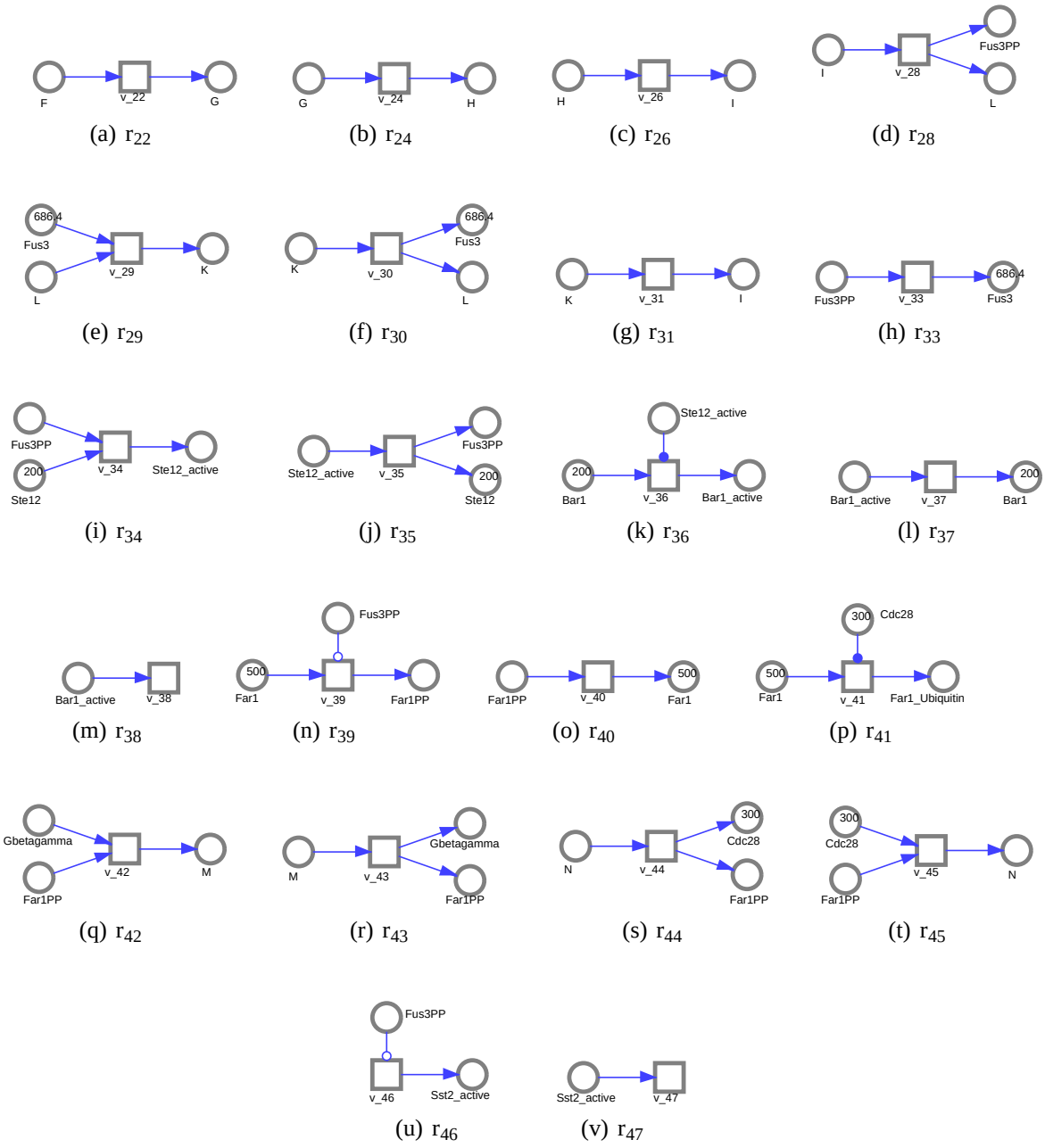


Abbildung B.24: Hefe: Unterstrukturen (Forts.)

α	=	1000	$Ste2_{active}$	=	0	k_{21}	=	5.0
A	=	105.94	Ste5	=	158.33	k_{22}	=	47.0
B	=	77.87	Ste7	=	36.4	k_{23}	=	5.0
Bar1	=	200	Ste11	=	158.33	k_{24}	=	345.0
$Bar1_{active}$	=	0	Ste12	=	200	k_{25}	=	5.0
C	=	235.72	$Ste12_{active}$	=	0	k_{26}	=	50.0
Cdc28	=	300	Ste20	=	1000	k_{27}	=	5.0
D	=	0	k_1	=	0.03	k_{28}	=	140.0
E	=	0	k_2	=	0.0012	k_{29}	=	10.0
F	=	0	k_3	=	0.6	k_{30}	=	1.0
Far1	=	500	k_4	=	0.24	k_{31}	=	250.0
Far1PP	=	0	k_5	=	0.024	k_{32}	=	5.0
$Far1_{Ubiquitin}$	=	0	k_6	=	0.0036	k_{33}	=	50.0
Fus3	=	686.4	k_7	=	0.24	k_{34}	=	18.0
Fus3PP	=	0	k_8	=	0.33	k_{35}	=	10.0
G	=	0	k_9	=	2000.0	k_{36}	=	0.1
GαGDP	=	0	k_{10}	=	0.1	k_{37}	=	0.1
GαGTP	=	0	k_{11}	=	5.0	k_{38}	=	0.01
G$\alpha\beta\gamma$	=	1666.67	k_{12}	=	1.0	k_{39}	=	18.0
G$\beta\gamma$	=	0	k_{13}	=	3.0	k_{40}	=	1.0
H	=	0	k_{14}	=	1.0	k_{41}	=	0.002
I	=	0	k_{15}	=	3.0	k_{42}	=	0.1
K	=	0	k_{16}	=	3.0	k_{43}	=	0.01
L	=	0	k_{17}	=	100.0	k_{44}	=	0.01
M	=	0	k_{18}	=	5.0	k_{45}	=	0.1
N	=	0	k_{19}	=	1.0	k_{46}	=	200.0
$Sst2_{active}$	=	0	k_{20}	=	10.0	k_{47}	=	1.0
Ste2	=	1666.67						

$$\frac{d\alpha}{dt} = -v_1$$
$$\frac{dA}{dt} = v_{12} - v_{13} - v_{16}$$
$$\frac{dB}{dt} = v_{14} - v_{15} - v_{16}$$
$$\frac{dBar1}{dt} = v_{37} - v_{36}$$
$$\frac{dBar1_{active}}{dt} = v_{36} - v_{37} - v_{38}$$
$$\frac{dC}{dt} = v_{11} + v_{16} - v_{10} - v_{17}$$
$$\frac{dCdc28}{dt} = v_{44} - v_{45}$$
$$\frac{dD}{dt} = v_{10} + v_{19} - v_{11} - v_{18}$$
$$\frac{dE}{dt} = v_{18} - v_{19} - v_{20} - v_{21}$$
$$\frac{dF}{dt} = v_{20} - v_{22} - v_{23}$$
$$\frac{dFar1}{dt} = v_{40} - v_{39} - v_{41}$$
$$\frac{dFar1_{Ubiquitin}}{dt} = v_{41}$$
$$\frac{dFus3PP}{dt} = v_{28} + v_{35} - v_{33} - v_{34}$$
$$\frac{dG}{dt} = v_{22} - v_{24} - v_{25}$$
$$\frac{dG\alpha GDP}{dt} = v_7 + v_8 - v_9$$
$$\frac{dG\alpha GTP}{dt} = v_6 - v_7 - v_8$$
$$\frac{dG\alpha\beta\gamma}{dt} = v_9 - v_6$$
$$\frac{dH}{dt} = v_{24} - v_{26} - v_{27}$$
$$\frac{dI}{dt} = v_{26} + v_{31} - v_{28}$$
$$\frac{dK}{dt} = v_{29} - v_{30} - v_{31}$$
$$\frac{dL}{dt} = v_{30} + v_{28} - v_{29} - v_{32}$$
$$\frac{dM}{dt} = v_{42} - v_{43}$$
$$\frac{dN}{dt} = v_{45} - v_{44}$$
$$\frac{dSst2_{active}}{dt} = v_{46} - v_{47}$$
$$\frac{dSte2}{dt} = v_3 - v_2 - v_5$$
$$\frac{dSte2_{active}}{dt} = v_2 - v_3 - v_4$$
$$\frac{dSte12}{dt} = v_{35} - v_{34}$$
$$\frac{dSte12_{active}}{dt} = v_{34} - v_{35}$$

$$\frac{dFar1PP}{dt} = v_{39} + v_{43} + v_{44} - v_{40} - v_{42} - v_{45}$$
$$\frac{dFus3}{dt} = v_{17} + v_{23} + v_{25} + v_{15} + v_{27} + v_{30} + v_{33} + v_{21} - v_{14} - v_{29}$$
$$\frac{dG\beta\gamma}{dt} = v_6 + v_{11} + v_{25} + v_{27} + v_{43} + v_{21} + v_{23} + v_{32} - v_9 - v_{10} - v_{42}$$
$$\frac{dSte5}{dt} = v_{13} + v_{17} + v_{21} + v_{23} + v_{25} + v_{27} + v_{32} - v_{12}$$
$$\frac{dSte7}{dt} = v_{15} + v_{17} + v_{21} + v_{23} + v_{25} + v_{32} + v_{27} - v_{14}$$
$$\frac{dSte11}{dt} = v_{13} + v_{17} + v_{21} + v_{23} + v_{25} + v_{32} + v_{27} - v_{12}$$
$$\frac{dSte20}{dt} = v_{25} + v_{19} + v_{27} + v_{32} + v_{21} + v_{23} - v_{18}$$

$$
\begin{aligned}
v_1 &= \alpha * \mathrm{Bar1}_{\mathrm{active}} * k_1 \\
v_2 &= \mathrm{Ste2} * \alpha * k_2 \\
v_3 &= \mathrm{Ste2}_{\mathrm{active}} * k_3 \\
v_4 &= \mathrm{Ste2}_{\mathrm{active}} * k_4 \\
v_5 &= \mathrm{Ste2} * k_5 \\
v_6 &= \mathrm{Ste2}_{\mathrm{active}} * G\alpha\beta\gamma * k_6 \\
v_7 &= G\alpha\mathrm{GTP} * k_7 \\
v_8 &= G\alpha\mathrm{GTP} * \mathrm{Sst2}_{\mathrm{active}} * k_8 \\
v_9 &= G\alpha\mathrm{GDP} * G\beta\gamma * k_9 \\
v_{10} &= G\beta\gamma * C * k_{10} \\
v_{11} &= D * k_{11} \\
v_{12} &= \mathrm{Ste5} * \mathrm{Ste11} * k_{12} \\
v_{13} &= A * k_{13} \\
v_{14} &= \mathrm{Ste7} * \mathrm{Fus3} * k_{14} \\
v_{15} &= B * k_{15} \\
v_{16} &= A * B * k_{16} \\
v_{17} &= C * k_{17} \\
v_{18} &= D * \mathrm{Ste20} * k_{18} \\
v_{19} &= E * k_{19} \\
v_{20} &= E * k_{20} \\
v_{21} &= E * k_{21} \\
v_{22} &= F * k_{22} \\
v_{23} &= F * k_{23} \\
v_{24} &= G * k_{24} \\
v_{25} &= G * k_{25} \\
v_{26} &= H * k_{26} \\
v_{27} &= H * k_{27} \\
v_{28} &= I * k_{28} \\
v_{29} &= L * \mathrm{Fus3} * k_{29} \\
v_{30} &= K * k_{30} \\
v_{31} &= K * k_{31} \\
v_{32} &= L * k_{32} \\
v_{33} &= \mathrm{Fus3PP} * k_{33} \\
v_{34} &= \mathrm{Ste12} * \mathrm{Fus3PP} * k_{34} \\
v_{35} &= \mathrm{Ste12}_{\mathrm{active}} * k_{35} \\
v_{36} &= \mathrm{Ste12}_{\mathrm{active}} * \mathrm{Bar1} * k_{36} \\
v_{37} &= \mathrm{Bar1}_{\mathrm{active}} * k_{37} \\
v_{38} &= \mathrm{Bar1}_{\mathrm{active}} * k_{38} \\
v_{39} &= \mathrm{Far1} * \frac{\mathrm{Fus3PP}^2}{100^2 + \mathrm{Fus3PP}^2} * k_{39} \\
v_{40} &= \mathrm{Far1PP} * k_{40} \\
v_{41} &= \mathrm{Far1} * \mathrm{Cdc28} * k_{41} \\
v_{42} &= G\beta\gamma * \mathrm{Far1PP} * k_{42} \\
v_{43} &= M * k_{43} \\
v_{44} &= N * k_{44} \\
v_{45} &= \mathrm{Far1PP} * \mathrm{Cdc28} * k_{45} \\
v_{46} &= \frac{\mathrm{Fus3PP}^2}{4^2 + \mathrm{Fus3PP}^2} * k_{46} \\
v_{47} &= \mathrm{Sst2}_{\mathrm{active}} * k_{47}
\end{aligned}
$$

$$
\begin{aligned}
v_1 &: \alpha \rightarrow \\
v_2 &: \mathrm{Ste2} \rightarrow \mathrm{Ste2_{active}} \\
v_3 &: \mathrm{Ste2_{active}} \rightarrow \mathrm{Ste2} \\
v_4 &: \mathrm{Ste2_{active}} \rightarrow \\
v_5 &: \mathrm{Ste2} \rightarrow \\
v_6 &: \mathrm{G}\alpha\beta\gamma \rightarrow \mathrm{G}\alpha\mathrm{GTP} + \mathrm{G}\beta\gamma \\
v_7 &: \mathrm{G}\alpha\mathrm{GTP} \rightarrow \mathrm{G}\alpha\mathrm{GDP} \\
v_8 &: \mathrm{G}\alpha\mathrm{GTP} \rightarrow \mathrm{G}\alpha\mathrm{GDP} \\
v_9 &: \mathrm{G}\beta\gamma + \mathrm{G}\alpha\mathrm{GDP} \rightarrow \mathrm{G}\alpha\beta\gamma \\
v_{10} &: \mathrm{G}\beta\gamma + \mathrm{C} \rightarrow \mathrm{D} \\
v_{11} &: \mathrm{D} \rightarrow \mathrm{G}\beta\gamma + \mathrm{C} \\
v_{12} &: \mathrm{Ste5} + \mathrm{Ste11} \rightarrow \mathrm{A} \\
v_{13} &: \mathrm{A} \rightarrow \mathrm{Ste11} + \mathrm{Ste5} \\
v_{14} &: \mathrm{Fus3} + \mathrm{Ste7} \rightarrow \mathrm{B} \\
v_{15} &: \mathrm{B} \rightarrow \mathrm{Ste7} + \mathrm{Fus3} \\
v_{16} &: \mathrm{B} + \mathrm{A} \rightarrow \mathrm{C} \\
v_{17} &: \mathrm{C} \rightarrow \mathrm{Ste5} + \mathrm{Ste11} + \mathrm{Ste7} + \mathrm{Fus3} \\
v_{18} &: \mathrm{D} + \mathrm{Ste20} \rightarrow \mathrm{E} \\
v_{19} &: \mathrm{E} \rightarrow \mathrm{D} + \mathrm{Ste20} \\
v_{20} &: \mathrm{E} \rightarrow \mathrm{F} \\
v_{21} &: \mathrm{E} \rightarrow \mathrm{Ste5} + \mathrm{Ste11} + \mathrm{Ste7} + \mathrm{G}\beta\gamma + \mathrm{Ste20} + \mathrm{Fus3} \\
v_{22} &: \mathrm{F} \rightarrow \mathrm{G} \\
v_{23} &: \mathrm{F} \rightarrow \mathrm{Ste5} + \mathrm{Ste11} + \mathrm{Ste7} + \mathrm{Fus3} + \mathrm{G}\beta\gamma + \mathrm{Ste20} \\
v_{24} &: \mathrm{G} \rightarrow \mathrm{H} \\
v_{25} &: \mathrm{G} \rightarrow \mathrm{G}\beta\gamma + \mathrm{Ste20} + \mathrm{Ste5} + \mathrm{Ste11} + \mathrm{Ste7} + \mathrm{Fus3} \\
v_{26} &: \mathrm{H} \rightarrow \mathrm{I} \\
v_{27} &: \mathrm{H} \rightarrow \mathrm{Ste20} + \mathrm{G}\beta\gamma + \mathrm{Ste5} + \mathrm{Fus3} + \mathrm{Ste7} + \mathrm{Ste11} \\
v_{28} &: \mathrm{I} \rightarrow \mathrm{Fus3PP} + \mathrm{L} \\
v_{29} &: \mathrm{L} + \mathrm{Fus3} \rightarrow \mathrm{K} \\
v_{30} &: \mathrm{K} \rightarrow \mathrm{L} + \mathrm{Fus3} \\
v_{31} &: \mathrm{K} \rightarrow \mathrm{I} \\
v_{32} &: \mathrm{L} \rightarrow \mathrm{Ste11} + \mathrm{Ste20} + \mathrm{Ste7} + \mathrm{G}\beta\gamma + \mathrm{Ste5} \\
v_{33} &: \mathrm{Fus3PP} \rightarrow \mathrm{Fus3} \\
v_{34} &: \mathrm{Ste12} + \mathrm{Fus3PP} \rightarrow \mathrm{Ste12_{active}} \\
v_{35} &: \mathrm{Ste12_{active}} \rightarrow \mathrm{Ste12} + \mathrm{Fus3PP} \\
v_{36} &: \mathrm{Bar1} \rightarrow \mathrm{Bar1_{active}} \\
v_{37} &: \mathrm{Bar1_{active}} \rightarrow \mathrm{Bar1} \\
v_{38} &: \mathrm{Bar1_{active}} \rightarrow \\
v_{39} &: \mathrm{Far1} \rightarrow \mathrm{Far1PP} \\
v_{40} &: \mathrm{Far1PP} \rightarrow \mathrm{Far1} \\
v_{41} &: \mathrm{Far1} \rightarrow \mathrm{Far1_{Ubiquitin}} \\
v_{42} &: \mathrm{Far1PP} + \mathrm{G}\beta\gamma \rightarrow \mathrm{M} \\
v_{43} &: \mathrm{M} \rightarrow \mathrm{Far1PP} + \mathrm{G}\beta\gamma \\
v_{44} &: \mathrm{N} \rightarrow \mathrm{Cdc28} + \mathrm{Far1PP} \\
v_{45} &: \mathrm{Cdc28} + \mathrm{Far1PP} \rightarrow \mathrm{N} \\
v_{46} &: \rightarrow \mathrm{Sst2_{active}} \\
v_{47} &: \mathrm{Sst2_{active}} \rightarrow
\end{aligned}
$$

Literaturverzeichnis

[Aik85] AIKEN, Richard C.: *Stiff Computation*. New York, Oxford : Oxford University Press, 1985

[Aul04] AULBACH, Bernd: *Gewöhnliche Differentialgleichungen*. Heidelberg : Spektrum Akademischer Verlag, 2004

[Bau96] BAUMGARTEN, Bernd: *Petri-Netze*. Berlin, Heidelberg : Spektrum Akademischer Verlag, 1996

[CSK+03] CHO, Kwang-Hyun ; SHIN, Sung-Young ; KIM, Hyun-Woo ; WOLKENHAUER, Olaf ; MCFERRAN, Brian ; KOLCH, Walter: Mathematical modeling of the influence of RKIP on the ERK signaling pathway. In: *Lecture Notes in Computer Science* 2602 (2003), S. 127–141

[DA87] DAVID, Rene ; ALLA, Hassane: Continuous Petri Nets. In: *Proc. of the 8th European Workshop on Application and Theory of Petri Nets, Saragossa, Spain* (1987), S. 275–294

[DA92] DAVID, Rene ; ALLA, Hassane: *Petri Nets and Grafcet*. London : Prentice Hall, 1992

[DA05] DAVID, Rene ; ALLA, Hassane: *Discrete, Continuous, and Hybrid Petri Nets*. Berlin, Heidelberg : Springer, 2005

[Fie04] FIEBER, Markus: *Entwurf und Implementierung eines generischen, adaptiven Werkzeugs zur Arbeit mit Graphen*, BTU Cottbus, Diplomarbeit, 2004

[Gan05] GANDER, Walter: ODE45 in Matlab, Materialien zur Vorlesung „Wissenschaftliches Rechnen“, ETH Zürich, `http://www.icos.ethz.ch/education/courses/wissenschaftliches_rechnen/matlabode45.pdf`. (Sommersemester 2005)

[GC] GIUA, Alessandro ; CORONA, Daniele: Bibliography on hybrid petri nets, `http://bode.diee.unica.it/~hpn/`.

[Gri72] GRIGORIEFF, Rolf D.: *Numerik gewöhnlicher Differentialgleichungen, Band 1*. Stuttgart : Teubner, 1972

[Gri77] GRIGORIEFF, Rolf D.: *Numerik gewöhnlicher Differentialgleichungen, Band 2*. Stuttgart : Teubner, 1977

[Hai] HAIRER, Ernst: Fortran and Matlab Codes, http://www.unige.ch/math/folks/hairer/software.html.

[Han98] HANKE, Martin: Numerische Mathematik II, Universität Karlsruhe, http://www.ubka.uni-karlsruhe.de/cgi-bin/psgunzip/1997/mathematik/12/12.pdf. (Wintersemester 1997/1998)

[Jun04] JUNKER, Bjørn H.: *Potato*, Institut für Pflanzengenetik und Kulturpflanzenforschung Gatersleben, Diss., 2004

[Kho00] KHOLODENKO, Boris N.: Negative feedback and ultrasensitivity can bring about oscillations in the mitrogen-activated protein kinase cascades. In: *Eur. J. Biochem.* 267 (2000), S. 1583–1588

[KK04] KOFAHL, Bente ; KLIPP, Edda: Modelling the dynamics of the yeast pheromone pathway. In: *Whiley InterScience* 21 (2004), S. 831–850

[KSLR00] KNOBLAUCH, Hans ; SCHUSTER, Herbert ; LUFT, Friedrich C. ; REICH, Jens: A pathway model of lipid metabolism to predict the effect of genetic variability on lipid levels. In: *J Mol Med* 78 (2000), S. 507–515

[Lon05] LONITZ, Kristina: *Hybrid Systems Modelling in Engineering & Life Sciences*, Universität Koblenz-Landau, Diplomarbeit, 2005

[Men] MENDES, Pedro: Gepasi, http://www.gepasi.org/.

[MK] MENDES, Pedro ; KUMMER, Ursula: Copasi, http://www.copasi.org/.

[MK98] MENDES, Pedro ; KELL, Douglas B.: Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. In: *Bioinformatics* 14 (1998), S. 869–883

[MMN] MATSUNO, Hiroshi ; MIYANO, Satoru ; NAGASAKI, Masao: Genomic Object Net, http://www.GenomicObject.net/.

[Mor] MOREAU, Jean-Pierre: Numerical Analysis, http://perso.wanadoo.fr/jean-pierre.moreau/.

[Pet62] PETRI, Carl A.: *Kommunikation mit Automaten*, Rhein.-Westf. Institut für Instr. Mathematik an der Universität Bonn, Schrift Nr. 2, Diss., 1962

[PNW] Petri nets world, http://www.informatik.uni-hamburg.de/TGI/PetriNets/.

[Rei91] REISIG, Wolfgang: *Petrinetze, Eine Einführung*. Berlin, Heidelberg, New York, Tokyo : Springer Verlag, 1991

[SR05] SILVA, Manuel ; RECALDE, Laura: Continuization of Timed Petri Nets: From Performance Evaluation to Observation and Control. In: *Lecture Notes in Computer Science* 3536 (2005), S. 26–47

[Sta90] STARKE, Peter H.: *Analyse von Petri-Netz-Modellen*. Stuttgart : Teubner Verlag, 1990

[Sta03] STARKE, Peter. H.: Integrated Net Analyzer INA, `http://www.informatik.hu-berlin.de/lehrstuehle/automaten/ina/`. (2003)

[SW95] STREHMEL, Karl ; WEINER, Rüdiger: *Numerik gewöhnlicher Differentialgleichungen*. Stuttgart : Teubner, 1995

[Wan77] WANNER, G: On the integration of stiff differential equations. In: *Numerical Analysis, ISNM* Vol. 37 (1977), S. 209–226

[Wei84] WEISSINGER, Johannes: *Vorlesungen zur höheren Mathematik, Band 3*. Mannheim : B.I.-Wissenschaftsverlag, 1984

[WS79] WERNER, Helmut ; SCHABACK, Robert: *Praktische Mathematik II*. Berlin, Heidelberg : Springer, 1979