



Universität Ulm
Fakultät für Mathematik und Wirtschaftswissenschaften
Mai 2002

Diplomarbeit

Multirate-Verfahren für Differential-Algebraische Netzwerkgleichungen

eingereicht von :

Michael Striebel
geb. in Leutkirch im Allgäu

Betreuer:

PD Dr. Michael Günther

Inhaltsverzeichnis

Einleitung	1
1 Mathematische Modellbildung	3
1.1 Netzwerkansatz	3
1.2 Grundbegriffe aus der Schaltungstheorie	4
1.3 Modifizierte Knotenanalyse	7
1.4 Differential-Algebraische Netzwerkgleichungen	9
1.5 Multirate-Eigenschaft elektrischer Schaltungen	15
2 Multirate-Verfahren	17
2.1 Multirate-Verfahren für Systeme in Zustandsform	17
2.2 Multirate-Verfahren für Deskriptorsysteme	24
3 Verfahrenskonstruktion	41
3.1 Konsistenzordnung und Grundannahmen	43
3.2 Taylorentwicklung für partitionierte Index-1-DAEs	44
3.3 Konsistenz im compound-Schritt	55
3.4 Konsistenz der späteren Mikroschritte und dense-output	67
3.5 Konvergenz und Stabilität	73
4 MDAE– ein Multirate-DAE-Verfahren	79
4.1 MDAE-Koeffizientensatz	80
4.2 Nichtautonome Probleme und Koeffizientensatz	83
4.3 Schrittweitensteuerung und Implementierung	87
4.4 Numerischer Test	89
Zusammenfassung und Ausblick	92
Anhang	94
Literaturverzeichnis	104
Danksagung	107

Einleitung

Vom heimischen Herd bis zur NASA-Mission auf dem Mars finden sich im modernen Leben elektrische Schaltungen und übernehmen die unterschiedlichsten Aufgaben. In der digitalen Armbanduhr etwa fungieren elektrische Schwingkreise als Taktgeber, im Rundfunkgerät werden eingehende Signale verstärkt, gefiltert, weitergeleitet. So verschieden die Funktionsweisen elektrischer Schaltungen auch sind, ihre Ausgangssignale liegen immer in Form von Spannungsabfällen vor.

Bevor neu entwickelte Schaltungen ihrer Bestimmung zugeführt werden, muss geklärt werden, ob sie die Anforderungen erfüllen, die sie erfüllen müssen. D.h. es ist anhand von Tests zu untersuchen, ob die Ausgangssignale den erwarteten entsprechen.

Nun sind moderne elektrische Netzwerke so komplex, dass es unmöglich wäre, ihr quantitatives und qualitatives Verhalten am realen Prototypen zu studieren. Die **mathematische Modellbildung** stellt zusammen mit der **numerischen Simulation** ein Werkzeug dar, die Funktionsweise eines Netzwerkes vor dessen tatsächlicher realer Konstruktion zu bestimmen und somit über seine Umsetzung zu entscheiden.

Die Aufgabe, Netzwerktopologien – die Verknüpfungen unterschiedlicher Bauteile – auf ein mathematisches Modell abzubilden, übernehmen hierbei industrielle Simulationspakete. Bei dieser automatischen Konstruktion von Netzwerkgleichungen wird aufgrund der Effizienz nicht auf eine möglichst geringe Anzahl von Unbekannten geachtet. So ergeben sich durch eine redundante Modellbildung die Netzwerkgleichungen als ein **differential-algebraischen System**, das das transiente, d.h. das zeitliche Verhalten der zugrundeliegenden Schaltung beschreibt. Die Simulation besteht nun aus der Lösung dieses Systems, das Differentialgleichungen mit algebraischen Gleichungen verbindet.

Betrachtet man eine elektrische Schaltung nicht als ein zusammenhängendes Ganzes, so zeigen sich in den einzelnen Schaltungsteilen häufig große Aktivitätsunterschiede. Es gibt Gruppen von Bauteilen, die zu einem Zeitpunkt aktiver sind als andere. Dies bedeutet anschaulich: würde man die Spannungsabfälle jedes Schaltungsbestandteiles abgreifen, so bekäme man an den aktiveren Teilen ein höherfrequentes Signal als an den weniger aktiven. Die Zuordnung zum jeweils latenten, d.h. weniger aktiven, oder aktiven Teil kann sich hierbei in der Zeit ändern oder konstant bleiben. So durchläuft in Schaltungen, die Eingangssignale

durchreichen, jeder Transistor und jede Kapazität die Aktivitätslevel latent-aktiv-latent. In Schaltungen, die digitale und analoge Techniken miteinander verbinden, sind jedoch die digitalen Schaltungsteile stets aktiver als die analogen.

Klassische Verfahren beachten diese Eigenschaft nicht. Sie behandeln die Netzwerkgleichungen durch eine adaptive¹ Schrittweitensteuerung als ein Ganzes, berechnen also die latenten Teile mit denselben (kleinen) Schrittweiten wie die aktiven. Sogenannte Multirate-Verfahren machen sich die unterschiedlichen Aktivitätslevel jedoch zunutze und berechnen latente Teile mit größeren und aktive mit kleineren Schrittweiten. Hierbei entsteht zwar durch die Kopplung – aktive beeinflussen die latenten Teile und umgekehrt – ein Mehraufwand, die Vorteile der Partitionierung überwiegen aber bei schwacher Kopplung und einem (gegenüber dem latenten) kleinen Teil von aktiven Elementen.

Ziel dieser Arbeit ist es, ein Multirate-Verfahren für differential-algebraische Netzwerkgleichungen zu entwickeln, dessen Schrittweitensteuerung auf einer Fehlerschätzung der algebraischen Variablen basiert. Hierzu werden Multirate-Verfahren für gewöhnliche Differentialgleichungen mit Integrationsmethoden für differential-algebraische Systeme vereint.

Die Arbeit ist in vier Kapitel eingeteilt:

Das erste Kapitel gibt einen Überblick über die mathematische Modellbildung elektrischer Schaltungen. Einen Schwerpunkt bildet die mathematische Struktur der Netzwerkgleichungen. In diesem Zusammenhang wird eine kurze Einführung in die analytischen Eigenschaften differential-algebraischer Gleichungen gegeben und das Multirate-Phänomen elektrischer Netzwerke erläutert.

Im zweiten Kapitel wird aus Multirate-Verfahren für gewöhnliche Differentialgleichungen ein Multirate-Algorithmus für differential-algebraische Gleichungen auf der Basis Rosenbrock-Wanner-ähnlicher Verfahren entwickelt.

Auf diese Verfahrensdefinition folgt im dritten, zentralen Kapitel dieser Arbeit die Verfahrenskonstruktion. Die Techniken der „graphischen“ Differentiation mittels Butcherbäumen werden mit einer neuen Klasse von Bäumen auf partitionierte differential-algebraische Systeme übertragen. Dies führt zu einem Automatismus, Bedingungsgleichungen für die Verfahrensparameter zu erhalten, so dass beliebige Konsistenzordnungen erzielt werden können. Der Konsistenzbetrachtung folgt die Betrachtung der Konvergenz und der Stabilität – soweit dies möglich ist. Die Implementierung und den Test eines Verfahrens der Ordnung 3 mit eingebettetem Verfahren der Ordnung 2 zur Schrittweitensteuerung ist Inhalt des vierten Kapitels

¹sich anpassende

Kapitel 1

Mathematische Modellbildung elektrischer Schaltungen

In diesem einführenden Kapitel wird zunächst kurz erläutert, in welche Klassen sich elektrische Schaltungen einteilen lassen. Danach wird auf den Netzwerkansatz zur Simulation zeitabhängiger Prozesse eingegangen sowie ein kurzer Überblick über die Grundbegriffe der Schaltungstheorie (Basiselemente, Kirchhoffsche Gesetze, Elementgleichungen) gegeben. Anschließend wird die mathematische Struktur der Netzwerkgleichungen und ihre Klassifizierung vorgestellt. Daraufhin werden die besonderen analytischen Eigenschaften der von modernen Software-Tools gelieferten Netzwerkgleichungen diskutiert und letztlich wird angesprochen, wie eine technische Besonderheit, das Multirate-Verhalten, elektrischer Schaltungen in der Simulation ausgenutzt werden kann.

1.1 Netzwerkansatz

Jedwede Simulation elektrischer Schaltungen erfordert die Überführung vorhandener Netzwerktopologien in ein mathematisch begreifbares Modell. Dabei werden elektrische Schaltungen zunächst in zwei Klassen unterteilt:

- **konzentrierte Schaltungen:**
Die räumliche Dimension dieser Schaltungen kann vernachlässigt werden, da sie im Vergleich zur Wellenlänge ihrer normalen Betriebsfrequenz unbedeutend klein ist.
- **verteilte Schaltungen:**
Um eine hinreichende Modellbildung zu gewährleisten, müssen die Ausdehnungen dieser Schaltung berücksichtigt werden.

Da verteilte Schaltungen als Grenzwert einer Folge konzentrierter Schaltungen modelliert werden können, werden im weiteren Verlauf nur konzentrierte Schal-

tungen betrachtet. Diese bestehen aus **konzentrierten Grundelementen** bzw. **Basiselementen**.

Mit Hilfe des **Netzwerkansatzes** kann aus einer physikalischen Schaltung automatisch deren zugehöriges mathematisches Modell generiert werden. Heute übernehmen Software-Werkzeuge, sog. TCAD¹-Tools, diese Aufgabe. Um eine solche Modellbildung sinnvoll und effizient durchführen zu können, müssen gewisse idealisierende Grundannahmen getroffen werden:

- **Allgemeine Annahmen:**

Kein Wärmeverlust, reale Bauelemente werden durch die entsprechenden idealen² Basiselemente ersetzt, die Leitungen werden als widerstandslos betrachtet.

- **Art der Basiselemente:**

Widerstand, Kapazität, Induktivität, Strom- und Spannungsquellen (Tabelle 1.1)

- **Topologie:**

Die Verbindungen zwischen den einzelnen Schaltungsteilen (den Basiselementen) beschreiben deren funktionellen Zusammenhang und nicht deren tatsächliche Lage im Raum.

- **Physikalische Gesetze:**

Die charakteristischen Gleichungen der Basiselemente folgen den Kirchhoffschen Gesetzen (Tabelle 1.1).

Auf die Modellierung von **Halbleiterbauelementen** wird kurz unter 1.2 eingegangen.

Bei Günther/Rentrop ([GR00]) ist die numerische Simulation elektrischer Schaltkreise auch in allgemeiner Form dargestellt.

1.2 Grundbegriffe aus der Schaltungstheorie

Die Modellbildung einer elektrischen Schaltung wird in der industriellen Anwendung von Softwarepaketen übernommen. Um deren Vorgehen verstehen zu können, werden in diesem Abschnitt einige Grundbegriffe und grundlegende Zusammenhänge aus der Schaltungstheorie erklärt.

Jene Punkte der Schaltung, an denen zwei oder mehr Leitungen zusammenreffen, werden als **Knoten** bezeichnet. Das Spannungspotential eines jeden dieser Knoten bezüglich eines ausgewählten Bezugsknoten (**Masse**) ist die **Knotenspannung**, das zwischen jeweils zwei Knoten die **Teil-** oder **Zweigspannung**.

¹engl.: technologically computer aided design

²insbesondere die Strom- und Spannungsquellen liefern Strom bzw. Spannung unabhängig von der Last

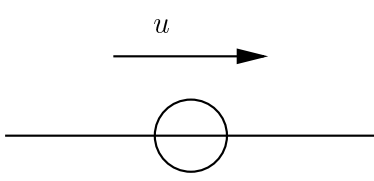
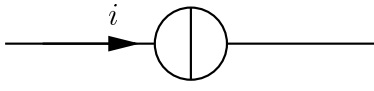
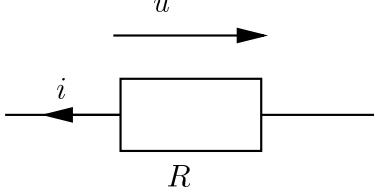
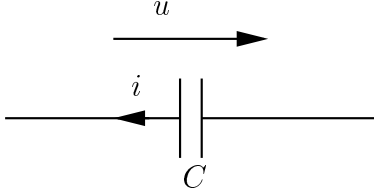
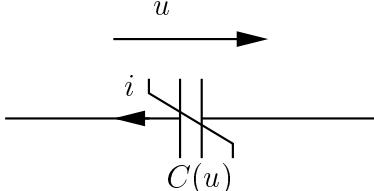
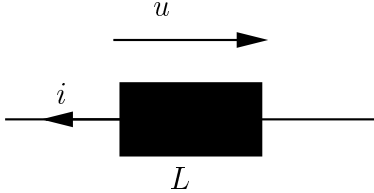
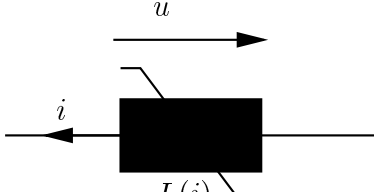
Basiselement	Netzwerksymbol	Gleichung
Spannungsquelle		$u = u_{in}(t)$
Stromquelle		$i = i_{in}(t)$
Linearer Widerstand		$i = G \cdot u, G = 1/R$
Lineare Kapazität		$i = C \cdot \dot{u}$
Nichtlineare Kapazität		$i = C(u) \cdot \dot{u}$
Lineare Induktivität		$u = L \cdot \frac{d}{dt}i$
Nichtlineare Induktivität		$u = L(i) \cdot \frac{d}{dt}i$

Tabelle 1.1: Symbole und charakteristische Gleichungen der Basiselemente

Ein **Zweig** der Schaltung beinhaltet alle, zwischen jeweils zwei Knoten gelegenen und vom selben Strom (**Zweigstrom**) durchflossenen Elemente und Leitungen. Schließen sich mehrere Zweige zu einem Kreis, so bilden sie eine **Masche**.

Die aus den Maxwellschen Gleichungen abgeleiteten sog. **charakteristischen Gleichungen** beschreiben das physikalische Verhalten der Basiselemente **Widerstand** (Ohmscher Widerstand), **Kapazität** (Kondensator), **Induktivität** (Spule), **Stromquelle** und **Spannungsquelle** unter idealisierten Annahmen, d. h. unter Vernachlässigung jeglicher Seiteneffekte. In Tabelle 1.1 findet sich eine Gegenüberstellung von jeweiligem Basiselement und zugehöriger charakteristischer Gleichung.

Die dort aufgestellten Elemente lassen sich drei Klassen zuteilen:

- **Unabhängige Quellen:**

Spannungs- bzw. Stromquellen, die lediglich zeitabhängige Zweigströme und -spannungen erzeugen. Diese beschreiben die Netzanregung.

- **Statische Elemente:**

Diese Elemente sind durch eine (beim Widerstand lineare) algebraische Beziehung zwischen Strom und Spannung der Form

$$f(U, I) = 0$$

gekennzeichnet.

- **Energiespeichernde (dynamische) Elemente:**

Diese Elemente (Induktivität, Kapazität) charakterisiert eine Beziehung zwischen Strom, Spannung und deren Ableitungen nach der Zeit.

Halbleiterbauelemente, wie die in integrierten Schaltungen häufig auftretenden MOS³-Transistoren und Dioden, werden – abhängig von der Modellierungstiefe – durch verschiedenartige Ersatzschaltungen mit Hilfe der Basiselemente abgebildet, um deren physikalisches Verhalten unterschiedlich genau zu beschreiben ([GR00]).

Zwischen den Größen **Knotenspannung** $U(t)$ und **Zweigstrom** $I(t)$ gelten zwei Arten von Beziehungen:

1. Die **Kirchhoffschen Gesetze**:

- **Kirchhoffsches Stromgesetz** bzw. **Knotenregel** (KCL⁴):

Die Summe der in einen Knoten einfließenden Ströme I_m ist zu jedem Zeitpunkt t Null:

³engl.: **metall-oxid-semiconductor**

⁴engl.: **Kirchhoff's current law**

$$\sum_m I_m(t) = 0$$

- **Kirchhoffsches Spannungsgesetz bzw. Maschenregel (KVL⁵):**
Die Summe der Spannungen U_m in einer Masche ist zu jedem Zeitpunkt t Null, umläuft man die Masche in einer vorgegebenen Richtung:

$$\sum_m U_m(t) = 0$$

Die Kirchhoffschen Gesetze stellen rein topologische Gesetzmäßigkeiten dar, denen das gesuchte Modell zu gehorchen hat.

2. Die grundlegenden Elementgleichungen:

Die an den Netzwerkelementen anliegenden Spannungen und fließenden Ströme werden durch ihre grundlegenden Elementgleichungen beschrieben. Diese können bei energiespeichernden Elementen auf zwei Arten erhalten werden:

- **kapazitäts- bzw. induktivitätsorientiert:**

Dies ist die **klassische** Vorgehensweise, bei der nur der Strom I sowie die Spannung U als Unbekannte gehandelt werden. Die an einem Element abfallende Spannung wird in Beziehung zum durchfließenden Strom I gesetzt (s. Tabelle 1.1).

- **ladungs- bzw. flussorientiert:**

Im Falle eines ladungsspeichernden Elements (Kapazität) wird eben diese Ladung als Funktion der abfallenden Spannung definiert. Durch zeitliche Ableitung erhält man hieraus den Zweigstrom. Analog wird für ein flussspeicherndes Element (Induktivität) der magnetische Fluss als Funktion des jeweiligen Stromes definiert. Die abfallende Spannung ergibt sich aus der Ableitung des Flusses nach der Zeit.

1.3 Modifizierte Knotenanalyse

Ein konzentriertes elektrisches Netzwerk lässt sich nun mithilfe der **Kirchhoffschen Gesetze**, der **Elementgleichungen** und **Ersatzschaltungen für Halbleiterbauelemente** beschreiben. Die von TCAD-Tools benutzte Methode der **modifizierten Netzwerkanalyse (MNA)**⁶ geht nun schrittweise wie folgt vor, um aus einer gegebenen Netzwerktopologie die **Netzwerkgleichungen** zu erhalten:

⁵engl.: Kirchhoff's voltage law

⁶engl.: modified nodal analysis

1. Mit Ausnahme des Masseknotens wird für jeden Knoten das Kirchhoffsche Stromgesetz (KCL) angewandt.
2. Die Zweigströme aller spannungsgesteuerten Elemente (Kapazitäten, Widerstände) werden durch ihre charakteristischen Gleichungen ersetzt (s. Tabelle 1.1).
3. Alle Zweigspannungen werden mit der Hilfe des Kirchhoffschen Spannungsgesetzes (KVL) in Knotenspannungen übertragen.

Diese Vorgehensweise führt auf die **klassische** – auf Induktivitäten und Kapazitäten gestützte – Formulierung des Netzwerkmodells mit Strömen und Spannungen als einzigen Unbekannten. Bei der ladungs- und flussorientierten Formulierung wird ausgenutzt, dass sich Ladung bzw. Fluss von ladungs- bzw. flussspeichernden Elementen (Kapazitäten, Induktivitäten) als Zeitintegrale aus der angelegten Spannung bzw. dem fließenden Strom ergeben, woraus Strom resp. Spannung durch Differentiation zu erhalten sind:

$$q(t) = \int_0^t C \dot{u} dt \Rightarrow i(t) = \dot{q}(t) \quad (1.1)$$

$$\phi(t) = \int_0^t L \frac{d}{dt} i dt \Rightarrow u(t) = \dot{\phi}(t) \quad (1.2)$$

In der modifizierten Netzwerkanalyse werden zusätzlich zu den oben genannten Schritten

1. Ströme aus ladungsspeichernden Elementen (Kapazitäten) durch $i(t) = \dot{q}(t)$ (s. 1.1) ersetzt,
2. die Elementgleichungen aller stromgesteuerten Elemente (Spannungsquelle, Induktivitäten) – hier werden im Falle von Induktivitäten Spannungen durch $u(t) = \dot{\phi}(t)$ ersetzt (s. 1.2) – dem System hinzugefügt,
3. Ladung q und Fluss ϕ als weitere Unbekannte aufgenommen.

Angesichts dieses scheinbar aufwendigeren Vorgehens mag sich die Frage stellen, warum Ladung und Fluss überhaupt eingeführt werden und nicht bei der klassischen Formulierung geblieben wird. Zum einen ist bei der ladungs-/ flussorientierten Methode dem Prinzip der Energieerhaltung (z. B. Ladungserhaltung) aus der Physik anschaulich Rechnung getragen. Zum anderen ist es aus Software-Sicht einfacher, ein größeres System aus mehr Unbekannten aufzubauen, als eine Optimierung auf möglichst wenige Gleichungen durchzuführen. Für weitere Argumente sei einmal mehr auf [GR00] und die dort erwähnte Literatur verwiesen.

Unabhängig davon, welche der beiden Vorgehensweisen nun verfolgt wird, erhält man Netzwerkgleichungen, deren Struktur von den Netzwerkelementen bestimmt wird:

- **rein algebraische Systeme:**

Enthält die Schaltung nur algebraische Elemente wie beispielsweise Widerstände, Dioden und Bipolartransistoren, so resultiert dies in einem System linearer oder nichtlinearer algebraischer Gleichungen. Die Lösung dieser Probleme soll jedoch nicht Inhalt dieser Arbeit sein.

- **Differential-Algebraische Systeme:**

Sind auch differentielle Elemente vorhanden, so tauchen in den Netzwerkgleichungen auch Ableitungen der Unbekannten auf.

Im Falle einer klassischen Modellbildung, bei der auf eine möglichst geringe Anzahl von Unbekannten geachtet wird, können die Netzwerkgleichungen, durch einige analytische Vorarbeiten auf die sogenannte **Zustandsform**⁷ gebracht werden, die einem System gewöhnlicher Differentialgleichungen entspricht:

$$\dot{x}(t) = f(t, x(t)), \quad x(t_0) = x_0.$$

Insbesondere beim ladungs-/flussorientierten Vorgehen wird jedoch zugunsten der Effektivität auf eine Minimierung der Anzahl der Unbekannten verzichtet und man erhält die Netzwerkgleichungen in der **Deskriptorform**. Dies ist in allgemeinsten Form ein implizites Differentialgleichungssystem:

$$\mathcal{F}(\dot{x}, x, t) = 0. \tag{1.3}$$

Meist gilt hierbei

$$\det \left(\frac{\partial \mathcal{F}}{\partial \dot{x}} \right) = 0,$$

was aus (1.3) ein differential-algebraisches System macht.

Da industrielle TCAD-Tools mit dem redundanten MNA-Verfahren auf differential-algebraische Netzwerkgleichungen in der Deskriptorform (1.3) führen, werden im Weiteren nur solche Systeme betrachtet.

1.4 Differential-Algebraische Netzwerkgleichungen

Die in der Simulation elektrischer Schaltkreise auftretenden differential-algebraischen Netzwerkgleichungen sind eine Sonderform von allgemeinen differential-algebraischen Gleichungen. Solche Systeme bringen gewisse analytische und numerische Besonderheiten und Probleme mit sich. Diese Besonderheiten werden am Sonderfall der differential-algebraischen Netzwerkgleichungen erläutert. Die

⁷in englischsprachigen Veröffentlichungen als **state-space form** bezeichnet

grundlegenden Aussagen hieraus können dann auch auf den allgemeinen Fall differential-algebraischer Systeme übertragen werden.

In allgemeinsten Form sind **differential-algebraischen Gleichungen (DAEs)**⁸ implizite Differentialgleichungssysteme:

$$\mathcal{F}(t, x, \dot{x}) = 0 \quad \text{wobei} \quad \det \left(\frac{\partial \mathcal{F}}{\partial \dot{x}} \right) = 0 \quad \text{zulässig ist} \quad (1.4)$$

Differential-algebraische Netzwerkgleichungen stellen sich als Sonderform der allgemeinen DAE (1.4) dar und können einem der drei folgenden Arten zugeordnet werden.

1. Linear-implizite Systeme:

Enthält die Schaltung nur

- lineare Kapazitäten und Induktivitäten als differentielle Elemente,
- lineare Widerstände als algebraische Elemente,
- unabhängige Strom- und Spannungsquellen als rein zeitabhängige Komponenten,

so ergeben sich die Netzwerkgleichungen als Anfangswertproblem der Form

$$\begin{aligned} B \cdot \dot{x} + D \cdot x + s(t) &= 0, \\ x(t_0) &= x_0. \end{aligned} \quad (1.5)$$

Die Matrix B enthält im wesentlichen die Kapazitäten und Induktivitäten, die Matrix D entsprechend die Leitwerte und topologische Informationen. Die rein zeitabhängige Funktion $s(t)$ beschreibt die Netzanregung.

2. Linear-implizite nichtlineare Systeme:

Enthält die Schaltung zusätzlich nichtlineare algebraische Elemente wie Dioden oder Bipolartransistoren, so stellen sich die Netzwerkgleichungen als Anfangswertproblem folgender Form dar:

$$\begin{aligned} B \cdot \dot{x} + f(x) + s(t) &= 0, \\ x(t_0) &= x_0, \end{aligned} \quad (1.6)$$

mit einer nichtlinearen Funktion $f(x)$.

3. Quasilinear-implizite Systeme:

Treten zu alledem noch nichtlineare differentielle Elemente wie nichtlineare Induktivitäten und nichtlineare Kapazitäten in Erscheinung, so schlägt sich dies auf das zu betrachtende Anfangswertproblem in folgender Form nieder:

⁸engl.: **D**ifferential **A**lgebraic **E**quations

$$\begin{aligned} B(x) \cdot \dot{x} + f(x) + s(t) &= 0 \\ x(t_0) &= x_0 \end{aligned} \tag{1.7}$$

Hierbei hängt die Matrix $B(x)$ von Strom/Spannung bzw. Ladung/Fluss ab.

Alle diese DAE-Typen können in **autonomer** und **nichtautonomer** Form vorkommen. Im autonomen Fall ist die Netzanregung $s(t)$ nicht von der Zeit abhängig, d. h. die Schaltung wird nur von konstanten Strom-/Spannungsquellen beeinflusst. Ist dagegen $s(t)$ ein nicht-konstanter, von der Zeit abhängiger Vektor, so hat man es mit einem nichtautonomen System zu tun.

Anhand linear-impliziter Systeme lassen sich nun einige Grundbegriffe (konsistente Anfangswerte, Indexkonzept) aus der Theorie der differential-algebraischen Gleichungen erklären. Anfangswerte können für DAE-Probleme nicht frei, sondern müssen **konsistent**, d. h. passend, gewählt werden. Der **Index** einer differential-algebraischen Gleichung ist eine Art Maß für die „Entfernung“ einer DAE zu einer gewöhnlichen Differentialgleichung. Je größer diese „Entfernung“ ist, umso größer sind die auftauchenden Probleme ([GHR00]). Die zugrundeliegenden Konzepte können auch auf linear-implizite nichtlineare und quasilinear-implizite Systeme (hierzu siehe [HW96, insbes. VI.6, VII.1]) bzw. auf die allgemeine DAE-Darstellung (1.4) ([SFR91, RSW96]) übertragen werden.

Konsistente Anfangswerte

Bereits am linear-impliziten Fall

$$\begin{aligned} B \cdot \dot{x} + D \cdot x + s(t) &= 0, \\ x(t_0) &= x_0, \end{aligned}$$

ist leicht zu sehen, dass der Anfangswert (x_0, t_0) nicht wie bei einer gewöhnlichen Differentialgleichung (ODE⁹) frei gewählt werden kann. Der Anfangswert muss eine **Konsistenzbedingung** erfüllen. Als Mindestanforderung stellt sich diese Konsistenzbedingung dar als:

$$D \cdot x_0 + s(t_0) \in \text{Im}(B).$$

Im späteren Index-1 Fall ist diese Bedingung notwendig und hinreichend. Im Falle eines höheren Indexes ist eine hinreichende Konsistenzbedingung jedoch versteckt und ergibt sich erst nach einer Reihe von Differentiationen.

Einen konsistenten Anfangswert zu finden kann u. U. sehr schwer sein. Außerdem denke man beispielsweise daran, dass die Auswirkungen fehlerbehafteter

⁹engl.: **ordinary differential equation**

Anfangswerte möglicherweise nicht mehr kontrollierbar sind, da diese eventuell die Konsistenzbedingung nicht erfüllen, und damit noch nicht einmal eine Lösung existieren muss¹⁰.

Das Index-Konzept für DAEs

Die analytischen Eigenschaften einer DAE (und damit letztlich ihr Verhalten bzgl. der numerischen Integration) können durch ein **Index-Konzept** studiert werden, das diese DAE einer Klasse, im Verhalten ähnlicher differential-algebraischer Gleichungen zuordnet.

Im linear-impliziten Fall

$$B \cdot \dot{x} + D \cdot x + s(t) = 0 \quad (1.8)$$

hängt dieser **Index** nur von der Struktur der Matrizen B und D ab.

- ODE-Fall:

Ist B regulär, so lässt sich (1.8) in eine gewöhnliche Differentialgleichung transformieren:

$$\dot{x} = -B^{-1}(D \cdot x + s(t))$$

Darauf lassen sich nun alle aus der Theorie der ODEs bekannten Sätze anwenden (Existenz, Eindeutigkeit, ...). Da bei der elektrischen Netzwerkanalyse normalerweise die Matrizen B und D nur schwach besetzt sind, wird man sich bei der numerischen Berechnung aus Effizienzgründen vor einer Invertierung jedoch hüten.

Die zu untersuchende DAE ist also eine ODE, die ohne zusätzliche Differentiationen erhalten werden kann. Deshalb sagt man, die DAE besitzt den **Index 0**.

- DAE-Fall:

Nun ist B singulär und D sei der Einfachheit halber als regulär angenommen (der Fall B und D singulär ist z. B. unter [HW96, VII.1] dargestellt). Dann führt eine Multiplikation von (1.8) mit D^{-1} auf:

$$D^{-1}B \cdot \dot{x} + x = -D^{-1} \cdot s(t) \quad (1.9)$$

Die *Jordan-Zerlegung* des Matrixproduktes $D^{-1}B$ liefert reguläre Matrizen $\tilde{B}$ und T und eine nilpotente Matrix N , so dass gilt:

$$D^{-1}B = T^{-1} \begin{pmatrix} \tilde{B} & 0 \\ 0 & N \end{pmatrix} T. \quad (1.10)$$

¹⁰Bei ODEs hängen dagegen die Lösungen stetig differenzierbar von den Anfangswerten ab

Dies in (1.9) eingesetzt und anschließend von links mit T multipliziert führt auf:

$$\begin{pmatrix} \tilde{B} & 0 \\ 0 & N \end{pmatrix} T\dot{x} + Tx = -TD^{-1}s(t). \quad (1.11)$$

Aufgrund der Struktur von (1.11) liegt es nahe, folgende Transformation anzuwenden:

$$\begin{pmatrix} y \\ z \end{pmatrix} := Tx \quad \text{und} \quad \begin{pmatrix} \eta(t) \\ \delta(t) \end{pmatrix} := -TD^{-1}s(t),$$

wobei die Dimensionen von y und z resp. $\eta(t)$ und $\delta(t)$ mit den Dimensionen der Matrizen $\tilde{B}$ und N korrespondieren.

Somit ist die DAE (1.8) in eine ODE und einen nilpotenten Teil zerlegt:

$$\dot{y} = \tilde{B}^{-1}(\eta(t) - y) \quad (1.12a)$$

$$N\dot{z} = \delta(t) - z \quad (1.12b)$$

Der Nilpotenz-Grad ν der Matrix N , also die kleinste natürliche Zahl für die gilt: $N^\nu = 0$, $N^{\nu-1} \neq 0$, liefert nun den **algebraischen Index** der linear-impliziten DAE (1.8):

– *Index 1:*

Im einfachsten Fall ist $\nu = 1$, d. h. $N = 0$. Dann liest sich (1.12b) als algebraische Gleichung für z ,

$$z = \delta(t), \quad (1.13)$$

woraus sich nach einmaliger Differentiation nach der Zeit eine gewöhnliche Differentialgleichung für z gewinnen lässt.

– *höherer Index:*

Ist jedoch $\nu > 1$, so gestaltet es sich schwieriger, eine algebraische Bedingungsgleichung bzw. eine ODE für z zu erhalten. Man differenziere (1.12b) sukzessive nach der Zeit und multipliziere auf das jeweilige Ergebnis von links N dazu. Da $N^\nu = 0$ ist, bricht dieser Prozess nach ν Schritten ab:

$$\begin{aligned} N\dot{z} &= \delta(t) - z \\ N^2\ddot{z} &= N\dot{\delta}(t) - N\dot{z} \\ N^3z^{(3)} &= N^2\ddot{\delta}(t) - N^2\ddot{z} \\ &\vdots \\ N^{\nu-1}z^{(\nu-1)} &= N^{\nu-2}\delta^{(\nu-2)}(t) - N^{\nu-2}z^{(\nu-2)} \\ 0 &= N^{\nu-1}\delta^{(\nu-1)}(t) - N^{\nu-1}z^{(\nu-1)} \end{aligned}$$

Durch einen Prozess des rekursiven Auflörens nach Termen der Form $N^\mu z^{(\mu)}$ und Einsetzens, beginnend mit der letzten Gleichung, erreicht man eine explizite Darstellung für z :

$$z = \delta(t) - N\dot{\delta}(t) + N^2\ddot{\delta}(t) - \dots + (-1)^{\nu-1}N^{\nu-1}\delta^{(\nu-1)}(t). \quad (1.14)$$

Differenziert man (1.14) einmal mehr nach der Zeit t ergibt sich auch für z eine gewöhnliche Differentialgleichung und die DAE (1.12a, 1.12b) ist somit nach ν -facher Ableitung überführt in ein System von ODEs.

$$\begin{aligned} \dot{y} &= \tilde{B}^{-1}(\eta(t) - y) \\ \dot{z} &= \dot{\delta}(t) - \dots + (-1)^{\nu-1}N^{\nu-1}\delta^{(\nu)}(t) \end{aligned}$$

Die Lösung von (1.8) muss eine algebraische Nebenbedingung erfüllen, die im Index-1 Fall in (1.13) noch leicht zu erkennen ist, sich aber erst nach einer Reihe von Differentiationen explizit zeigt, sollte der Nilpotenz-Grad von N größer als 1 sein (zu sehen in (1.14)). Diese algebraische Nebenbedingung wird anschaulich greifbar, und in der späteren numerischen Verfahrenskonstruktion bedeutsam, stellt man sie sich als eine Mannigfaltigkeit vor, auf der die Lösung des ODE-Teils zu verlaufen hat.

Besonders deutlich zeigt sich, wie bereits erwähnt, die algebraische Bedingung in der Wahl des Anfangswertes (x_0, t_0) . Daher sind DAE-Lösungen nicht mehr wie bei gewöhnlichen Differentialgleichungen stetig differenzierbar von den Anfangswerten abhängig.

Darüber hinaus kann nicht einmal eine stetige Lösung $x(t)$ der DAE (1.8) garantiert werden. Denn dazu muss im Index-1 Fall (1.13) die Inhomogenität $\delta(t)$ stetig sein und im Falle eines höheren Indexes nach (1.14) sogar die $(\nu - 1)$ -te Ableitung von $\delta(t)$.

Aus den Betrachtungen, die im Falle einer linear-impliziten DAE zum **algebraischen Index** geführt haben, lassen sich Index-Konzepte für den allgemeinen DAE-Fall gewinnen:

- **Differentieller Index:**

Der differentielle Index einer DAE ist die minimale Anzahl an Differentiationen, die nötig sind, um die DAE in eine ODE zu überführen.

Da die numerische Differentiation ein nicht-stabiler Prozess ist, ist zu erwarten, dass die Probleme, die bei der numerischen Betrachtung einer DAE auftauchen, mit dem differentiellen Index wachsen.

- **Störungs-Index:**

Der Störungs-Index einer DAE ist die maximale, sich auf die Ableitung der Lösung auswirkende Ableitungsordnung einer auftretenden Störung.

([GHR00])

Bzw.

Der Störungsindex einer DAE ist eins plus die maximale Ableitungsordnung einer Störung, die sich auf die Lösung auswirkt. ([HW96, VII.1, Def 1.3])

Fasst man in vorangegangener Betrachtung $\delta(t)$ etwa als hochfrequentes störendes Rauschen auf, dann wird der Einfluss dieses Rauschens umso größer, je höhere Ableitungen von $\delta(t)$ in (1.14) eingehen. Anschaulich wird dieser Störungsindex durch ein kleines Zahlenbeispiel:

man denke sich z. B. $\delta(t) = \sin(10t)$ und einen Störungsindex von 5, dann hat man $\delta^{(4)}(t) = 10^4 \cdot \sin(10t) = 10^4 \cdot \delta(t)$, d. h. der Einfluss der Störung auf die Lösung wird um den Faktor 10^4 verstärkt.

Nach der Beschreibung der differential-algebraischen, somit mathematischen, Eigenschaften der Netzwerkgleichungen, wollen wir uns nun der Multirate-Eigenschaft elektrischer Schaltungen zuwenden. Begründet sich erstere Eigenschaft v. a. auf dem redundanten, jedoch effizienten, MNA-Verfahren industrieller Simulationstools, ist die Multirate-Eigenschaft ein physikalisch-technisches Phänomen, das unabhängig von der mathematischen Modellierung auftritt.

1.5 Multirate-Eigenschaft elektrischer Schaltungen

Viele elektrische Schaltungen besitzen unabhängig von ihrer mathematischen Modellierung eine Besonderheit, die auch in der numerischen Transienten-Analyse – der Untersuchung ihres zeitlichen Verhaltens – ausgenutzt werden kann.

Beobachtet man die zeitlichen Veränderungen der Knotenspannungen, so stellt man fest, dass sich diese an einigen Knoten schneller ändern als an anderen. Mit anderen Worten, würde man die Knotenspannungen abgreifen, so erhielte man höher- und und niederfrequenter Signale. Dies zeigt deutlich, dass nicht alle Teile einer Schaltung immer gleich aktiv sind, sondern **Aktivitätsunterschiede** zu Tage treten. In einem ersten Schritt könnte man so eine Schaltung in Gebiete aufteilen, die **aktiv** sind, und solche, die weniger aktiv bzw. **latent** sind. Dies könnte man weiter verfeinern in mehrere Aktivitätsstufen, muss dabei aber stets bedenken, dass alle wie auch immer entstandenen Teile zu einem Ganzen gehören, sich also gegenseitig beeinflussen, miteinander **gekoppelt** sind. Aktive Schaltungsteile bestimmen das Verhalten der weniger aktiven mit, und die weniger aktiven steuern die aktiveren.

Die Grundidee, die Multirate-Eigenschaft in der numerischen Berechnung auszunutzen, besteht darin, für verschieden aktive Teile mit verschieden großen Schrittweiten zu rechnen. Wie leicht vorzustellen ist, kostet es zusätzliche Arbeit, die physikalische Kopplung der verschieden aktiven Schaltungsbereiche in einem numerisches Lösungsverfahren zu berücksichtigen. Der Grund hierfür liegt

darin, dass Daten aus verschiedenen feinen Zeitdiskretisierungen ineinander einfließen müssen. Beispielsweise müssen Daten, die weniger aktiven Teilen zugeordnet sind, somit nur an wenigen Zeitpunkten vorliegen, der Berechnung von aktiveren Teilen auf einem entsprechend feineren Zeitgitter angemessen zugänglich gemacht werden. Ist der Anzahl der weniger aktiven Teile größer als die Anzahl der aktiven, und ist die Kopplung „schwach“, so wird der zusätzliche Aufwand durch die Einsparung an Schritten für die wenig aktiven Teile ausgeglichen und übertroffen¹¹.

Um die Anzahl der Kopplungsgrößen klein zu halten, beschränken wir uns in dieser Arbeit auf zwei Partitionen, einen **aktiven** und einen **latenten** Teil. Somit haben wir nur zwei Kopplungen vorliegen – Einkopplung des latenten in den aktiven Teil und umgekehrt. Wie leicht nachzurechnen ist, wären es bei drei Aktivitätslevel bereits 6, bei vier 12, usw. .

Die relativen Aktivitätsunterschiede können nun zeitlich konstant bleiben (z. B. in radio frequency-Anwendungen¹², wo analoge – somit niedergrequente und latente – und digitale – somit hochfrequente und aktive – Netzwerke gekoppelt werden). Sie können sich aber auch laufend ändern (man denke etwa an eine Inverterkette: hier wird ein Signal durch eine Reihe von Transistoren gereicht, die nacheinander vom latenten in den aktiven und zurück in den latenten Zustand gehen).

In der numerischen Betrachtung macht man sich nun diese Partitionierung folgendermaßen zu Nutze: Man überträgt die Einteilung im technischen Sinne auf eine Einteilung im mathematischen Sinne, indem man den Vektor der Unbekannten aufsplittet in latente und aktive Komponenten:

$$x = \begin{pmatrix} x_l \\ x_a \end{pmatrix}.$$

In gleicher Weise wird auch die zugrundeliegende Differentialgleichung – hier als implizite Differentialgleichung geschrieben – partitioniert:

$$\mathcal{F}(t, x, \dot{x}) = \begin{pmatrix} \mathcal{F}_l(t, x_l, x_a, \dot{x}_l, \dot{x}_a) \\ \mathcal{F}_a(t, x_l, x_a, \dot{x}_l, \dot{x}_a) \end{pmatrix}$$

Anschließend wendet man numerische Verfahren an, die im latenten Teil $\mathcal{F}_l$ mit einer großen Schrittweite h_l , im aktiven Teil $\mathcal{F}_a$ mit einer kleinen Schrittweite h_a rechnen und die Kopplung angemessen berücksichtigen. Wie dies genau vonstatten gehen kann, ist das Anliegen dieser Arbeit und wird im Weiteren ausgeführt.

¹¹dies macht eine Multirate-Strategie erst sinnvoll

¹²Hierzu zählen z. B. der Rundfunk- und Fernsehbereich, der Mobilfunk, . . .

Kapitel 2

Multirate-Verfahren – Effiziente numerische Berechnung der Netzwerkgleichungen

Im vorangegangenen Kapitel wurden die spezielle mathematische Struktur von Netzwerkgleichungen, der DAE-Form und die physikalisch-technische Multirate-Eigenschaft und deren Übertragung auf das mathematische Modell mittels Partitionierung der Netzwerkgleichungen vorgestellt.

In diesem Kapitel beschäftigen wir uns mit der **numerischen Transienten-Analyse**, der numerischen Berechnung des zeitlichen Verhaltens der zur Netzwerkgleichung gehörenden Schaltung.

Nach einem Überblick über **Multirate-Verfahren für Netzwerkgleichungen in der Zustandsform**, für gewöhnliche Differentialgleichungssysteme also, wird ein **Multirate-Verfahren für Index-1 Netzwerkgleichungen in Deskriptorform** entwickelt.

2.1 Multirate-Verfahren für Netzwerkgleichungen in der Zustandsform

Zunächst werden Entwicklungen auf dem Gebiet der Multirate-Verfahren für ODE-Systeme dargestellt. Auf eine formale Ausführung der jeweiligen Verfahren wird größtenteils verzichtet, zumal diese in den genannten Arbeiten nachzulesen sind und letztlich im folgenden Abschnitt eingehen.

Um in der elektrischen Simulation ODE-Systeme zu erhalten, beschränkt man sich auf diejenigen Schaltungen, deren Netzwerkgleichungen sich in der **Zustandsform**

$$\begin{aligned}
\dot{x}(t) &= f(x) \\
x(t_0) &= x_0 \\
x \in \mathbb{R}^n \quad , \quad f : \mathbb{R}^n &\rightarrow \mathbb{R}^n \\
t &\in [t_0, t_{end}]
\end{aligned} \tag{2.1}$$

schreiben lassen.

Es mag auffallen, dass obige ODE als autonomes – nicht von der Zeit t abhängiges – System geschrieben ist. Dies geschieht nur zur Vereinfachung, zumal jedes nichtautonome System in ein autonomes überführt werden kann, indem man t zum Vektor der Unbekannten hinzufügt und die zusätzliche triviale Differentialgleichung $\dot{t} = 1$ einführt.

Herkömmliche (adaptive) ODE-Löser, wie z. B. das unter MATLAB implementierte ROW-Verfahren `ode23s`, betrachten Anfangswertprobleme der Form (2.1) als Ganzes und führen die numerische Integration in jedem Schritt mit einer universellen Schrittweite h durch. Diese ist nach der adaptiven Schrittweitensteuerung bestimmt durch die jeweils aktivsten Schaltungsteile resp. Unbekannten des Systems.

In hochintegrierten elektrischen Schaltungen, Schaltungen, die aus mehreren tausend Bauteilen auf kleinstem Raum bestehen und die **Multirate-Eigenschaft** besitzen, führt dies zu einem unnötigen Mehraufwand. Während nämlich die aktiven Teile nach einer kurzen Schrittweite verlangen, um die jeweiligen Genauigkeitsforderungen erfüllen zu können, wäre es für weniger aktive Unbekannte ausreichend, sie in größeren Abständen abzutasten.

Geht man nun von einer Zweiteilung des Systems in einen **aktiven** und einen **latenten** Teil aus¹, so kann man auf den aktiven Teil ein Verfahren anwenden, das mit einer kleinen Schrittweite h_a arbeitet, und entsprechend auf den latenten Teil eines mit einer größeren Schrittweite h_l .

Formal betrachtet wird das AWP² (2.1) überführt in ein System gekoppelter ODEs:

¹Eine Zweiteilung verspricht eine Reduktion des von der Schrittweite verursachten Aufwandes und einen moderaten Anstieg der Arbeit, die durch die Kopplung entsteht.

²Anfangswertproblem

$$\begin{aligned}
x &= \begin{pmatrix} x_l \\ x_a \end{pmatrix} \quad \text{mit} \quad \begin{cases} \dot{x}_l = f_l(x_l, \boxed{x_a}) \\ \dot{x}_a = f_a(\boxed{x_l}, x_a) \end{cases} \\
x_0 &= \begin{pmatrix} x_{l,0} \\ x_{a,0} \end{pmatrix} \\
x_l &\in \mathbb{R}^{n_l} \quad , \quad f_l : \mathbb{R}^{n_l} \times \mathbb{R}^{n_a} \rightarrow \mathbb{R}^{n_l} \\
x_a &\in \mathbb{R}^{n_a} \quad , \quad f_a : \mathbb{R}^{n_l} \times \mathbb{R}^{n_a} \rightarrow \mathbb{R}^{n_a} \\
n_a + n_l &= n.
\end{aligned} \tag{2.2}$$

Die eingerahmten Funktionsparameter kennzeichnen hier die jeweils zu berücksichtigende Kopplung.

Eine Partitionierung macht nur dann wirklich Sinn, wenn der weitaus größere Teil latent ist, d. h. wenn $n_a \ll n_l$ gilt. Denn ist dem nicht so, dann bleibt zu befürchten, dass die Rechenarbeit, die durch die Kopplung entsteht, die Einsparung übertrifft, die ein Vorgehen mit einer größeren Schrittweite bringt. Aus dem gleichen Grund wird man natürlicherweise verlangen, dass die Kopplung zwischen den Aktivitätsbereichen von schwacher Natur ist, d. h. die aktiven Unbekannten haben auf die latenten nur einen „kleinen“ Einfluss und umgekehrt. Aus formaler Sicht bietet sich an, dies via Matrixnormen³ und partielle Ableitungen – die als existent vorausgesetzt werden – auszudrücken:

$$\begin{aligned}
\left\| \frac{\partial f_l}{\partial x_a}(t) \right\| &\ll \left\| \frac{\partial f_l}{\partial x_l}(t) \right\| \\
\left\| \frac{\partial f_a}{\partial x_l}(t) \right\| &\ll \left\| \frac{\partial f_a}{\partial x_a}(t) \right\| \\
\forall t &\in [t_0, t_{end}].
\end{aligned}$$

Die grundlegende Idee, Multirate-Eigenschaften numerisch zu nutzen, besteht nun darin, von einem gegebenen Startpunkt t_0 mit Anfangswerten $x_{l,0}, x_{a,0}$ auszugehen, in einem **Makroschritt** eine Näherung $x_{l,1} \approx x_l(t_0 + h_l)$ für den latenten Teil zu bestimmen und in dem somit entstandenen Intervall $[t_0, t_0 + h_l]$ durch **m Mikroschritte** Näherungen $x_{a,1}, x_{a,2}, \dots, x_{a,m-1}, x_{a,m} \approx x_a(t_0 + h_l)$ zu errechnen. Abbildung 2.1 zeigt anschaulich eine solche Unterteilung eines Makroschrittes in mehrere Mikroschritte, wobei hier die Mikroschritte äquidistant gewählt wurden und zusätzlich so, dass $m = \frac{h_l}{h_a} \in \mathbb{N}$ gilt. Die Aktivitätsdifferenzen in der Schaltung werden durch ein Multirate-Verfahren also auf die **Diskretisierungsebene** „abgebildet“.

Hier stellt sich als nächstes die Frage, welche Art von numerischen Verfahren geeignet ist.

³Die Wahl der Matrixnorm ist irrelevant, da sich Normen gegeneinander abschätzen lassen.

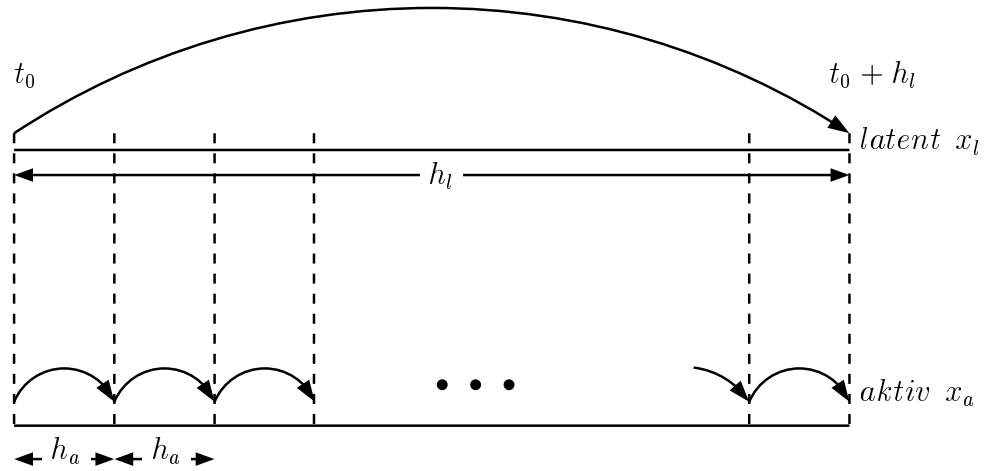


Abbildung 2.1: Macro- und feste Micro-Schritte

Die sich nur langsam verändernde Lösung des latenten Teils lässt auf ein **steifes** Problem schließen, was **implizite** Verfahren nötig macht. Die Aussagen der **2. Dalquistischen Barriere** ([HW96]) – es gibt kein A-stabiles Mehrschrittverfahren mit Konsistenzordnung > 2 – lassen Mehrschrittverfahren als kaum geeignet erscheinen.

Dies alles favorisiert **implizite Runge-Kutta-Verfahren (RK)** oder, um die dortigen nichtlinearen Gleichungssysteme zu umgehen, **Rosenbrock-Wanner-Verfahren (ROW)** (zu ROW-Verfahren siehe [HW96]). Zum einen verleiht die Einkopplung des steifen latenten Teils in den aktiven auch diesem einen steifen Charakter. Zum anderen kann durch eine eventuell unsaubere Partitionierung nicht ausgeschlossen werden, dass x_a selbst steife Komponenten besitzt. Deshalb kommen auch im aktiven Teil die entsprechenden RK- bzw. ROW-Verfahren zum Einsatz.

In allgemeinsten Form lassen sich solche, in den Mikroschritten äquidistante, Verfahren für einen Makroschritt folgendermaßen formulieren:

Schrittweiten:

$$h_a = \frac{1}{m} \cdot h_l, \quad m \in \mathbb{N}$$

latente Komponenten:

$$\begin{aligned} x_{l,1} &= x_{l,0} + \sum_{i=1}^s b_{L,i} \cdot k_{L,i} \\ &\approx x_l(t_0 + h_l) \end{aligned}$$

aktive Komponenten:

$$\begin{aligned} x_{a,\lambda} &= x_{a,\lambda-1} + \sum_{i=1}^s b_{A,i} \cdot k_{A,i}^\lambda \\ &\approx x_a(t_0 + \lambda \cdot h_a) \quad (\lambda = 1, \dots, m) \end{aligned} \tag{2.3}$$

Zuwächse latent:

$$k_{L,i} = \phi_L(x_{l,0}, \boxed{\tilde{X}_{A,i}}, k_{L,1}, \dots, k_{L,s}) \quad i = 1, \dots, s$$

Zuwächse aktiv:

$$k_{A,i}^\lambda = \phi_A(x_{a,\lambda-1}, \boxed{\tilde{X}_{L,i}^\lambda}, k_{A,1}, \dots, k_{A,s}) \quad i = 1, \dots, s$$

Hierbei bezeichnen ϕ_L und ϕ_A die entsprechenden Verfahrensvorschriften, für die aufgrund der Implizitheit der Verfahren in der Berechnung des i -ten Zuwachses $\partial\phi_L/\partial k_{L,i} \neq 0$ bzw. $\partial\phi_A/\partial k_{A,i}^\lambda \neq 0$ gilt. Außerdem sind hier die Stufenzahlen s der Verfahren für latenten und aktiven Teil gleich gewählt worden.

Die eingerahmten Parameter sind die eingekoppelten Größen aus der jeweils anderen Diskretisierungsebene:

$$\begin{aligned} \tilde{X}_{A,i} &\approx x_a(t_0 + \alpha_i^L h_l) \\ \tilde{X}_{L,i}^\lambda &\approx x_l(t_0 + \lambda \cdot h_a + \alpha_i^A h_a), \quad \lambda = 1, \dots, m \end{aligned} \tag{2.4}$$

Die Auswertungsstellen $t_0 + \alpha_i^L h_l$ bzw. $t_0 + \lambda \cdot h_a + \alpha_i^A h_a$ ergeben sich aus der Betrachtung von latentem und aktiven Teil der partitionierten ODE (2.2) als jeweils nichtautonome Differentialgleichungen und der Anwendung eines ROW-Verfahrens hierauf (siehe auch z. B. [HW96, IV.7], [BG01])

Nun gibt es verschiedene Ansätze, die **Kopplungsparameter** zu bestimmen, die kurz vorgestellt werden sollen:

- **Extrapolation/Interpolation ([GR93]):**

Bei dieser Vorgehensweise wird in jedem Makroschritt von t_0 bis $t_0 + h_l$ zunächst der latente Teil berechnet. Die benötigten Werte für den aktiven

Teil im Intervall $[t_0, t_0 + h_l]$ werden durch eine Extrapolation der bereits aus vorigen Schritten stammenden Werte von x_a bestimmt. Anschließend werden die m Mikroschritte für den aktiven Part durchgeführt, wobei sich die Werte $\tilde{X}_{L,i}^\lambda$ durch Interpolation von x_l auf dem Intervall $[t_0, t_0 + h_l]$ ergeben. Diese Reihenfolge, der „slowest first“, also zuerst den langsamen Teil, dann den schnellen Teil zu berechnen, spiegelt die weitgehende Entkopplung wieder. Günther und Rentrop ([GR93]) untersuchten die Konsistenz- und Konvergenzeigenschaften dieser Vorgehensweise und entwickelten mit MR0W(2)3 ein Verfahren der Ordnung (2)3, dessen Zuwachsberechnung (wie der Name vermuten lässt) auf einem ROW-Ansatz basiert⁴.

Ein Extrapolations-/Interpolationsschema macht jedoch aus einem Einschrittverfahren in gewisser Weise ein Zweischrittverfahren, da neben der eigentlichen Berechnung der Zuwächse immer noch ein Extrapolations- bzw. Interpolationsverfahren mitgezogen werden muss.

- **Generalised-Multirate ([KR99]):**

Ein weiterer Ansatz, der die Extrapolation/Interpolation vermeidet und aus dem „Zweischritt“-Verfahren ein echtes Einschrittverfahren macht, stammt von Kværnø und Rentrop ([KR99]): *generalised-multirate*.

Die Werte $\tilde{X}_{A,i}$ und $\tilde{X}_{L,i}^\lambda$ werden aus gewichteten Zuwächsen $k_{L,i}^\lambda, k_{A,i}$ in Runge-Kutta-Manier bestimmt. Die Zuwächse $k_{L,i}^\lambda$ und $k_{A,i}$, mit denen die Kopplung berechnet wird, sind dieselben, die in der jeweils zugehörigen Aktivitätsschicht benutzt werden. Sie müssen also nicht neu für die Kopplung berechnet werden, sondern können, lediglich anders gewichtet, aus vorigen Schritten übernommen werden. Auf eine genaue Definition der Zuwächse wird an dieser Stelle verzichtet. Die Idee wird im neu zu definierenden Verfahren übernommen. Für eine genaue Verfahrensdefinition des *generalised multirate* sei auf die Arbeit von Kværnø und Rentrop ([KR99]) verwiesen.

Das von Kværnø und Rentrop auf Runge-Kutta-Basis entwickelte Verfahren MRKI arbeitet innerhalb eines Makroschrittes mit äquidistanten Mikroschritten. Dies bedeutet, dass man die Mikroschrittweiten während eines Makroschrittes nicht adaptiv anpassen kann. Zudem führt die Ablehnung eines Mikroschrittes zur Ablehnung des gesamten Makroschrittes. Dieser Makroschritt und alle darin enthaltenen Mikroschritte müssen dann mit neuen Schrittweiten nochmals berechnet werden.

- **Mixed Multirate ([BGK]):**

Der Arbeit von Bartel, Günther und Kværnø ([BGK]) entstammt nun die Idee des *mixed-multirate*. Dieser Ansatz baut auf dem *generalised-multirate*

⁴das Verfahren besitzt Konsistenzordnung 3 mit einem eingebetteten Verfahren der Ordnung 2 zur Schrittweiten- und Fehlerkontrolle

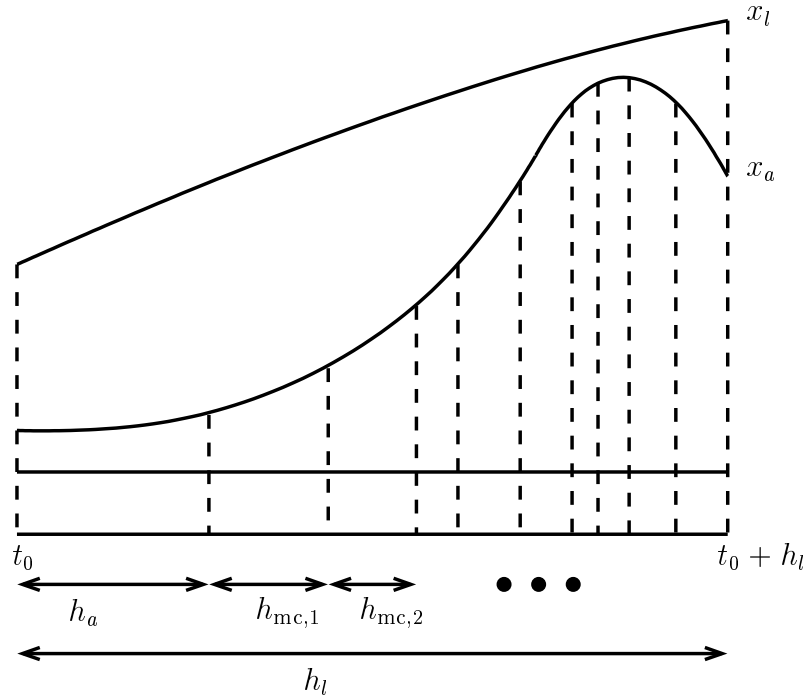


Abbildung 2.2: Diskretisierung mixed-multirate

(GM) von Kværnø/Rentrop auf, ermöglicht jedoch eine adaptive Schrittweitensteuerung in den Mikroschritten.

Dies geschieht indem im sog. **compound-step**⁵ die Zuwächse des Makro- und des ersten Mikroschrittes wie beim GM in einem Schritt errechnet werden, die **späteren** Mikroschritte aber unabhängig vom Schrittweitenverhältnis $m = h_l/h_{a,1}$ des compound-Schrittes mit neuen, an den Lösungsverlauf des aktiven Teils angepassten Schritten $h_{a,2}, h_{a,3}, \dots$ arbeiten (s. Abb. 2.2). Dies bringt neben der adaptiven Schrittweitensteuerung den Vorteil, dass abgelehnte Mikroschritte nicht mehr zur Ablehnung des gesamten Makroschrittes und aller Mikroschritte führen. Im schlimmsten Fall wird ein compound-Schritt abgelehnt. Dann müssen lediglich der Makroschritt und ein Mikroschritt zusammen neu berechnet werden

Rein anschaulich kommt damit, besser als in den anderen beschriebenen Multirate-Zugängen die rein funktionale Kopplung der verschiedenen Aktivitätsbereiche zur Geltung. Denn in der Berechnung des einen Teiles gehen zwar Daten des anderen ein. Dennoch arbeiten der latente und der aktive Teil auf weitgehend entkoppelten Diskretisierungsbereichen.

Schematisch stellt sich dieses Verfahren also folgendermaßen dar:

⁵engl.: compound=zusammengesetzt, gemischt

1. Verknüpfung von Makroschritt und erstem Mikroschritt im compound-step liefert Näherungen $x_{l,1} \approx x_l(t_0 + h_l)$ und $x_{a,1} \approx x_a(t_0 + h_{a,1})$. Das Schrittweitenverhältnis $m = h_l/h_{a,1}$ sollte nicht zu groß sein, zumal dies in die Berechnung von $\tilde{X}_{A,i}$ eingeht (deshalb „weitgehend entkoppelte Diskretisierungsbereiche“ (s. o)).
2. In den späteren Mikroschritten wird letztlich das Anfangswertproblem

$$\begin{aligned} \dot{x}_a(t) &= f_a(x_l(t), x_a(t), t) \\ x_a(t_0 + h_{a,1}) &= x_{a,1} \\ t &\in [t_0 + h_{a,1}, t_0 + h_l] \end{aligned}$$

gelöst. Werte $x_l(t)$, für den latenten Teil, liegen hier jedoch nicht exakt vor, sondern müssen in geeigneter Weise approximativ beschafft werden.

Um den Mehraufwand an Rechenarbeit möglichst gering zu halten, verwendet man das Konzept des **dense-output**⁶ ([HNW00, II.6]), um Approximationen an $x_l(t)$ auf dem Intervall $[t_0 + h_{a,1}, t_0 + h_l]$ zu erhalten.

Dieses Verfahren nutzt die zur Gewinnung von $x_{l,1} \approx x(t_0 + h_l)$ bereits berechneten RK- bzw. ROW-Zuwächse $k_{L,i}$ aus und gewichtet diese anders:

$$\begin{aligned} x_l(t_0 + \theta h_l) &\approx x_{l,0} + \sum_{i=1}^s b_{L,i}(\theta) \cdot k_{L,i} \\ \theta &\in [0, 1] \end{aligned} \tag{2.5}$$

Die Gewichte $b_{L,i}(\theta)$ sind Polynome in θ , deren Grad von der verlangten Konsistenzordnung abhängt. Dies wird zu einem späteren Zeitpunkt noch diskutiert werden.

Außerhalb der Multirate-Theorie haben dense-output-Formeln ihre Anwendung beispielsweise in der graphischen Aufbereitung numerischer Simulationen, um eine „kontinuierliche“ Lösung zu erhalten.

2.2 Multirate-Verfahren für Deskriptorsysteme

Nach der Vorstellung bestehender Multirate-Techniken für Netzwerkgleichungen in der Zustandsform (einem System gewöhnlicher Differentialgleichungen) wollen wir uns nun der Entwicklung eines Multirate-Verfahrens für Netzwerkgleichungen in Deskriptorform (einem DAE-System also) zuwenden.

Wir beschränken uns hierbei auf **spezielle Index-1-Systeme**, wie sie in der **ladungsorientierten** Modellierung elektrischer Schaltungen entstehen, die keine flussspeichernden Elemente (Induktivitäten) enthalten.

⁶engl.: dense=dicht

TCAD-Tools liefern für die große Zahl von Schaltungen, die diese Einschränkung erfüllen, Netzwerkgleichungen der Form:

$$\begin{aligned}
 \mathcal{F}(x, \dot{q}) &= A \cdot \dot{q}(x) - f(x) = 0 \\
 x(t_0) &= x_0 \\
 \text{mit } t &\in [t_0, t_{\text{end}}] \\
 x &\in \mathbb{R}^n \\
 q &: \mathbb{R}^n \rightarrow \mathbb{R}^r \\
 f &: \mathbb{R}^n \rightarrow \mathbb{R}^n \\
 A &\in \{-1, 0, 1\}^{n \times r} \\
 \dot{q}(x) &= \frac{\partial}{\partial x} q(x) \cdot \dot{x}(t)
 \end{aligned} \tag{2.6}$$

Hier stellt x die n Knotenspannungen, $[f(x)]_j$ die in den j -ten Knoten einfließenden Zweigströme statischer Elemente, die Inzidenzmatrix A beschreibt die Topologie des Netzwerkes und $[A \cdot \dot{q}(x)]_j$ die in den j -ten Knoten einfließenden Zweigströme aus dynamischen, ladungsspeichernden Elementen (Kapazitäten) dar (s. [G98]). Da ein Knoten wie bekannt ein Schaltungspunkt ist, an dem mindestens zwei Leitungen, also mindestens zwei Ströme zusammentreffen, ist die Anzahl r der Zweigströme stets größer als die Anzahl n der Knoten:

$$r > n.$$

Neben der **technischen Einschränkung**, nur Schaltungen zu betrachten, die keine flussspeichernden Bauteile besitzen, wollen wir eine **mathematische Forderung** an die Netzwerkgleichung (2.6) stellen:

$$A \cdot \frac{\partial q}{\partial x}(x) \quad \text{sei in einer Umgebung der Lösung } x(t) \text{ stetig und regulär} \tag{2.7}$$

Dies führt, wie sich zeigen wird, auf eine spezielle **Index⁷-1 Hypothese**.

Um die physikalisch-technische Multirate-Eigenschaft der elektrischen Schaltung auf ein numerisches Verfahren zu übertragen, muss auch im mathematischen Modell (2.6) eine Partitionierung in latenten und aktiven Teil erfolgen. Hier setzen wir als weitere Forderung voraus, dass eine Kopplung nur über algebraische, somit statische Elemente (wie Widerstände, Dioden, Bipolartransistoren) und nicht über dynamische Elemente (wie Kondensatoren) erfolgt. Bedenkt man jedoch, dass gerade dynamische Elemente die Eigenschaft *langsam* (*latent*) oder *schnell* (*aktiv*) besitzen, so ist diese formale Forderung in Wirklichkeit eine natürliche

⁷Wann immer im Weiteren vom **Index** die Rede ist, sei damit der **differentielle Index** gemeint

Begebenheit. Denn Knotenspannungen, die gemeinsam ein dynamisches Element steuern, werden natürlicherweise den gleichen Aktivitätsgrad aufweisen.

Nach einer eventuellen Umordnung von Unbekannten (Knotenspannungen) und Funktionen lässt sich, bei der vorausgesetzten statischen Kopplung, die Multirate-Eigenschaft auf die Netzwerkgleichung (2.6) übertragen:

$$A \cdot \dot{q}(x) - f(x) = 0 \rightsquigarrow \begin{pmatrix} A_l & 0 \\ 0 & A_a \end{pmatrix} \cdot \begin{pmatrix} \dot{q}_l(x_l) \\ \dot{q}_a(x_a) \end{pmatrix} - \begin{pmatrix} f_l(x_l, \boxed{x_a}) \\ f_a(\boxed{x_l}, x_a) \end{pmatrix} = 0 \quad (2.8)$$

Ausmultipliziert ergibt sich somit:

$$\begin{array}{ll} \text{latent} & : \quad A_l \cdot \dot{q}_l(x_l) = f_l(x_l, \boxed{x_a}) \\ \text{aktiv} & : \quad A_a \cdot \dot{q}_a(x_a) = f_a(\boxed{x_l}, x_a) \end{array} \quad (2.9)$$

Es ist nun naheliegend, für die Ladungen q_l und q_a zusätzliche Variablen z_l bzw. z_a einzuführen. Dadurch wird (2.9) zu einem System zweier gekoppelter Anfangswert-DAEs ausgebaut:

latent: $A_l \cdot \dot{z}_l = f_l(x_l, \boxed{x_a})$ $0 = z_l - q_l(x_l)$	aktiv: $A_a \cdot \dot{z}_a = f_a(\boxed{x_l}, x_a)$ $0 = z_a - q_a(x_a)$
--	---

partitionierte Netzwerkgleichung (2.10)

mit den (natürlicherweise) konsistenten Anfangswerten

$\begin{pmatrix} x_{l,0} \\ z_{l,0} \end{pmatrix} = \begin{pmatrix} x_l(t_0) \\ z_l(t_0) \end{pmatrix}$ $f_l(x_{l,0}, x_{a,0}) \in \text{Im}\left(A_l \cdot \frac{\partial q_l}{\partial x_l}(x_{l,0})\right)$ $z_{l,0} = q_l(x_{l,0})$	$\begin{pmatrix} x_{a,0} \\ z_{a,0} \end{pmatrix} = \begin{pmatrix} x_a(t_0) \\ z_a(t_0) \end{pmatrix}$ $f_a(x_{l,0}, x_{a,0}) \in \text{Im}\left(A_a \cdot \frac{\partial q_a}{\partial x_a}(x_{a,0})\right)$ $z_{a,0} = q_l(x_{a,0})$
---	---

und den Dimensionen

$x_l \in \mathbb{R}^{n_l}$ $z_l \in \mathbb{R}^{r_l}$ $A_l \in \mathbb{R}^{n_l \times r_l}$ $f_l : \mathbb{R}^{n_l} \times \mathbb{R}^{n_a} \rightarrow \mathbb{R}^{n_l}$ $q_l : \mathbb{R}^{n_l} \rightarrow \mathbb{R}^{r_l}$	$x_a \in \mathbb{R}^{n_a}$ $z_a \in \mathbb{R}^{r_a}$ $A_a \in \mathbb{R}^{n_a \times r_a}$ $f_a : \mathbb{R}^{n_l} \times \mathbb{R}^{n_a} \rightarrow \mathbb{R}^{n_a}$ $q_a : \mathbb{R}^{n_a} \rightarrow \mathbb{R}^{r_a}$
---	---

$n_l + n_a = n$

Aufgrund der Forderung (2.7) an die Netzwerkgleichung und der Struktur der Partitionierung (2.8), erfüllt das DAE-System (2.10) eine **spezielle Index-1** Bedingung:

$$\begin{aligned} A_l \cdot \frac{\partial q_l}{\partial x_l}(x_l(t)) &\in \mathbb{R}^{n_l \times n_l}, \\ A_a \cdot \frac{\partial q_a}{\partial x_a}(x_a(t)) &\in \mathbb{R}^{n_a \times n_a} \end{aligned}$$

sind in einer Umgebung der Lösung $(x_l(t), x_a(t))^T$ **stetig** und **regulär**. Damit sind auch die Anfangswerte stets konsistent, wie sich ebenfalls leicht nachrechnen lässt.

Für die Entwicklung eines Multirate-Verfahrens ist es wichtig, das spezielle Index-1-System (2.10) in ein System zu transformieren, welches ohne die Inzidenz-Matrizen A_l, A_a auf der linken Seite auskommt. Für ein solches, linear transformiertes System kann auf einfache Weise ein ROW-basiertes Multirate-Verfahren konstruiert werden. Da ROW-Verfahren invariant bezüglich linearen Transformationen sind (s. [HW96]), kann durch einfache Rücktransformation ein Multirate-Verfahren für das Ausgangsproblem (2.10) gewonnen werden.

Diese notwendige, weil vereinfachende Transformation beschreibt das folgende Lemma.

Lemma 2.1 (Transformation in gekoppeltes Hessenbergssystem)

Es gelte $A_\lambda \cdot \frac{\partial q_\lambda}{\partial x_\lambda}(x_\lambda(t))$ regulär $\forall \lambda \in \{l, a\}$ in einer Umgebung der Lösung $(x_l(t), x_a(t))^T, t \in [t_0, t_{end}]$ dann ist das gekoppelte Netzwerksystem (2.10) äquivalent zu einem **System gekoppelter Index-1 DAEs in Hessenberg-Form**⁸:

$\begin{aligned} \dot{y}_l &= w_l(x_l, x_a) \\ 0 &= y_l - g_l(x_l) \\ 0 &= \tilde{y}_l - \tilde{g}_l(x_l) \end{aligned}$	$\begin{aligned} \dot{y}_a &= w_a(x_l, x_a) \\ 0 &= y_a - g_a(x_l) \\ 0 &= \tilde{y}_a - \tilde{g}_a(x_l) \end{aligned}$	(2.11)
--	--	--------

gekoppeltes Hessenberg-System

mit

$$\begin{aligned} y_\lambda &\in \mathbb{R}^{n_\lambda} \\ \tilde{y}_\lambda &\in \mathbb{R}^{r_\lambda - n_\lambda} \\ w_\lambda &: \mathbb{R}^{n_l} \times \mathbb{R}^{n_a} \rightarrow \mathbb{R}^{n_\lambda} \quad \text{für } \lambda \in \{l, a\} \\ g_\lambda &: \mathbb{R}^{n_\lambda} \rightarrow \mathbb{R}^{n_\lambda} \\ \tilde{g}_\lambda &: \mathbb{R}^{n_\lambda} \rightarrow \mathbb{R}^{r_\lambda - n_\lambda} \end{aligned}$$

⁸Zur in Hessenberg-Form von DAEs siehe z. B. [RSW96]

Beweis 2.2 *OBdA. betrachten wir den Fall $\lambda = l$*

Da $A_l \cdot \frac{\partial q_l}{\partial x_l} \in \mathbb{R}^{n_l \times n_l}$ invertierbar ist, besitzt A_l vollen Zeilenrang n_l . Somit lässt sich mithilfe der Singulärwertzerlegung A_l aufspalten in

$$A_l = S_l \cdot \underbrace{\left(\mathcal{I}_{n_l \times n_l} \mid \mathcal{O}_{n_l \times (r_l - n_l)} \right)}_{\mathbb{R}^{n_l \times r_l}} \cdot T_l$$

mit regulären Matrizen $S_l \in \mathbb{R}^{n_l \times n_l}$ und $T_l \in \mathbb{R}^{r_l \times r_l}$, sowie

$$\begin{aligned} \mathcal{I}_{n_l \times n_l} &= \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix} \in \mathbb{R}^{n_l \times n_l}, \\ \mathcal{O}_{n_l \times (r_l - n_l)} &= \begin{pmatrix} 0 & & 0 \\ & \ddots & \\ 0 & & 0 \end{pmatrix} \in \mathbb{R}^{n_l \times (r_l - n_l)}. \end{aligned}$$

Dies eingesetzt ergibt für (2.10):

$$\begin{aligned} \left(\mathcal{I}_{n_l \times n_l} \mid \mathcal{O}_{n_l \times (r_l - n_l)} \right) \cdot T_l \cdot z_l &= S_l^{-1} \cdot f_l(x_l, x_a) \\ 0 &= T_l \cdot z_l - T_l \cdot q_l(x_l) \end{aligned} \quad (2.12)$$

Die Struktur von (2.12) führt zu einer Aufspaltung des Vektors $T_l \cdot z_l$ und der vektorwertigen Funktion $T_l \cdot q_l(x_l)$ in:

$$\begin{pmatrix} y_l \\ \tilde{y}_l \end{pmatrix} \stackrel{\text{def}}{=} T_l \cdot z_l \quad \text{mit} \quad y_l \in \mathbb{R}^{n_l}, \tilde{y}_l \in \mathbb{R}^{r_l - n_l}$$

bzw.

$$\begin{pmatrix} g_l(x_l) \\ \tilde{g}_l(x_l) \end{pmatrix} \stackrel{\text{def}}{=} T_l \cdot q_l(x_l) \quad \text{mit} \quad g_l : \mathbb{R}^{n_l} \rightarrow \mathbb{R}^{n_l}, \tilde{g}_l : \mathbb{R}^{n_l} \rightarrow \mathbb{R}^{r_l - n_l}.$$

Damit liest sich (2.12), in der geforderten Form als:

$$\begin{aligned} \dot{y}_l &= w_l(x_l, x_a) \quad (\stackrel{\text{def}}{=} S_l^{-1} \cdot f_l(x_l, x_a)) \\ 0 &= y_l - g_l(x_l) \\ 0 &= \tilde{y}_l - \tilde{g}_l(x_l). \end{aligned} \quad (2.13)$$

Bleibt noch zu zeigen, dass diese DAE den Index 1 besitzt.

Durch Ableiten des algebraischen Teils von (2.13) nach t erhält man:

$$0 = \dot{y}_l - \frac{\partial g_l}{\partial x_l} \cdot \dot{x}_l \quad (2.14a)$$

$$0 = \dot{\tilde{y}}_l - \frac{\partial \tilde{g}_l}{\partial x_l} \cdot \dot{x}_l. \quad (2.14b)$$

Offensichtlich reicht es nun, die Regularität von $\frac{\partial g_l}{\partial x_l}$ zu zeigen:

$$\begin{aligned} \frac{\partial g_l}{\partial x_l} &= \frac{\partial}{\partial x_l} \left(\mathcal{I}_{n_l \times n_l} \mid \mathcal{O}_{n_l \times (r_l - n_l)} \right) \cdot T_l \cdot q_l \\ &= \underbrace{S_l^{-1}}_{\text{reg.}} \cdot \underbrace{A_l \cdot \frac{\partial q_l}{\partial x_l}}_{\text{reg. nach Vorr.}} \quad \text{regulär} \quad \square \end{aligned}$$

Für das gekoppelte Hessenbergsystem (2.11) soll nun ein Multirate-Verfahren zur numerischen Berechnung aller Unbekannten entwickelt werden. Die besondere Struktur dieses Systems wird nun ausgenutzt, um Approximationen für $x_l, x_a, y_l, y_a, \tilde{y}_l, \tilde{y}_a$ zu erhalten:

- Die ersten beiden Zeilen des gekoppelten Hessenbergsystems bilden, wie man leicht sieht, bereits eine Index-1-DAE in Hessenberg-Form. Deswegen wird für diesen Teil ein Multirate-Verfahren in den Unbekannten x_l, x_a, y_l, y_a entwickelt.

Die bestimmenden Gleichungen lauten:

$$\begin{array}{|c|c|} \hline \begin{array}{l} \dot{y}_l = w_l(x_l, x_a) \\ 0 = y_l - g_l(x_l) \end{array} & \begin{array}{l} \dot{y}_a = w_a(x_l, x_a) \\ 0 = y_a - g_a(x_l) \end{array} \\ \hline \end{array} \quad (2.15a)$$

- In der dritten Zeile des gekoppelten Hessenbergsystems tritt keine Kopplung mehr auf. Die Unbekannten $\tilde{y}_l, \tilde{y}_a$ ergeben sich aus

$$\begin{aligned} \tilde{y}_l &= \tilde{g}_l(x_l) \\ \tilde{y}_a &= \tilde{g}_a(x_a) \end{aligned} \quad (2.15b)$$

Multirate-Verfahren für das gekoppelte Hessenbergsystem

Die Lösung des gekoppelten Hessenbergsystems (2.11), liegt, wie gesehen in der Lösung des Teilproblems (2.15a), in den Unbekannten x_l, x_a, y_l, y_a und der Lösung von (2.15b) in $\tilde{y}_l, \tilde{y}_a$.

Im eigentlichen Sinne einer numerischen Verfahrensdefinition stellt letzteres keine Schwierigkeiten dar, ist doch nur eine Funktionsauswertung nötig, um $\tilde{y}_l, \tilde{y}_a$ aus den zuvor berechneten x_l, x_a zu erhalten.

Wir werden nun ein Multirate-Verfahren für das Teilproblem (2.15a) entwickeln, welches die Idee des **mixed-multirate** von Bartel/Günther/Kværnø ([BGK]) mit der Arbeit von Günther zu **ROW-Verfahren für ladungsorientierte Netzwerkgleichungen** verbindet. Ersteres liefert die Vorteile der Partitionierung von Netzwerkgleichungen in aktive und latente Komponenten, das zweite liefert eine adaptive Schrittweitensteuerung auf den beiden Aktivitätslevel über die jeweils algebraischen Unbekannten x_l, x_a . Eine solche Schrittweitensteuerung wird angestrebt, da wir uns in der Simulation für die Spannungen interessieren, die im ladungsorientierten Netzwerkansatz als algebraische Unbekannte x_l, x_a auftreten.

Vom mixed-multirate-Ansatz wird zunächst die Idee übernommen, eine Zerteilung der numerischen Berechnung in **compound-Schritt** und **restliche Mikroschritte** vorzunehmen.

Teil 1: compound-Schritt

Da die Einkopplung des jeweils anderen Aktivitätslevels im Teilproblem (2.15a) für aktiven und latenten Teil in gleicher Weise geschieht und der eigentliche Unterschied in der numerischen Berechnung in den Schrittweiten h_l, h_a liegt, wird eine Verfahrensherleitung nur für den **latenten Teil** durchgeführt. Das Vorgehen für den aktiven Part ist entsprechend.

Durch ε -**Einbettung** – der algebraische Teil der DAE wird in eine künstliche Differentialgleichung eingebettet (s.[HW96, VI]) – geht (2.15a) über in ein ODE-System:

$$\begin{array}{rclclcl} \dot{y}_l & = & w_l(x_l, x_a) & \rightsquigarrow & \dot{y}_l & = & w_l(x_l, x_a) & \rightsquigarrow & \dot{y}_l & = & w_l(x_l, x_a) \\ 0 & = & y_l - g_l(x_l) & & \varepsilon \dot{x}_l & = & y_l - g_l(x_l) & & \dot{x}_l & = & \frac{1}{\varepsilon}(y_l - g_l(x_l)) \end{array} \quad (2.16)$$

Von rechts nach links gelesen, ergibt sich, dass man aus dem ODE-System nach Multiplikation der zweiten Zeile mit ε und dem anschließenden Nullsetzen von ε zur Ausgangs-DAE kommt.

Auf die ODE lässt sich nun ein s-stufiges **ROW-Verfahren** anwenden (s.[HW96, IV.7]):

$$\begin{pmatrix} y_{l,1} \\ x_{l,1} \end{pmatrix} = \begin{pmatrix} y_{l,0} \\ x_{l,0} \end{pmatrix} + \sum_{i=1}^s b_i^L \begin{pmatrix} l_i^L \\ k_i^L \end{pmatrix}$$

Die Zuwächse sind gegeben durch

$$\begin{pmatrix} l_i^L \\ k_i^L \end{pmatrix} = h_l \cdot \begin{pmatrix} w_l(x_{l,0} + \sum_{j=1}^{i-1} \alpha_{ij}^L k_j^L, \boxed{\tilde{X}_{A,i}}) \\ \frac{1}{\varepsilon}(y_{l,0} + \sum_{j=1}^{i-1} \alpha_{ij}^L l_j^L - g_l(x_{l,0} + \sum_{j=1}^{i-1} \alpha_{ij}^L k_j^L)) \end{pmatrix} \\ + h_l \cdot J \cdot \sum_{j=1}^i \gamma_{ij}^L \begin{pmatrix} l_j^L \\ k_j^L \end{pmatrix}$$

mit der Jacobi-Matrix:

$$J = \begin{pmatrix} 0 & \frac{\partial w_l}{\partial x_l} \\ \mathcal{I}_{n_l \times n_l} & \frac{\partial g_l}{\partial x_l} \end{pmatrix}$$

und den Verfahrenskoeffizienten ($\lambda \in \{L, A\}$):

$$\begin{aligned}
\mathcal{A}^\lambda &= \begin{pmatrix} 0 & \cdots & 0 \\ \alpha_{21}^\lambda & \ddots & \vdots \\ \vdots & & \\ \alpha_{s1}^\lambda & \cdots & \alpha_{s(s-1)}^\lambda & 0 \end{pmatrix} & \mathcal{D}^\lambda &= \begin{pmatrix} 0 & \cdots & 0 \\ \delta_{21}^\lambda & \ddots & \vdots \\ \vdots & & \\ \delta_{s1}^\lambda & \cdots & \delta_{s(s-1)}^\lambda & 0 \end{pmatrix} \\
\mathcal{G}^\lambda &= \begin{pmatrix} \gamma_{21}^\lambda & \cdots & 0 \\ \gamma_{21}^\lambda & \ddots & \vdots \\ \vdots & & \\ \gamma_{s1}^\lambda & \cdots & \gamma_{s(s-1)}^\lambda & \gamma^\lambda \end{pmatrix} & \mathcal{B}^\lambda &= \left(\beta_{ij}^\lambda \right)_{i,j=1,\dots,s} = \mathcal{A}^\lambda + \mathcal{G}^\lambda \\
b^\lambda &= \begin{pmatrix} b_1^\lambda \\ \vdots \\ b_s^\lambda \end{pmatrix} & & \mathcal{A}^\lambda, \mathcal{B}^\lambda, \mathcal{D}^\lambda, \mathcal{G}^\lambda \in \mathbb{R}^{s \times s} \\
& & & b^\lambda \in \mathbb{R}^s.
\end{aligned}$$

Die partiellen Ableitungen werden an den Stellen $(x_l(t_0), x_a(t_0))$ bzw. $x_l(t_0)$ (bei w_l resp. g_l) ausgewertet.

Für die Kopplungsgröße $\tilde{X}_{A,i} \approx x_a(t_0 + \alpha_i^L h_l)$ (vgl. (2.4)) wird ein ROW-ähnliches Schema gewichteter Zuwächse aus der aktiven Ebene angesetzt:

$$\tilde{X}_{a,i} \stackrel{\text{def}}{=} x_{a,0} + \frac{h_l}{h_a} \sum_{j=1}^s \delta_{ij}^L k_j^A$$

D.h. die analytische Einkopplung vom aktiven in den latenten Teil wird im numerischen Verfahren abgebildet auf die Einkopplung der Zuwächse k_i^A aus der aktiven in die latente Schicht.

Nun ist also für die in (2.16) künstlich erzeugte ODE bereits ein s-stufiges Multirate-Verfahren definiert. Durch **Umkehrung der ε -Einbettung** erhält man aus der künstlich erzeugten ODE das eingebettete DAE-Problem. Durch Umkehrung der ε -Einbettung in gleicher Weise für das numerische Verfahren erhält man ein Multirate-Verfahren für das Teilproblem (2.15a).

Mit der Vereinfachung $\gamma_{ii}^\lambda = \gamma^\lambda$ ($\lambda = L, A$) $\forall i = 1, \dots, s$) ergibt sich ein **einfach-diagonal-implizites**⁹ Verfahren. Wie den Koeffizienten zu entnehmen ist, ist dieses Verfahren in der Einkopplung des jeweils anderen Aktivitätslevels nicht-implizit ($\delta_{ii}^L = 0, i = 1, \dots, s$).

Für den latenten Teil stellt sich das Multirate-Verfahren für das Teilproblem (2.15a) somit folgendermaßen dar:

$$\begin{aligned}
x_{l,1} &= x_{l,0} + \sum_{i=1}^s b_i^L \cdot k_i^L \\
y_{l,1} &= y_{l,0} + \sum_{i=1}^s b_i^L \cdot l_i^L
\end{aligned} \tag{2.17a}$$

⁹in englischsprachigen Veröffentlichungen als singly-diagonal-implicit Runge-Kutta- bzw. Rosenbrock-Wanner-Methods bezeichnet

Die Zuwächse bestimmen sich aus

$$\begin{aligned}
l_i^L - h_l \frac{\partial w_l}{\partial x_l} \gamma^L k_i^L &= h_l w_l(x_{l,0} + \sum_{j=1}^{i-1} \alpha_{ij}^L k_j^L, x_{a,0} + m \sum_{j=1}^{i-1} \delta_{ij}^L k_j^A) \\
&\quad + h_l \frac{\partial w_l}{\partial x_l} \sum_{j=1}^{i-1} \gamma_{ij}^L k_j^L \\
-\gamma^L l_i^L + \gamma^L \frac{\partial g_l}{\partial x_l} k_i^L &= y_{l,0} - g_l(x_{l,0} + \sum_{j=1}^{i-1} \alpha_{ij}^L k_j^L) \\
&\quad + \sum_{j=1}^{i-1} \beta_{ij}^L l_j^L - \frac{\partial g_l}{\partial x_l} \sum_{j=1}^{i-1} \gamma_{ij}^L k_j^L
\end{aligned} \tag{2.17b}$$

mit dem Schrittweitenverhältnis: $m = \frac{h_l}{h_a}$.

Die Bestimmung der Unbekannten $(x_{l,1}, y_{l,1})$ mittels (2.17a) birgt jedoch ein Problem:

$(x_{l,1}, y_{l,1})$ sind Näherungen an die Lösung $(x_l(t_0 + h_l), y_l(t_0 + h_l))$ des Teilproblems (2.15a), sie müssen also auch die algebraische Bedingung

$$y_{l,1} - g_l(x_{l,1}) = 0$$

erfüllen.

Dies kann jedoch mit der inkrementellen Bestimmung von $x_{l,1}$ und $y_{l,1}$ in (2.17a) nicht garantiert werden.

Abhilfe schafft hier die Bestimmung der differentiellen Unbekannten $y_{l,1}$ mittels der einfachen Funktionsauswertung

$$y_{l,1} = g_l(x_{l,1}) \tag{2.17c}$$

Anschaulich entspricht dies einer Projektion der Approximation auf die Mannigfaltigkeit

$$\mathcal{M}_l = \{(x, y) | y - g_l(x) = 0\},$$

auf der die Lösung von (2.15a) zu verlaufen hat.

Somit ist der compound-Schritt eines **Multirate-Verfahrens** zur Lösung des **gekoppelten Hessenbergsystems (2.11)** für den latenten Teil gegeben durch:

$$\begin{aligned}
 x_{l,1} &= x_{l,0} + \sum_{i=1}^s b_i^L \cdot k_i^L, \\
 y_{l,1} &= g_l(x_{l,1}), \\
 \tilde{y}_{l,1} &= \tilde{g}_l(x_{l,1}),
 \end{aligned}
 \tag{2.18}$$

mit den Zuwächsen k_i^L, l_i^L aus (2.17b).

compound-Schritt, nichttransformiertes Multirate-Verfahren für Hessenbergsystem

Im aktiven Fall gestaltet sich die Definition analog, dabei wird in der Bestimmung der Zuwächse m durch $1/m$ ersetzt.

Auf obigem Verfahren wird später (Kap. 3) die **Konsistenzbetrachtung im compound-Schritt** beruhen, somit ist es für die weiteren theoretischen Betrachtungen von großer Bedeutung.

Wie bereits geschildert, werden wir aus Verfahren (2.18) durch lineare Transformation ein Multirate-Verfahren für die Netzwerkgleichung (2.10) erhalten. An dieser Stelle sei nochmals darauf hingewiesen, dass ROW-Verfahren invariant bezüglich linearer Transformationen sind, diese also keine Auswirkungen auf die Konsistenz- und Konvergenzeigenschaften haben.

Im Hinblick auf eine möglichst effiziente Implementierung eines Multirate-Verfahrens fällt auf, dass die Zuwächse l_i^L nur in die Gewichte k_i^L eingehen, jedoch nicht in die Approximation $x_{l,1}$.

Mithilfe einiger linearer Transformationen ist es möglich, dies zu umgehen, d. h. nur mit einer „Sorte“ von Gewichten κ_i^L rechnen zu müssen.

Nach einer Vorstellung der nötigen Transformationen, wird die Überführung anhand des latenten Teils kurz vorgestellt (s. hierzu auch [G98]).

Transformationen:

$$\begin{aligned}
 \kappa^\lambda &= \begin{pmatrix} \kappa_1^\lambda \\ \vdots \\ \kappa_s^\lambda \end{pmatrix} = \mathcal{G}^\lambda \cdot \begin{pmatrix} k_1^\lambda \\ \vdots \\ k_s^\lambda \end{pmatrix}, \quad d^\lambda = \begin{pmatrix} d_1^\lambda \\ \vdots \\ d_s^\lambda \end{pmatrix} = (\mathcal{G}^\lambda)^{-T} \cdot \begin{pmatrix} b_1^\lambda \\ \vdots \\ b_s^\lambda \end{pmatrix} \\
 \sigma^\lambda &= \left(\sigma_{ij}^\lambda \right)_{i,j=1,\dots,s} = \mathcal{A}^\lambda \cdot (\mathcal{G}^\lambda)^{-1} = \begin{pmatrix} 0 & \cdots & 0 \\ & \ddots & \vdots \\ * & & 0 \end{pmatrix} \\
 \varrho^\lambda &= \left(\varrho_{ij}^\lambda \right)_{i,j=1,\dots,s} = \mathcal{D}^\lambda \cdot (\mathcal{G}^\lambda)^{-1} = \begin{pmatrix} 0 & \cdots & 0 \\ & \ddots & \vdots \\ * & & 0 \end{pmatrix} \\
 \{\lambda, \bar{\lambda}\} &= P = \{L, A\}
 \end{aligned}
 \tag{2.19}$$

Bem.: $(\mathcal{G}^\lambda)^{-1}$ existiert, da implizites Verfahren, also $\gamma^\lambda \neq 0$

Mit den neuen Zuwächsen κ_i^L und Gewichten d_i^L berechnet sich $x_{l,1}$ nun als:

$$x_{l,1} = x_{l,0} + \sum_{i=1}^j d_i^L \cdot \kappa_i^L.$$

Die Zuwächse l_i^L, κ_i^L bestimmen sich aus:

$$\begin{pmatrix} \mathcal{I}_{s \times s} & -h_l \frac{\partial w_l}{\partial x_l} \\ -\gamma^L \mathcal{I}_{s \times s} & \frac{\partial g_l}{\partial x_l} \end{pmatrix} \begin{pmatrix} l_i^L \\ \kappa_i^L \end{pmatrix} = \begin{pmatrix} h_l w_l(\tilde{a}_{L,i}, \tilde{d}_{L,i}) \\ y_{l,0} - g_l(\tilde{a}_{L,i}) + \sum_{j=1}^{i-1} \beta_{ij}^L l_j^L \end{pmatrix} \quad (2.20)$$

mit $\tilde{a}_{L,i} \stackrel{\text{def}}{=} x_{l,0} + \sum_{j=1}^i \sigma_{ij}^L \kappa_j^L$

$\tilde{d}_{L,i} \stackrel{\text{def}}{=} x_{a,0} + m \sum_{j=1}^i \varrho_{ij}^L \kappa_j^A \quad (m = \frac{h_l}{h_a})$

Die Matrix

$$\begin{pmatrix} 0 & \frac{\partial g_l}{\partial x_l} - h_l \gamma^L \frac{\partial w_l}{\partial x_l} \\ -\gamma^L \mathcal{I}_{s \times s} & \frac{\partial g_l}{\partial x_l} \end{pmatrix},$$

die sich durch einfache Gauß-Elimination aus der Matrix des linearen Gleichungssystems (2.20) ergibt, ist für genügend kleine h_l regulär, da $\frac{\partial g_l}{\partial x_l}$ in einer Umgebung der Lösung regulär ist.

Dies führt zusammen mit

$$l_i^L = h_l w_l(\tilde{a}_{L,i}, \tilde{d}_{L,i}) + h_l \frac{\partial w_l}{\partial x_l} \kappa_i^L$$

auf die Bestimmung der Zuwächse κ_i^L , die nunmehr unabhängig von den Werten l_i^L ist:

$$\left(\frac{\partial g_l}{\partial x_l} - h_l \gamma^L \frac{\partial w_l}{\partial x_l} \right) \kappa_i^L = y_{l,0} - g_l(\tilde{a}_{L,i}) + h_l \sum_{j=1}^i \beta_{ij}^L w_l(\tilde{a}_{L,i}, \tilde{d}_{L,i}) + h_l \sum_{j=1}^{i-1} \beta_{ij}^L \frac{\partial w_l}{\partial x_l} \kappa_j^L$$

Somit ist der **compound-Schritt** eines Multirate-Verfahrens für das gekoppelte Hessenbergsystem (2.10) gegeben durch:

$x_{l,1} = x_{l,0} + \sum_{i=1}^j d_i^L \cdot \kappa_i^L$	$x_{a,1} = x_{a,0} + \sum_{i=1}^j d_i^A \cdot \kappa_i^A$
mit: $\left(\frac{\partial g_l}{\partial x_l} - h_l \gamma^L \frac{\partial w_l}{\partial x_l} \right) \kappa_i^L =$ $g_l(x_{l,0}) - g_l(\tilde{a}_{L,i}) + h_l \sum_{j=1}^i \beta_{ij}^L w_l(\tilde{a}_{L,i}, \tilde{d}_{L,i}) + h_l \sum_{j=1}^{i-1} \beta_{ij}^L \frac{\partial w_l}{\partial x_l} \kappa_j^L$ $\left(\frac{\partial g_a}{\partial x_a} - h_a \gamma^A \frac{\partial w_a}{\partial x_a} \right) \kappa_i^A =$ $g_a(x_{a,0}) - g_a(\tilde{a}_{A,i}) + h_a \sum_{j=1}^i \beta_{ij}^A w_l(\tilde{d}_{A,i}, \tilde{a}_{A,i}) + h_a \sum_{j=1}^{i-1} \beta_{ij}^A \frac{\partial w_a}{\partial x_a} \kappa_j^A$ $\tilde{a}_{\lambda,i} = x_{\lambda,0} + \sum_{j=1}^{i-1} \sigma_{ij}^\lambda \kappa_j^\lambda \qquad \tilde{d}_{\lambda,i} = x_{\bar{\lambda},0} + m^{\chi(\lambda)} \sum_{j=1}^{i-1} \varrho_{ij}^\lambda \kappa_j^{\bar{\lambda}}$ $m = \frac{h_l}{h_a} \qquad \{\lambda, \bar{\lambda}\} = \{L, A\}$ $\chi : \{L, A\} \rightarrow \{-1, 1\} \qquad \chi(L) = 1 \qquad \chi(A) = -1$	

compound-Schritt für das gekoppelte Hessenbergsystem (2.21)

In obiger Verfahrensdefinition wurden die Unbekannten $y_{l,1}, \tilde{y}_{l,1}, y_{a,1}, \tilde{y}_{a,1}$ nicht mehr erwähnt. Dies liegt zum einen darin begründet, dass sie durch einfache Funktionsauswertungen (s. Verfahren (2.18)) zu bekommen sind. Zum anderen wurden sie nur zu Hilfszwecken eingeführt und repräsentieren die Ströme, die in der elektrischen Simulation nicht von Interesse sind. Bei der Übertragung von (2.21) auf die partitionierte Netzwerkgleichung (2.10) werden wir uns den Strömen nochmals zuwenden.

Teil 2: Spätere Mikroschritte

Nach der Durchführung des compound-Schritts existieren eine Näherung für den latenten Teil zum Zeitpunkt $t_0 + h_l$ und eine Näherung für den aktiven Teil zum Zeitpunkt $t_0 + h_a$:

$$\begin{aligned} x_{l,1} &\approx x_l(t_0 + h_l) \\ x_{a,1} &\approx x_a(t_0 + h_a), \quad h_a < h_l \end{aligned}$$

In einem zweiten Schritt müssen nun Approximationen an den aktiven Teil im Zeitintervall $(t_0 + h_a, t_0 + h_l]$ bestimmt werden. Wie im mixed-multirate ([BGK]) geschieht dies durch ein Verfahren, dessen Schrittweiten adaptiv gesteuert werden, also weitgehend unabhängig von den compound-Schrittweiten h_l, h_a sind.

Unter der Annahme, wir besäßen genaue Werte $x_l(t)$ im genannten Intervall, bestünden die **späteren Mikroschritte** lediglich aus der Lösung des nichtautonomen Index-1-Problems:

$$\begin{aligned} \dot{y}_a &= \tilde{w}_a(x_a, t) \quad (\stackrel{\text{def}}{=} w_a(x_l(t), x_a(t))) \\ 0 &= y_a - g_a(x_a) \\ 0 &= \tilde{y}_a - \tilde{g}_a(x_a), \end{aligned} \tag{2.22}$$

Dieses Problem könnte nun im Prinzip mit jedem adaptiven DAE-Löser angegangen werden.

Nun besitzen wir jedoch nicht die exakten Werte $x_l(t)$ im Intervall $[t_0 + h_a, t_0 + h_l]$, sondern müssen mit geeigneten Approximationen arbeiten.

Diese Näherungen an die Kopplungsgösse liefert, wie im mixed-multirate, das Konzept des **dense-output** (vgl. (2.5)):

$$\begin{aligned} x_l(t_0 + \theta h_l) &\approx x_{l,0} + \sum_{i=1}^s d_i^L(\theta) \cdot \kappa_i^L \\ \theta &\in [0, 1], \end{aligned}$$

welches bereits berechnete Zuwächse κ_i^L anders gewichtet. Die Gewichte $d_i^L(\theta)$ sind Polynome in θ .

Das angestrebte Multirate-Verfahren für das gekoppelte Hessenbergsystem (2.11) benutzt also in den **späteren Mikroschritten** ein Verfahren, das einen Index-1-DAE-Löser mit dem dense-output-Schema zur Bestimmung der Kopplung $x_l(t)$ kombiniert.

Um auch in den späteren Mikroschritten eine Schrittweitensteuerung auf Basis der algebraischen Unbekannten x_a , der Spannungen also, zu erhalten muss darauf geachtet werden, dass der gewählte Index-1-DAE-Löser bereits auf einer solchen Schrittweitensteuerung beruht. Hier bietet sich der CHORAL-Algorithmus von Günther ([G98]) an.

Da wir nicht an einer Implementierung eines Multirate-Verfahrens für Netzwerkgleichungen in der Form des gekoppelten Hessenbergsystems (2.11), sondern in der allgemeineren Form der partitionierten Netzwerkgleichungen (2.10) verzichten wir an dieser Stelle auf eine Definition der späteren Mikroschritte.

Im nächsten Abschnitt, in dem wir die bereits erzielten Ergebnisse auf die partitionierte Netzwerkgleichung übertragen, wird eine darauf angepasste Definition gegeben.

Die Konsistenzbetrachtung für die späteren Mikroschritte reduziert sich, wie zu sehen sein wird, auf „getrennte“ Genauigkeitsforderungen an das dense-output-Schema und den zugrundeliegenden Index-1-DAE-Löser.

Multirate-Verfahren für die partitionierte Netzwerkgleichung

Nach den bisherigen Anstrengungen sind wir nun in der Lage, ein **Multirate-Verfahren** für Modellgleichungen in Form der **partitionierten Netzwerkgleichung (2.10)** zu definieren.

Hierzu wenden wir die linearen Transformationen aus dem Lemma (2.1), (Transformation in gekoppeltes Hessenbergsystem) in umgekehrter Richtung auf das Multirate-Verfahren aus dem vorigen Abschnitt an. Dies wird dies latenten Fall kurz aufgezeigt.

Werden im Multirate-Verfahren (2.21) für das gekoppelte Hessenberg-System die im Beweis zu Lemma (2.1) durchgeführten Transformationen und Umbenennungen

$$\begin{aligned} g_l(x_l) &= S_l^{-1} \cdot A_l \cdot q_l(x_l) \\ \tilde{g}_l(x_l) &= \left(\mathcal{O}_{(r_l-n_l) \times n_l} \mid \mathcal{I}_{(r_l-n_l) \times (r_l-n_l)} \right) \cdot T_l \cdot q_l(x_l) \\ w_l(x_l, x_a) &= S_l^{-1} \cdot f_l(x_l, x_a) \end{aligned}$$

eingesetzt, so ergibt sich für die Zuwächse κ_i^L :

$$\begin{aligned} \left(\frac{\partial}{\partial x_l} (S_L^{-1} A_L q_l) - h_l \gamma^L \frac{\partial}{\partial x_l} (S_L^{-1} f_l) \right) \kappa_i^L &= S_L^{-1} A_L (q_l(x_{l,0}) - q_l(\tilde{a}_{L,i})) \\ &+ h_l \sum_{j=1}^i \beta_{ij}^L S_L^{-1} f_l(\tilde{a}_{L,j}, \tilde{d}_{L,j}) \\ &+ h_l \sum_{j=1}^{i-1} \beta_{ij}^L \frac{\partial}{\partial x_l} (S_L^{-1} f_l) \kappa_j^L \end{aligned}$$

Damit ist ein Multirate-Verfahren für (2.10) gegeben durch

compound-Schritt:

$$x_{l,1} = x_{l,0} + \sum_{i=1}^s d_i^L \cdot \kappa_i^L$$

$$x_{a,1} = x_{a,0} + \sum_{i=1}^s d_i^A \cdot \kappa_i^A$$

mit:

$$\begin{aligned} (A_l \frac{\partial q_l}{\partial x_l} - h_l \gamma^L \frac{\partial f_l}{\partial x_l}) \kappa_i^L &= A_l (q_l(x_{l,0}) - q_l(\tilde{a}_{L,i})) \\ &\quad + h_l \sum_{j=1}^i \beta_{ij}^L f_l(\tilde{a}_{L,j}, \tilde{d}_{L,j}) + h_l \sum_{j=1}^{i-1} \beta_{ij}^L \frac{\partial f_l}{\partial x_l} \kappa_j^L \\ (A_a \frac{\partial q_a}{\partial x_a} - h_a \gamma^A \frac{\partial f_a}{\partial x_a}) \kappa_i^A &= A_a (q_a(x_{a,0}) - q_a(\tilde{a}_{A,i})) \\ &\quad + h_a \sum_{j=1}^i \beta_{ij}^A f_l(\tilde{d}_{A,j}, \tilde{a}_{A,j}) + h_a \sum_{j=1}^{i-1} \beta_{ij}^A \frac{\partial f_a}{\partial x_a} \kappa_j^A \\ \tilde{a}_{\lambda,i} &= x_{\lambda,0} + \sum_{j=1}^{i-1} \sigma_{ij}^\lambda \kappa_j^\lambda \quad \tilde{d}_{\lambda,i} = x_{\bar{\lambda},0} + m^{\chi(\lambda)} \sum_{j=1}^{i-1} \varrho_{ij}^\lambda \kappa_j^{\bar{\lambda}} \end{aligned}$$

$$m = \frac{h_l}{h_a}$$

$$\{\lambda, \bar{\lambda}\} = \{L, A\}$$

$$\chi: \{L, A\} \rightarrow \{-1, 1\}$$

$$\chi(L) = 1$$

$$\chi(A) = -1$$

spätere Mikroschritte:

$$\theta = \frac{1}{h_l} h_a, \frac{1}{h_l} (h_a + h_{\text{mc},1}), \dots, \underbrace{\frac{1}{h_l} (h_a + \dots + h_{\text{mc},\mu})}_{=1}$$

$$x_{a,1+\nu} = x_{a,\nu} + \sum_{i=1}^{s_C} d_i^C \cdot \kappa_i^C \quad (\nu = 1, \dots, \mu)$$

mit:

$$\begin{aligned} (A_a \frac{\partial q_a}{\partial x_a} \Big|_{(x_a,\nu)} - h_l \gamma^C \frac{\partial f_a}{\partial x_a} \Big|_{(x_l^{ds}(\theta), x_a,\nu)}) \kappa_i^C &= A_a (q_a(x_{a,\nu-1}) - q_a(\tilde{a}_{C,\nu})) \\ &\quad + h_{\text{mc},\nu} \sum_{j=1}^i \beta_{ij}^C f_a(x_l^{ds}(\theta), \tilde{a}_{C,j}) + h_{\text{mc},\nu} \sum_{j=1}^{i-1} \beta_{ij}^C \frac{\partial f_a}{\partial x_a} \Big|_{(x_l^{ds}(\theta), x_a,\nu)} \kappa_j^C \\ \tilde{a}_{C,i} &= x_{a,\nu} + \sum_{j=1}^{i-1} \sigma_{ij}^C \kappa_j^C \quad x_l^{ds}(\theta) = x_{l,0} + \sum_{j=1}^i d_j^L(\theta) \kappa_j^L \end{aligned}$$

Die Verfahrenskoeffizienten $d_i^C, \beta_{ij}^C, \sigma_{ij}^C, \gamma^C$ der späteren Mikroschritte sind die Verfahrenskoeffizienten des zugrundeliegenden s_C -stufigen Index-1-Lösers, in unserem Fall also die des CHORAL-Verfahrens. Diese werden an späterer Stelle im Einzelnen aufgeführt.

Um die Berechnung der differentiellen Variablen konsistent zu halten, bleibt zu zeigen, dass sich die Ströme z_l, z_a der ursprünglichen partitionierten Netzwerkgleichung (2.10) als $z_l = q_l(x_l)$ bzw. $z_a = q_a(x_a)$ ergeben.

Nach den Transformationen aus dem Beweis zum Transformations-Lemma (2.1) gilt für den latenten Teil:

$$T_l \cdot q_l(x_l) = \begin{pmatrix} g_l(x_l) \\ \tilde{g}_l(x_l) \end{pmatrix} = \begin{pmatrix} y_l \\ \tilde{y}_l \end{pmatrix} = T_l \cdot z_l$$

und da T_l regulär ist, also $z_l = q_l(x_l)$. Für den aktiven Teil gilt Entsprechendes.

Eingebettetes Verfahren zur Schrittweitensteuerung

Der im jeweiligen Schritt gemachte Fehler zwischen Näherung $x_{l,1}, x_{a,1}, x_{a,2}, \dots$ und exaktem Werte $x_l(t), x_a(t)$ kann, aus Ermangelung der exakten Werte, nicht exakt bestimmt werden. Deswegen werden zusätzliche Approximationen $\hat{x}_{l,1}, \dots$ anderer (in unserem Fall niedrigerer Ordnung) bestimmt, und mit diesen der Fehler geschätzt:

$$\text{z. B.: } \|x_{l,1} - x_l(t_0 + h_l)\| \rightsquigarrow \|x_{l,1} - \hat{x}_{l,1}\|$$

Um den zusätzlichen Berechnungsaufwand möglichst gering zu halten, werden wir ein **eingebettetes Verfahren** benutzen. Dieses verwendet dieselben Zuwächse $\kappa_i^L, \dots$, gewichtet sie jedoch anders:

$$\begin{aligned} \hat{x}_{l,1} &= x_{l,0} + \sum_{i=1}^{\hat{s}} \hat{d}_i^L \cdot \kappa_i^L \\ \hat{x}_{a,1} &= x_{a,0} + \sum_{i=1}^{\hat{s}} \hat{d}_i^A \cdot \kappa_i^A \\ \hat{x}_{a,1+\nu} &= x_{a,\nu} + \sum_{i=1}^{\hat{s}_C} \hat{d}_i^C \cdot \kappa_i^C \end{aligned} \tag{2.24}$$

Eingebettetes Verfahren

Im Multirate-Verfahren für das gekoppelte Hessenbergsystem sind die Gewichte entsprechend $\hat{b}_i^L, \hat{b}_i^A, \hat{b}_i^C$.

Die neue Stufenzahlen $\hat{s}, \hat{s}_C$ richten sich nach der Ordnung der zweiten Näherungen $\hat{x}_{l,1}, \dots$

Zusammenfassend können wir festhalten:

Nach einer linearen Transformation der partitionierten Netzwerkgleichung (2.10) in die spezielle Form des gekoppelten Hessenbergsystems wurde für dieses Hessenbergsystem ein ROW-basiertes Multirate-Verfahren (2.18) entwickelt. Um eine möglichst effiziente spätere Implementierung zu ermöglichen, haben wir dieses erste Multirate-Verfahren (2.18) mittels weiterer linearer Transformationen in das Multirate-Verfahren (2.21) überführt. Dieses rechnet nurmehr mit den Zuwächsen $\kappa_i^L, \dots$ und nicht mehr mit Zuwächsen $k_i^L, l_i^L, \dots$. Schließlich erhielten wir durch eine lineare Transformation hieraus ein Multirate-Verfahren für das Ausgangsproblem der partitionierten Netzwerkgleichung (2.10).

Das weitere Vorgehen, Untersuchung der numerischen Eigenschaften und Implementierung, lässt sich mit folgendem Satz kurz zusammenfassen:

Die numerischen Eigenschaften (Konsistenz, Konvergenz, Stabilität) untersuchen wir anhand des nichttransformierten Multirate-Verfahrens (2.18) für das gekoppelte Hessenbergsystem, implementieren werden wir das Multirate-Verfahren (2.23) für die partitionierte Netzwerkgleichung.

Gerechtfertigt ist dies durch die spezielle Eigenschaft von ROW-Verfahren. Sie sind invariant bzgl. linearen Transformationen.

Kapitel 3

Verfahrenskonstruktion

Nach der Verfahrensdefinition des vorigen Kapitels steht nun die Verfahrenskonstruktion – die Bestimmung geeigneter **Verfahrenskoeffizienten** – im Mittelpunkt. Die Verfahrenskoeffizienten sind dabei so zu wählen, dass sie die **Konsistenz** im compound-Schritt und in den späteren Mikroschritten gewährleisten. Wie sich zeigen wird, kann hieraus bereits auf **Konvergenz** geschlossen werden. Da für differential-algebraisch Systeme noch keine der *Dahlquistischen Testgleichung für ODEs* entsprechende Referenzfunktion bekannt ist, wird auf den Begriff der „**Stabilität**“ nur kurz eingegangen werden.

Die Konsistenzanalyse gründet, wie für Differentialgleichungslöser üblich, auf einem Abgleich der Taylorreihen von exakter Lösung und numerischer Approximation. Im **compound-Schritt** werden wir hierfür aus der ODE-Analyse die Idee übernehmen, auftauchende Differentiale durch graphische Objekte, sogenannte **Butcher-Bäume** (vgl. [HNW00, II.2]), zu repräsentieren. Diese werden wir an die Struktur des gekoppelten Hessenbergsystems (2.11) entsprechend anpassen. Mit einer neu entwickelten Klasse von Bäumen können wir dann sehr einfach Bedingungsgleichungen für die Verfahrenskoeffizienten angeben.

Wie zu sehen sein wird, reduziert sich die Konsistenzanalyse in den **späteren Mikroschritten** auf eine Konsistenzbedingung des **dense-output-Schemas**, wenn der „übergeordnete“ Index-1-Löser (in unserem Fall CHORAL) bereits die geforderte Konsistenzordnung besitzt.

Bei der Verfahrensdefinition des letzten Kapitels wurden folgende Schritte durchgeführt:

1. Die partitionierte Netzwerkgleichung (2.10) wurde mithilfe des Transformations-Lemmas 2.1 in das äquivalente gekoppelte Hessenbergsystem (2.11) überführt.
2. Für das gekoppelte Hessenbergsystem (2.11) wurde das nicht-transformierte Multirate-Verfahren (2.18) entwickelt.

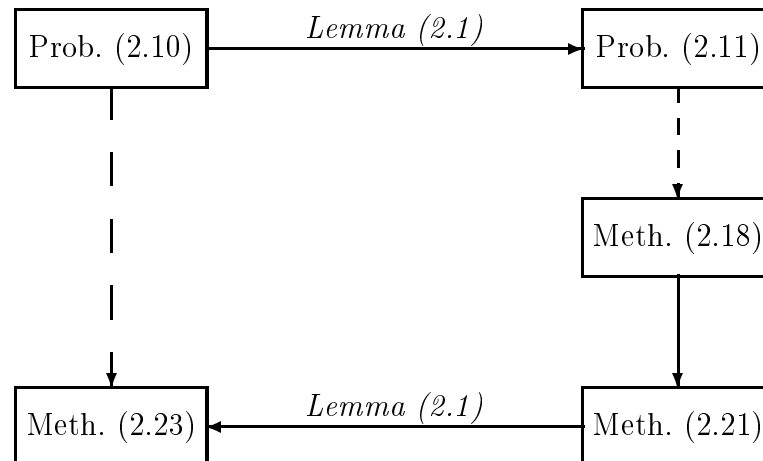


Abbildung 3.1: Schematische Verfahrensherleitung

3. Das nicht-transformierte Multirate-Verfahren (2.18) ging mittels linearer Transformationen in das Multirate-Verfahren (2.21) für das gekoppelte Hessenbergsystem über.
4. Das Multirate-Verfahren (2.23) für das Ausgangsproblem der partitionierten Netzwerkgleichung (2.10) erhielten wir durch eine weitere lineare Transformation aus dem Multirate-Verfahren (2.21).

Abbildung (3.1) verdeutlicht diese Abhängigkeiten in anschaulicher Weise. Zusammen mit der Tatsache, dass ROW-Verfahren invariant bzgl. linearen Transformationen sind, zeigt sie folgendes:

Da die Multirate-Verfahren (2.23), (2.21), (2.18) durch lineare Transformationen miteinander verbunden sind (in der Abbildung durch durchgezogene Linien gekennzeichnet) kann statt dem Verhalten des zu implementierenden Multirate-Verfahrens (2.23) für die partitionierte Netzwerkgleichung (2.10) das Verhalten des nichttransformierten Multirate-Verfahrens (2.18) in Bezug auf das gekoppelte Hessenbergsystem (2.11) untersucht werden.

Koeffizienten, die anhand dieser Betrachtung gefunden werden, können wir durch die Transformationen (2.19) in Verfahrenskoeffizienten für das Multirate-Verfahren (2.23) der partitionierten Netzwerkgleichung verwandeln.

Erinnern wir uns an die Entwicklung des nichttransformierten Multirate-Verfahrens (2.18). Das eigentlich bestimmende Problem war das Teilproblem (2.15a) des gekoppelten Hessenbergsystems (2.11). Dieses sei an dieser Stelle nochmals aufgeführt¹:

¹In diesem Kapitel wird die hier neu definierte Formelnummer (3.1) benutzt.

$\begin{aligned} \dot{y}_l &= w_l(x_l, x_a) \\ 0 &= y_l - g_l(x_l) \end{aligned}$	$\begin{aligned} \dot{y}_a &= w_a(x_l, x_a) \\ 0 &= y_a - g_a(x_l) \end{aligned}$	(3.1)
---	---	-------

3.1 Konsistenzordnung und Grundannahmen

Um die **Konsistenz** des Multirate-Verfahrens (2.18) untersuchen zu können, muss zunächst geklärt werden, was Konsistenz im Falle eines Multirate-Verfahrens für Netzwerkgleichungen bedeutet. Anschließend werden noch einige analytische Forderungen an das gekoppelte Hessenberg-Teilproblem (3.1) gestellt, um die Konsistenzanalyse (insb. die Taylorentwicklung) im mathematischen Sinne „sauber“ durchführen zu können.

Von zentraler Bedeutung für die Eigenschaften eines numerischen Integrationsverfahrens ist der **lokale Diskretisierungsfehler**, der Fehler zwischen exakter und approximierter Lösung nach einem Zeitschritt. Ein Verfahren heißt **konsistent**, wenn dieser Fehler mit feiner werdender Unterteilung der Zeitschicht gegen Null strebt. Die „Geschwindigkeit“, mit der dies geschieht, definiert die **Konsistenzordnung** des Verfahrens. Da ein Multirate-Verfahren mit verschiedenen Zeitdiskretisierungen arbeitet, muss sich dies auch im Konsistenzbegriff niederschlagen.

In der Simulation elektrischer Schaltungen interessiert man sich für die Spannungen (in unserem Fall x_l, x_a). Deshalb definiert man die Konsistenz nur über diese und lässt die anderen Unbekannten (die Ströme y_l, y_a , bzw. z_l, z_a) außer Acht:

Definition 3.1 (Konsistenzordnung in der Schaltungssimulation)

*Ein Multirate-Verfahren² in der Schaltungssimulation besitzt die Konsistenzordnung p , falls gilt:
im compound-Schritt:*

$$\begin{aligned} \|x_l(t_0 + h_l) - x_{l,1}\| &= \mathcal{O}(h_l^{p+1}), \\ \|x_a(t_0 + h_a) - x_{a,1}\| &= \mathcal{O}(h_a^{p+1}) \end{aligned}$$

und im η -ten späteren Mikroschritt $\eta = 1, \dots, \mu$:

$$\|x_a((t_0 + h_a + \sum_{\nu=1}^{\eta-1} h_{mc,\nu}) + h_{mc,\eta}) - x_{a,\eta}\| = \mathcal{O}(h_{mc,\eta}^{p+1})$$

Im letzteren Fall wird in jedem Mikroschritt nicht mit den Werten von eventuell vorhergehenden Mikroschritten als Startwert gerechnet, sondern es wird von jeweils exakten Anfangswerten ausgegangen.

²mit einer Zweiteilung der Netzwerkgleichungen in aktive und latente Komponenten, sowie compound-Schritte und spätere Mikroschritte

Bem.: Es wäre durchaus denkbar, hier mit verschiedenen Konsistenzordnungen p_1, p_2, p_3 zu arbeiten, was jedoch der Einfachheit halber im Weiteren vermieden wird.

Durch die Wahl geeigneter Verfahrenskoeffizienten wird die Konsistenzordnung den jeweiligen Bedürfnissen angepasst. Dabei bleibt immer zu bedenken, dass mit steigenden Genauigkeitsforderungen die Stufenzahl s des Verfahrens steigen wird.

Um entsprechende **Bedingungsgleichungen** für die Verfahrenskoeffizienten herleiten zu können, muss (3.1) einige analytische Forderungen erfüllen:

- $g_l(x_l), g_a(x_a)$ seien in einer Umgebung der Lösung genügend oft **stetig differenzierbar** nach x_l bzw. x_a .

Auf die Simulation elektrischer Schaltungen übertragen bedeutet dies, dass die ladungsspeichernden Elemente genügend glatt modelliert werden müssen.

- $\frac{\partial g_l}{\partial x_l}, \frac{\partial g_a}{\partial x_a}$ seien in einer Umgebung der Lösung **regulär**

Dies ist gerade die Index-1 Hypothese.

- die partiellen Ableitungen der Form $\frac{\partial^{\mu+n_u} w_\lambda}{\partial x_l^\mu \partial x_a^\nu}$ ($\lambda = l, a$) **existieren und seien stetig** für genügend große μ, ν
- w_l, w_a erfüllen in einer Umgebung der Lösung in jeweils beiden Koeffizienten eine **Lipschitzbedingung**:

$$\begin{aligned} \|w_\lambda(u_l, x_a) - w_\lambda(v_l, x_a)\| &\leq L_1^\lambda \cdot \|u_l - v_l\| \\ \|w_\lambda(x_l, u_a) - w_\lambda(x_l, v_a)\| &\leq L_2^\lambda \cdot \|u_a - v_a\| \end{aligned}$$

Das Erfüllen einer Lipschitzbedingung ist zunächst eine rein analytische Bedingung, wie sie im Existenz- und Eindeutigkeitssatz von Picard-Lindelöf für ODEs von Nöten ist. Eine Lipschitzbedingung in der jeweils eingekoppelten Größe ist darüberhinaus eine natürliche Bedingung, denkt man an die angesprochene schwache Einkopplung der jeweils anderen Aktivitätsschicht.

- w_l, w_a seien in einer Umgebung der Lösung beschränkt.

In Bezug auf die elektrische Netzwerkanalyse ist auch dies eine natürliche Bedingung, garantiert dies doch letztlich eine Beschränkung des Stromflusses, was der Energieerhaltung entspricht.

3.2 Taylorentwicklung für partitionierte Index-1-DAEs

Das Teilsystem (3.1) des gekoppelten Hessenbergsystems stellt, für sich genommen, bereits ein gekoppeltes Index-1-System in Hessenbergform dar. Wir werden

nun die **Taylorentwicklung** der zugehörigen **Lösungskomponenten** $x_l(t_0 + h_l), x_a(t_0 + h_a)$ um den Entwicklungspunkt $x_l(t_0)$ bzw. $x_a(t_0)$ anstreben.

Wie leicht zu sehen ist, gehen in die auftretenden Differentiale Ableitungen der vier Funktionen g_l, g_a, w_l, w_a ein. Daher ist vorauszusehen, dass sie entsprechend kompliziert ausfallen werden. Wir werden ein „Meer an Indizes und Differentia- len“ umgehen, indem wir die Idee der „graphischen“ Differentiation mithilfe von Butcher-Bäumen vom ODE-Fall auf vorliegendes Problem (3.1) übertragen.

Um die Indizes nicht unkontrolliert wuchern zu lassen, seien höhere Ableitungen durch **Multilinearformen** ausgedrückt, wie sie die folgende Definition beschreibt (siehe auch [HW96, VI.4])

Definition 3.2 *Für eine Funktion*

$$\begin{aligned} f &: \mathbb{R}_1^n \times \dots \times \mathbb{R}_q^n \rightarrow \mathbb{R}^m \\ (x_1, \dots, x_q) &\mapsto f(x_1, \dots, x_q) \end{aligned}$$

und Vektoren

$$u_1 \in \mathbb{R}_1^n, \dots, u_q \in \mathbb{R}_q^n$$

bezeichne (die jeweilige Existenz der partiellen Ableitungen vorausgesetzt)

$$\frac{\partial^k f}{\partial x_{\mu_1} \dots \partial x_{\mu_k}}(u_{\mu_1}, \dots, u_{\mu_k})$$

die höheren Ableitungen

$$\left(\sum_{j_1=1}^{n_{\mu_1}} \dots \sum_{j_k=1}^{n_{\mu_k}} \frac{\partial^k f_i}{\partial x_{\mu_1, j_1} \dots \partial x_{\mu_k, j_k}} \cdot u_{\mu_1, j_1} \dots u_{\mu_k, j_k} \right)_{i=1, \dots, m}$$

Im Kontext der Taylorentwicklung werden die jeweiligen Differentiale am Entwicklungspunkt ausgewertet.

Bem.: Als weitere Vereinfachung seien alle im Weiteren auftretenden Funktionsauswertungen jeweils an der Entwicklungsstelle $(x_{l,0}, x_{a,0})$ vorgenommen.

Taylorentwicklung der exakten Lösung

Bevor wir auf eine „graphische“ Differentiation eingehen, wollen wir einige Differentiale „zu Fuß“ berechnen.

Durch Ableiten der algebraischen Bedingung von (3.1) erhält man oBdA. für den latenten Teil

$$\dot{x}_l = \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} w_l(x_l, x_a). \quad (3.2)$$

Eine weitere zeitliche Differentiation liefert

$$\ddot{x}_l = \frac{d}{dt} \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \cdot w_l(x_l, x_a) + \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \cdot \frac{d}{dt} w_l(x_l, x_a).$$

Wobei für einen konstanten Vektor u gilt:

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \cdot u &= - \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \cdot \frac{\partial^2 g_l}{\partial x_l \partial x_l} (\dot{x}_l, \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \cdot u) \\ &= \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \cdot \frac{\partial^2 g_l}{\partial x_l \partial x_l} \left(\left(\frac{\partial g_l}{\partial x_l} \right)^{-1} w_l, \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \cdot u \right). \end{aligned}$$

Dies ist eine Konsequenz aus der Kettenregel und der Regel

$$(A^{-1}(x))' = -A^{-1}(x)A'(x)A^{-1}(x).$$

Damit folgt also für die Ableitung 2.Ordnung:

$$\begin{aligned} \ddot{x}_l &= - \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \frac{\partial^2 g_l}{\partial x_l \partial x_l} \left(\left(\frac{\partial g_l}{\partial x_l} \right)^{-1} w_l, \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} w_l \right) \\ &\quad + \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \frac{\partial w_l}{\partial x_l} \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} w_l + \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \frac{\partial w_l}{\partial x_a} \left(\frac{\partial g_a}{\partial x_a} \right)^{-1} w_a \end{aligned} \quad (3.3)$$

Bereits jetzt sieht man, dass diese Ausdrücke sehr kompliziert werden. Eine graphische Repräsentation der einzelnen Terme in (3.2) und (3.3) ist wünschenswert.

Es stellt sich heraus, dass es eine Darstellung dieser Ableitungsterme in Form von **Bäumen** gibt, die einerseits die Differentiale in kompakter Form erscheinen lassen und andererseits ein Vordringen in höhere Differentiationsebenen durch einfachen „Ausbau“ der Bäume ermöglicht.

Bäume und elementare Differentiale

Ähnlich zu den Butcher-Bäumen für gewöhnliche Differentialgleichungen ([HNW00, II.2]) und den von Hairer und Wanner beschriebenen differential-algebraischen Bäumen ([HW96, VI.4]) lassen sich die im Multirate-DAE-Problem (3.1) auftauchenden elementaren Differentiale (also die Terme aus beispielsweise (3.3)) durch entsprechend angepasste Bäume beschreiben.

Da wir es hier mit vier verschiedenen Funktionen (g_l, w_l, g_a, w_a) zu tun haben, benötigen wir vier verschiedenartige Baumknoten. Die **latenten Anteile** seien durch **runde** und die **aktiven** durch **quadratische Knoten** identifiziert.

Leere Knoten deuten auf eine Beteiligung der Funktionen g_l, g_a , **gefüllte Knoten** auf die Funktionen w_l, w_a hin. Die **Äste** der Bäume bestimmen die auftretenden partiellen Ableitungen, je nachdem welche Knoten sie verbinden

Als **Wurzel** sei der jeweils unterste Knoten eines Baumes, also der Knoten, der nicht Endpunkt eines Astes ist, bezeichnet.

Folgende Definition ergibt eine rekursive Beschreibung der Bäume

Definition 3.3 (Multirate-Differential-Algebraic-Trees (MDAT)) *Es sei $\text{MDAT} = \text{MDAT}_{y_l} \cup \text{MDAT}_{y_a} \cup \text{MDAT}_{x_l} \cup \text{MDAT}_{x_a}$ die Menge aller (Multirate-DAE) Bäume.*

Die Indizes y_l, y_a, x_l, x_a bei den obigen Teilmengen und bei den weiteren Bäumen bezeichnen die Art der jeweiligen Wurzel:

y_l	gefüllter Kreis
x_l	leerer Kreis
y_a	gefülltes Quadrat
x_a	leeres Quadrat

Alle möglichen Bäume ergeben sich durch folgende rekursive Beschreibung:

$$\begin{aligned} a) \quad \bullet &= \tau_{y_l} \in \text{MDAT}_{y_l}, & \circ &= \tau_{x_l} \in \text{MDAT}_{x_l} \\ & & \blacksquare &= \tau_{y_a} \in \text{MDAT}_{y_a}, & \square &= \tau_{x_a} \in \text{MDAT}_{x_a} \end{aligned}$$

$$\begin{aligned} b) \quad [u_1^L, \dots, u_m^L, u_1^A, \dots, u_n^A]_{y_l} &\in \text{MDAT}_{y_l} \\ \text{falls} \quad u_1^L, \dots, u_m^L &\in \text{MDAT}_{x_l} \\ u_1^A, \dots, u_n^A &\in \text{MDAT}_{x_a} \\ n, m \in \mathbb{N}_0, (m, n) &\neq (0, 0) \end{aligned}$$

$$\begin{aligned} c) \quad [u_1^A, \dots, u_m^A, u_1^L, \dots, u_n^L]_{y_a} &\in \text{MDAT}_{y_a} \\ \text{falls} \quad u_1^A, \dots, u_m^A &\in \text{MDAT}_{x_a} \\ u_1^L, \dots, u_n^L &\in \text{MDAT}_{x_l} \\ n, m \in \mathbb{N}_0, (m, n) &\neq (0, 0) \end{aligned}$$

$$\begin{aligned} d) \quad [t_1^L]_{x_l} &\in \text{MDAT}_{x_l} \\ \text{falls} \quad t_1^L &\in \text{MDAT}_{y_l} \end{aligned}$$

$$\begin{aligned} e) \quad [u_1^L, \dots, u_n^L]_{x_l} &\in \text{MDAT}_{x_l} \\ \text{falls} \quad u_1^L, \dots, u_n^L &\in \text{MDAT}_{x_l} \\ n \in \mathbb{N}, n &> 1 \end{aligned}$$

$$\begin{aligned} f) \quad [t_1^A]_{x_a} &\in \text{MDAT}_{x_a} \\ \text{falls} \quad t_1^A &\in \text{MDAT}_{y_a} \end{aligned}$$

$$\begin{aligned} g) \quad [u_1^A, \dots, u_n^A]_{x_a} &\in \text{MDAT}_{x_a} \\ \text{falls} \quad u_1^A, \dots, u_n^A &\in \text{MDAT}_{x_a} \\ n \in \mathbb{N}, n &> 1 \end{aligned}$$

Die Bäume in den eckigen Klammern „[]“ stehen in keiner geordneten Reihenfolge

Zusammen mit einem Index x_l, x_a, y_l, y_a beschreibt der Klammer-Operator „[]“ die Bildung eines neuen Baumes: Die Bäume innerhalb der Klammern werden über einen gemeinsamen Knoten, der neuen Wurzel miteinander verbunden. Der Index bestimmt die Art dieser neuen Wurzel (d. h. Kreis oder Quadrat, leer oder voll).

$$\begin{aligned} \text{z. B. } [\text{○} \text{---} \text{●}, \text{○} \text{---} \text{●}]_{x_l} &= \text{○} \text{---} \text{○} \text{---} \text{○} \text{---} \text{○} \text{---} \text{○} \in \text{MDAT}_{x_l}, \\ [\text{□} \text{---} \text{■}, \text{□} \text{---} \text{■}]_{x_a} &= [\text{□} \text{---} \text{■}, [\text{□} \text{---} \text{■}]_{x_a}]_{x_a} = [\text{□} \text{---} \text{■}, [\text{○} \text{---} \text{●}]_{y_a}]_{x_a} \in \text{MDAT}_{x_a} \end{aligned}$$

Da die MDAT-Bäume eine graphische Repräsentation von Differentialen verschiedener Ordnung sind, ist es natürlich, auch für sie einen Ordnungsbegriff einzuführen:

Definition 3.4 (Ordnung) Für $t \in \text{MDAT}$ ist die Ordnung $\varrho(t)$ die Anzahl der vollen Knoten (unabhängig von ihrer Form)

$$\text{z. B. } \varrho \left(\begin{array}{c} \text{●} \text{---} \text{■} \\ \text{○} \text{---} \text{○} \text{---} \text{○} \end{array} \right) = 3$$

Die rekursive Definition der MDAT-Bäume (3.3) ermöglicht nun eine einfache Zuordnung Baum $\leftrightarrow$ Differential:

Definition 3.5 (Vom Baum zum Differential) Aus den MDAT-Bäumen ergeben sich die zugehörigen Differentiale auf folgende Art und Weise:

- a) $F(\tau_{y_l}) = w_l, \quad F(\tau_{x_l}) = \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} w_l$
 $F(\tau_{y_a}) = w_a, \quad F(\tau_{x_a}) = \left(\frac{\partial g_a}{\partial x_a} \right)^{-1} w_a$
- b) $F(t^L) = \frac{\partial^{m+n} w_l}{\partial x_l^m \partial x_a^n} \left(F(u_1^L), \dots, F(u_m^L), F(u_1^A), \dots, F(u_n^A) \right)$
falls $t^L = [u_1^L, \dots, u_m^L, u_1^A, \dots, u_n^A]_{y_l} \in \text{MDAT}_{y_l}$
- c) $F(t^A) = \frac{\partial^{m+n} w_a}{\partial x_l^m \partial x_a^n} \left(F(u_1^L), \dots, F(u_n^L), F(u_1^A), \dots, F(u_m^A) \right)$
falls $t^A = [u_1^A, \dots, u_m^A, u_1^L, \dots, u_n^L]_{y_a} \in \text{MDAT}_{y_a}$
- d) $F(u^L) = \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} F(t_1^L)$
falls $u^L = [t_1^L]_{x_l} \in \text{MDAT}_{x_l}$

$$e) \quad F(u^L) = - \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \frac{\partial^n g_l}{\partial x_l^n} \left(F(u_1^L), \dots, F(u_n^L) \right)$$

$$\text{falls } u^L = [u_1^L, \dots, u_n^L]_{x_l} \in \text{MDAT}_{x_l}$$

$$f) \quad F(u^A) = \left(\frac{\partial g_a}{\partial x_a} \right)^{-1} F(t_1^A)$$

$$\text{falls } u^A = [t_1^A]_{x_a} \in \text{MDAT}_{x_a}$$

$$g) \quad F(u^A) = - \left(\frac{\partial g_a}{\partial x_a} \right)^{-1} \frac{\partial^n g_a}{\partial x_a^n} \left(F(u_1^A), \dots, F(u_n^A) \right)$$

$$\text{falls } u^A = [u_1^A, \dots, u_n^A]_{x_a} \in \text{MDAT}_{x_a}$$

Die Zugehörigkeit der Teilbäume $u_i^L, u_i^A, t_i^L, t_i^A$ zu den Teilmengen von MDAT besteht in analoger Weise zur Definition 3.3.

Da nach dem Satz von Schwarz die Reihenfolge der partiellen Differentiationen nicht relevant ist, können in obiger Definition die Teilbäume $u_i^L, u_i^A, t_i^L, t_i^A$ beliebig permutiert werden, ohne dass dies Auswirkungen auf die Differentiale $F(\cdot)$ hätte, d. h. die Differentiale sind wohl definiert.

Das beschriebene Vorgehen, aus Bäumen Differentiale zu gewinnen, sei an einem kleinen Beispiel erläutert:

$$F(\text{Diagram}) = F([\text{Diagram}],_{x_l}) = - \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} (F(\tau_{x_l}), F(\tau_{x_l}))$$

$$= - \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \frac{\partial^2 g_l}{\partial x_l \partial x_l} \left(\left(\frac{\partial g_l}{\partial x_l} \right)^{-1} w_l, \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} w_l \right)$$

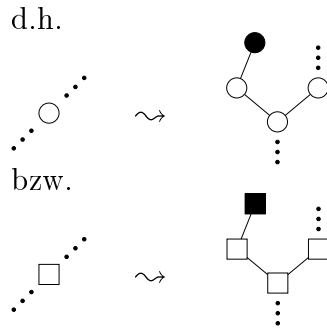
Dies ist gerade der erste Term aus (3.3).

Nun lassen sich nicht nur Bäume auf Differentiale und Differentiale auf Bäume abbilden. Es ist auch möglich, die sukzessive Ableitung direkt am graphischen Objekt des Baumes durchzuführen.

Vom Baum zum Baum: Graphische Differentiation

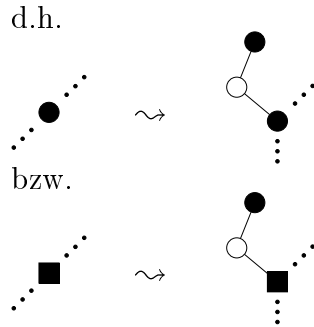
Wendet man auf jeweils einen Knoten eines Baumes $t \in \text{MDAT}$ jeweils eine der folgenden Regeln an, so erhält man eine Menge von Bäumen, deren aufaddierte zugehörige Differentiale die Ableitung des Differentials $F(t)$ ergeben.

1. Spalte leere Knoten auf in zwei gleichartige leere Knoten und hänge an den unteren der beiden, der Art des Knoten entsprechend, τ_{x_l} bzw. τ_{x_a} an.



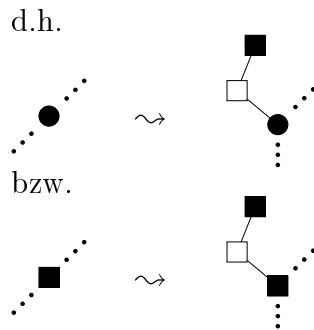
Dies entspricht der Differentiation von $\left(\frac{\partial g_l}{\partial x_l}\right)^{-1} v$ nach x_l und dem Hinzufügen des Terms $\dot{x}_l = \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l$, bzw. von $\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} v$ nach x_a und dem Hinzufügen des Terms $\dot{x}_a = \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a$ (v : konstanter Vektor).

2. Hänge an volle Knoten τ_{x_l} an.



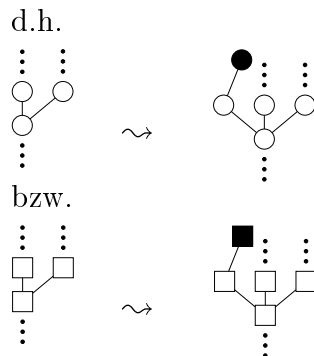
Dies entspricht der Differentiation von w_l nach x_l und dem Hinzufügen des Terms $\dot{x}_l = \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l$, bzw. von w_a nach x_l und dem Hinzufügen des Terms $\dot{x}_l = \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l$.

3. Hänge an volle Knoten τ_{x_a} an.



Dies entspricht der Differentiation von w_l nach x_a und dem Hinzufügen des Terms $\dot{x}_a = \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a$, bzw. von w_a nach x_a und Hinzufügen des Terms $\dot{x}_a = \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a$.

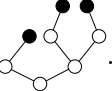
4. Falls an einem leeren Knoten zwei oder mehrere Bäume mit gleichartiger leerer Wurzel anschließen, so hänge an diesen Knoten τ_{x_l} bzw. τ_{x_a} , der Form des Knoten entsprechend, an. (Im Bild ist der Fall aufgezeigt, an dem zwei Bäume mit gleicher Wurzel anschliessen.)



Dies entspricht der Differentiation von $\frac{\partial^n g_l}{\partial x_l^n}(v_1, \dots, v_n)$ nach x_l und dem Hinzufügen des Terms $\dot{x}_l = \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l$, bzw. von $\frac{\partial^n g_a}{\partial x_a^n}(v_1, \dots, v_n)$ nach x_a und dem Hinzufügen des Terms $\dot{x}_a = \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a$ $v_1, \dots, v_n$: konstante Vektoren, $n > 1$.

Bei genauer Betrachtung dieser Regeln fallen mehrere Eigenschaften auf:

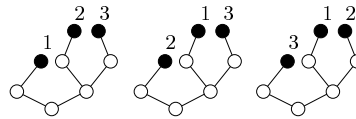
- Jede Anwendung einer der vier Regeln führt dazu, dass genau ein voller Knoten hinzukommt, die Ordnung des resultierenden Baumes also um jeweils eins höher liegt als die des Ursprungsbaumes. Übertragen auf die Ableitung der Differentiale bedeutet dies, dass die Ordnung des Differentials von Stufe zu Stufe wächst.
- Die obersten Knoten eines Baumes sind jeweils volle Knoten, d.h. jedes Differential endet mit einer Funktionsauswertung von w_l oder w_a .
- Aus einem Ursprungsbaum können gleichaussehende Bäume entstehen.
z.B. ergibt die Anwendung der Regel 4) auf die drei leeren Knoten des

Baumes  jeweils den Baum .

Um diesem mehrfachen Auftreten von Bäumen Rechnung zu tragen, bietet sich an, die vollen Knoten zu numerieren um damit Bäume unterscheidbar zu machen.

Hierbei ist zu beachten, dass volle Knoten in unverzweigten Teilbäumen von „unten nach oben“ entstehen. Somit macht eine Numerierung nur Sinn, wenn sie diese Entwicklung berücksichtigt, die vollen Knoten also in jedem Teilbaum **monoton aufsteigend beziffert** werden. So wie bei unbeschrifteten MDAT-Bäumen eventuelle Permutationen von Teilbäumen keinen Einfluss auf den Gesamtbaum und damit auf die zugeordneten Differentiale haben, darf dies auch bei der Numerierung keine Rolle spielen.

So gibt es beispielsweise für den obigen Baum drei verschiedene, monoton aufsteigende Beschriftungen, also genau so viele Numerierungen, wie verschiedene Baumentwicklungen vorliegen:



Definition 3.6 (Monoton beschriftete Bäume (LMDAT))

Ein Baum $t \in \text{MDAT}$, versehen mit einer monoton aufsteigenden Numerierung seiner vollen Knoten nach obigem Muster, wird **monoton beschrifteter Baum** genannt. Die Menge LMDAT (labelled multirate-differential-algebraic-trees) aller dieser Bäume, ergibt sich aus der Menge aller möglichen monotonen Beschrifteten aller möglichen MDAT-Elemente.

Entsprechendes gilt für die Teilmengen $\text{LMDAT} = \text{LMDAT}_{x_l} \cup \text{LMDAT}_{x_a} \cup \text{LMDAT}_{y_l} \cup \text{LMDAT}_{y_a}$

Wir sind natürlich daran interessiert, die Anzahl der verschiedenen Numerierungen eines Baumes mit einer allgemeingültigen Vorschrift berechnen zu können. Könnten wir dies nicht, so müssten wir uns für jeden Baum die Anzahl seiner Numerierungsmöglichkeiten überlegen.

Eine Möglichkeit, eine solche Vorschrift zu erhalten, bietet die Betrachtung von MDAT als Repräsentantensystem einer Äquivalenzrelation auf LMDAT. Ist diese gegeben durch:

$t_1, t_2 \in \text{LMDAT}, t_1 \sim t_2 \Leftrightarrow t_1$ *stimmt bis auf die Beschriftung überein mit* t_2 , dann ist MDAT in der Tat ein Repräsentantensystem, und die Anzahl der Numerierungsmöglichkeiten eines Baumes $t \in \text{MDAT}$ ist gerade die Mächtigkeit seiner zugehörigen Äquivalenzklasse in LMDAT.

Lemma 3.7 *Die Mächtigkeit der Äquivalenzklasse des Baumes $\tilde{t} \in \text{LMDAT}$, also die Anzahl der Möglichkeiten $\alpha(t)$, den Baum $t \in \text{MDAT}$, der aus $\tilde{t}$ durch Weglassen der Numerierung entsteht, monoton zu beschriften ergibt sich zu:*

$$\begin{aligned}
 a) \quad & \alpha(\tau_{y_l}) = 1, \quad \alpha(\tau_{x_l}) = 1 \\
 & \alpha(\tau_{y_a}) = 1, \quad \alpha(\tau_{x_a}) = 1 \\
 b) \quad & \alpha(t) = \left(\begin{array}{c} \varrho(t) - 1 \\ \varrho(u_1^L), \dots, \varrho(u_m^A) \end{array} \right) \cdot \alpha(u_1^L) \cdots \alpha(u_m^A) \cdot \frac{1}{\mu_1! \mu_2! \cdots} \\
 & \text{falls } t = [u_1^L, \dots, u_m^L, u_1^A, \dots, u_n^A]_{y_l} \in \text{MDAT}_{y_l} \\
 & \text{oder } t = [u_1^L, \dots, u_m^L, u_1^A, \dots, u_n^A]_{y_a} \in \text{MDAT}_{y_a} \\
 c) \quad & \alpha(u) = \alpha(t) \\
 & \text{falls } u = [t]_{x_l} \in \text{MDAT}_{x_l}, t \in \text{MDAT}_{y_l} \\
 & \text{oder } u = [t]_{x_a} \in \text{MDAT}_{x_a}, t \in \text{MDAT}_{y_a} \\
 d) \quad & \alpha(t) = \left(\begin{array}{c} \varrho(u) \\ \varrho(u_1), \dots, \varrho(u_n) \end{array} \right) \cdot \alpha(u_1) \cdots \alpha(u_n) \cdot \frac{1}{\mu_1! \mu_2! \cdots} \\
 & \text{falls } u = [u_1, \dots, u_n]_{x_l} \in \text{MDAT}_{x_l} \\
 & \quad u_1, \dots, u_n \in \text{MDAT}_{x_l} \\
 & \text{oder } u = [u_1, \dots, u_n]_{x_a} \in \text{MDAT}_{x_a} \\
 & \quad u_1, \dots, u_n \in \text{MDAT}_{x_a}
 \end{aligned}$$

Die Werte μ_i bezeichnet hierbei die Größe eventuell vorkommender Gruppen, (bis auf Permutationen) gleicher Bäume unter den jeweiligen Mengen der einen größeren Baum bildenden Teilbäume.

Beweis 3.8 *Siehe den Beweis zu Lemma 8 in [H81] mit der Zusatzüberlegung für die Teile c) und d), dass leere Knoten nicht numeriert werden, Bäume mit einem solchen Knoten als Wurzel keine beschriftete Wurzel besitzen.*

Als Beispiel ergibt sich also für den bereits mehrfach zitierten Baum:

$$\begin{aligned}
 \alpha(\text{Baum}) &= \binom{3}{1,2} \alpha(\text{Baum}_1) \cdot \alpha(\text{Baum}_2) \cdot \frac{1}{1!1!} \\
 &= \binom{3}{1,2} \cdot 1 \cdot \binom{2}{1,1} \cdot \alpha(\text{Baum}_1) \cdot \alpha(\text{Baum}_2) \cdot \frac{1}{2!} \\
 &= \frac{3!}{1 \cdot 2} \cdot \frac{2!}{1 \cdot 1} \cdot \frac{1}{2!} = 3
 \end{aligned}$$

Überträgt man nun die vorangegangenen Definitionen, die die Beziehungen vom umbeschrifteten Baum zum Differential beschreiben, und die Regeln der „graphischen“ Differentiation auf die beschrifteten Bäume LMDAT, dann ist es ein Leichtes, die Ableitungen der exakten Lösung x_l, x_a kompakt zu beschreiben:

Theorem 3.9 (Ableitungen der exakten Lösung) *Für die Ableitungen der exakten Lösung x_l, x_a von Problem (3.1) gilt:*

$$\begin{aligned}
 x_l^{(q)}(t_0) &= \sum_{u^L \in \text{LMDAT}_{x_l, \varrho(u^L)=q}} F(u^L)(x_{l,0}, x_{a,0}) = \sum_{u^L \in \text{MDAT}_{x_l, \varrho(u^L)=q}} \alpha(u^L) \cdot F(u^L)(x_{l,0}, x_{a,0}) \\
 x_a^{(q)}(t_0) &= \sum_{u^A \in \text{LMDAT}_{x_a, \varrho(u^A)=q}} F(u^A)(x_{l,0}, x_{a,0}) = \sum_{u^A \in \text{MDAT}_{x_a, \varrho(u^A)=q}} \alpha(u^A) \cdot F(u^A)(x_{l,0}, x_{a,0})
 \end{aligned}$$

Beweis 3.10 *Die beschriebene graphische Differentiation mit Hilfe der LMDAT-Bäume erzeugt, ausgehend von τ_{x_l} bzw. τ_{x_a} (in beschrifteter Form) jeden Baum höherer Ordnung genau einmal. So werden nach q Schritten alle Bäume der Ordnung q genau einmal erzeugt. Nimmt man nun statt der beschrifteten die umbeschrifteten Bäume, so muss bei jedem nun auftretenden Baum gezählt werden, auf wieviele Arten man dessen volle Knoten monoton numerieren kann, wie oft dieser Baum und damit dessen Differential F also auftaucht.*

Tabelle 3.1 zeigt alle Bäume und zugehörigen Differentiale der Ordnungen 1 und 2. Eine Aufstellung aller Differentiale bis zur Ordnung 3 finden sich im Anhang.

Mit Theorem (3.9) folgt nun abschliessend:

Lemma 3.11 (Taylorentwicklung der exakten Lösung) *Für die Taylorentwicklungen der Lösungen x_l, x_a von (3.1) um den Entwicklungspunkt t_0 gilt:*

$$x_l(t_0 + h_l) = \sum_{\substack{u^L \in \text{MDAT} \\ \varrho(u^L) \leq q}} \alpha(u^L) \cdot F(u^L)(x_{l,0}, x_{a,0}) \cdot \frac{h_l^{\varrho(u^L)}}{\varrho(u^L)!} + R_q^L(h_l), \quad (3.4a)$$

$$x_a(t_0 + h_a) = \sum_{\substack{u^A \in \text{MDAT} \\ \varrho(u^A) \leq q}} \alpha(u^A) \cdot F(u^A)(x_{l,0}, x_{a,0}) \cdot \frac{h_a^{\varrho(u^A)}}{\varrho(u^A)!} + R_q^A(h_a), \quad (3.4b)$$

mit Resten $\|R_q^L(h_l)\| = \mathcal{O}(h_l^{q+1})$ und $\|R_q^A(h_a)\| = \mathcal{O}(h_a^{q+1})$.

u_{x_l}	$\alpha(\cdot)$	$F(u_{x_l})$	u_{x_a}	$\alpha(\cdot)$	$F(u_{x_a})$
	1	$\left(\frac{\partial q_l}{\partial x_l}\right)^{-1} w_l$		1	$\left(\frac{\partial q_a}{\partial x_a}\right)^{-1} w_a$
	1	$-\left(\frac{\partial q_l}{\partial x_l}\right)^{-1} \frac{\partial^2 q_l}{\partial x_l \partial x_l} \left(\left(\frac{\partial q_l}{\partial x_l}\right)^{-1} w_l, \left(\frac{\partial q_l}{\partial x_l}\right)^{-1} w_l\right)$		1	$-\left(\frac{\partial q_a}{\partial x_a}\right)^{-1} \frac{\partial^2 q_a}{\partial x_a \partial x_a} \left(\left(\frac{\partial q_a}{\partial x_a}\right)^{-1} w_a, \left(\frac{\partial q_a}{\partial x_a}\right)^{-1} w_a\right)$
	1	$\left(\frac{\partial q_l}{\partial x_l}\right)^{-1} \frac{\partial w_l}{\partial x_l} \left(\frac{\partial q_l}{\partial x_l}\right)^{-1} w_l$		1	$\left(\frac{\partial q_a}{\partial x_a}\right)^{-1} \frac{\partial w_a}{\partial x_a} \left(\frac{\partial q_a}{\partial x_a}\right)^{-1} w_a$
	1	$\left(\frac{\partial q_l}{\partial x_l}\right)^{-1} \frac{\partial w_l}{\partial x_a} \left(\frac{\partial q_a}{\partial x_a}\right)^{-1} w_a$		1	$\left(\frac{\partial q_a}{\partial x_a}\right)^{-1} \frac{\partial w_a}{\partial x_l} \left(\frac{\partial q_l}{\partial x_l}\right)^{-1} w_l$

Tabelle 3.1: Elementare Differentiale, Ordnung 1 und 2

3.3 Konsistenz im compound-Schritt

Im letzten Abschnitt haben wir die Taylorentwicklung der Lösungskomponenten x_l, x_a durchgeführt (Lemma 3.11). Die MDAT-Bäume, die dort zur kompakten Formulierung der Taylorreihen führten, können wir auch in der **Taylorentwicklung der numerischen Approximationen** $x_{l,0}, x_{a,0}$ im compound-Schritt einsetzen.

Ist dafür eine geschlossene Darstellung gefunden, ergeben sich die Bedingungsgleichungen an die Verfahrenskoeffizienten durch einfachen Taylorreihenabgleich von jeweils exakter Lösung und numerischer Approximation.

Den Weg zu einer entsprechenden Darstellung der Taylorentwicklung der numerischen Approximation werden wir wiederum anhand des latenten Teils beschreiben.

Nach dem nichttransformierten Multirate-Verfahren für das Hessenbergsystem (3.1) ist eine Näherung $x_{l,1} \approx x_l(t_0 + h_l)$ gegeben durch:

$$\begin{aligned}
 x_{l,1} &= x_{l,0} + \sum_{i=1}^s b_i^L \cdot k_i^L \\
 \text{mit } 0 &= g_l(x_{l,0}) - g_l(\tilde{a}_i^L) + \sum_{j=1}^i \beta_{ij}^L \cdot l_j^L - \frac{\partial g_l}{\partial x_l} \sum_{j=1}^i \gamma_{ij}^L \cdot k_j^L \\
 l_i^L &= h_l w_l(\tilde{a}_i^L, \tilde{d}_i^L) + h_l \frac{\partial w_l}{\partial x_l} \sum_{j=1}^i \gamma_{ij}^L \cdot k_j^L \\
 \tilde{a}_i^L &= x_{l,0} + \sum_{j=1}^{i-1} \alpha_{ij}^L \cdot k_j^L \\
 \tilde{d}_i^L &= x_{a,0} + m \cdot \sum_{j=1}^{i-1} \delta_{ij}^L \cdot k_j^A
 \end{aligned} \tag{3.5}$$

$x_{l,1}$ kann also via $k_i^L, l_i^L, k_i^A, l_i^A$ als eine Funktion³ $x_{l,1}(h)$ in h aufgefasst werden, mit $x_{l,1}(0) = x_{l,0} = x_l(t_0)$ und $x_{l,1}(h_l) = x_{l,1} \approx x_l(t_0 + h_l)$. Somit suchen wir nach der Taylorentwicklung

$$x_{l,1} = x_{l,1}(h_l) = \sum_{p=0}^q \frac{h_l^p}{p!} \cdot x_{l,1}^{(p)}(0) \tag{3.6}$$

Die benötigten Ableitungen $x_{l,1}^{(p)}(0)$, $p = 1, \dots, q$ ergeben sich nach (3.5) zu:

$$x_{l,1}^{(p)}(0) = \sum_{i=1}^s b_i^L \cdot (k_i^L)^{(p)}(0). \tag{3.7}$$

³Wir verwenden hier für die Funktion und die Approximation an $x_l(t_0 + h_l)$ dieselbe Bezeichnung. Welches nun jeweils konkret gemeint ist, ergibt sich aus dem Kontext: mit $x_{l,1}(\cdot)$ meinen wir die Funktion, mit $x_{l,1}$ die Approximation.

Hier schreiben wir die Zuwächse und die Zwischenwerte $\tilde{a}_i^L, \tilde{d}_i^L, \tilde{a}_i^A, \tilde{d}_i^A$ als Funktionen in h , wie wir dies für $x_{l,1}$ tun.

Die Aufgabe, die Ableitungen der Zuwächse zu erhalten, bedarf nun einiges an analytischer Arbeit. Wie aus (3.5) ersichtlich, sind höhere Ableitungen von $h \cdot w_l(\tilde{a}_i^L, \tilde{d}_i^L)$ und $g_l(\tilde{a}_i^L)$ an der Stelle $h = 0$ nötig.

Diese lassen sich mithilfe der *Leibnizschen Formel* und des *Lemmas nach Faà di Bruno* (beide siehe [HNW00, II.2]) – Folgerungen aus der *Kettenregel* – bestimmen (alle Funktionsauswertungen finden an der Stelle $h = 0$ statt):

$$\text{i) } (h \cdot w_l(\tilde{a}_i^L, \tilde{d}_i^L))^{(p)} = p \cdot (w_l(\tilde{a}_i^L, \tilde{d}_i^L))^{(p-1)}$$

mit

$$(w_l(\tilde{a}_i^L, \tilde{d}_i^L))^{(p-1)} = \sum_{\substack{\mu_1 + \dots + \mu_l + \\ \nu_1 + \dots + \nu_n \\ = p-1}} \frac{\partial^{l+n} w_l}{\partial x_i^l \partial x_a^n} \left((\tilde{a}_i^L)^{(\mu_1)}, \dots, (\tilde{a}_i^L)^{(\mu_n)}, (\tilde{d}_i^L)^{(\nu_1)}, \dots, (\tilde{d}_i^L)^{(\nu_n)} \right)$$

und

$$(\tilde{a}_i^L)^{(\mu)} = \sum_{j=1}^{i-1} \alpha_{ij}^L (k_j^L)^{(\mu)}, \quad (\tilde{d}_i^L)^{(\nu)} = \left(\frac{1}{m} \right)^\nu \cdot m \cdot \sum_{j=1}^{i-1} \delta_{ij}^L (k_j^A)^{(\nu)} \quad (3.8)$$

Die obige Summe läuft über alle ungeordneten Tupel von Zahlen $\mu_i, \nu_i \in \mathbb{N}_0$, die in der Summe $p - 1$ ergeben.

Der Vorfaktor $(\frac{1}{m})$ in den Ableitungen der Werte $\tilde{d}_i^L$ rührt von der Nachdifferentiation der aktiven Zuwächse $k_i^A = k_i^A(h_a) = k_i^A(\frac{1}{m} \cdot h_l)$ her. (Entsprechend wird im aktiven Fall dieser Vorfaktor m sein).

$$\text{ii) } (g_l(\tilde{a}_i^L))^{(p)} = \sum_{\mu_1 + \dots + \mu_l = p} \frac{\partial^l g_l}{\partial x_i^l} \left((\tilde{a}_i^L)^{(\mu_1)}, \dots, (\tilde{a}_i^L)^{(\mu_n)} \right)$$

Die, die Zuwächse k_i^L, l_i^L bestimmenden Gleichungen von (3.5) p -mal nach h abgeleitet sind also:

$$\begin{aligned} 0 = & - \sum_{\substack{\mu_1 + \dots + \mu_l = p \\ l \geq 2}} \frac{\partial^l g_l}{\partial x_i^l} \left((\tilde{a}_i^L)^{(\mu_1)}, \dots, (\tilde{a}_i^L)^{(\mu_n)} \right) \\ & - \underbrace{\frac{\partial g_l}{\partial x_l} \left(\sum_{j=1}^{i-1} \alpha_{ij}^L \cdot (k_j^L)^{(p)} \right) - \frac{\partial g_l}{\partial x_l} \left(\sum_{j=1}^i \gamma_{ij}^L \cdot (k_j^L)^{(p)} \right) + \sum_{j=1}^i \beta_{ij}^L (l_j^L)^{(p)}}_{= -\frac{\partial g_l}{\partial x_l} \sum_{j=1}^i \beta_{ij}^L \cdot (k_j^L)^{(p)}} \end{aligned} \quad (3.9a)$$

$$\begin{aligned}
(l_i^L)^{(p)} = & \quad p \cdot \sum_{\substack{\mu_1 + \dots + \mu_l + \\ \nu_1 + \dots + \nu_n \\ = p-1 \\ l+n \geq 2}} \frac{\partial^{l+n} w_l}{\partial x_l^l \partial x_a^n} \left((\tilde{a}_i^L)^{(\mu_1)}, \dots, (\tilde{d}_i^L)^{(\nu_n)} \right) \\
& + p \cdot \underbrace{\frac{\partial w_l}{x_l} \sum_{j=1}^{i-1} \alpha_{ij}^L (k_j^L)^{(p-1)} + p \cdot \frac{\partial w_l}{x_l} \sum_{j=1}^i \gamma_{ij}^L (k_j^L)^{(p-1)}}_{p \cdot \frac{\partial w_l}{x_l} \sum_{j=1}^i \beta_{ij}^L (k_j^L)^{(p-1)}} \\
& + q \cdot \left(\frac{1}{m} \right)^{q-1} \cdot m \cdot \frac{\partial w_l}{\partial x_a} \sum_{j=1}^{i-1} \delta_{ij}^L (k_j^A)^{(q-1)}
\end{aligned} \tag{3.9b}$$

Es stellt sich heraus, dass die Einführung der Inversen Ω^λ der Koeffizientenmatrix $\mathcal{B}^\lambda$ ($\lambda = L, A$)

$$\Omega^\lambda = \left(\omega_{ij}^\lambda \right)_{i,j=1}^s \stackrel{\text{def}}{=} \left(\mathcal{B}^\lambda \right)^{-1} \quad (\lambda = L, A)$$

die Elimination der Zuwächse l_i^L ermöglicht.

Aus (3.9a) ergibt sich damit

$$\begin{aligned}
(k_i^L)^{(p)} = & \quad - \sum_{j=1}^i \omega_{ij}^L \sum_{\substack{\mu_1 + \dots + \mu_l = p \\ l \geq 2}} \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \frac{\partial^l g_l}{\partial x_l^l} \left((\tilde{a}_j^L)^{(\mu_1)}, \dots, (\tilde{a}_j^L)^{(\mu_l)} \right) \\
& + \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} (l_i^L)^{(p)}
\end{aligned} \tag{3.10}$$

und zusammen mit (3.9b) folgt daraus für $p \geq 2$

$$\begin{aligned}
(k_i^L)^{(p)} = & \quad - \sum_{j=1}^i \omega_{ij}^L \sum_{\substack{\mu_1 + \dots + \mu_l = p \\ l \geq 2}} \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \frac{\partial^l g_l}{\partial x_l^l} \left((\tilde{a}_j^L)^{(\mu_1)}, \dots, (\tilde{a}_j^L)^{(\mu_l)} \right) \\
& + p \cdot \sum_{\substack{\mu_1 + \dots + \mu_l + \\ \nu_1 + \dots + \nu_n + \\ = p-1 \\ l+n \geq 2}} \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \frac{\partial^l w_l}{\partial x_l^l \partial x_a^n} \left((\tilde{a}_i^L)^{(\mu_1)}, \dots, (\tilde{d}_i^L)^{(\nu_n)} \right) \\
& + p \cdot \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \frac{\partial w_l}{\partial x_l} \sum_{j=1}^i \beta_{ij}^L (k_j^L)^{(p-1)} \\
& + m \cdot \left(\frac{1}{m} \right)^{p-1} p \cdot \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \frac{\partial w_l}{\partial x_a} \sum_{j=1}^i \delta_{ij}^L (k_j^A)^{(p-1)}
\end{aligned} \tag{3.11}$$

In dieser letzten Darstellung (3.11) der Ableitungen erkennt man bereits die Strukturen der elementaren Differentiale aus dem letzten Abschnitt. Wie das folgende Lemma zeigt, sind wir nun in der Lage, die Ableitungen von $k_i^L(h)$ an der Stelle $h = 0$ durch die elementaren Differentiale und damit den Bäumen aus dem letzten Abschnitt zu beschreiben.

Lemma 3.12 (Differentiation der Zuwächse)

Im compound-Schritt gilt für die Differentiation der Zuwächse k_i^L, k_i^A

$$\begin{aligned} \text{latent: } (k_i^L)^{(p)}(0) &= \sum_{\substack{u^L \in \text{MDAT}_{x_l} \\ \varrho(u^L)=p}} \alpha(u^L) \cdot \gamma(u^L) \cdot \Phi_i(u^L) \cdot F(u^L)(x_{l,0}, x_{a,0}) \\ \text{aktiv: } (k_i^A)^{(p)}(0) &= \sum_{\substack{u^A \in \text{MDAT}_{x_a} \\ \varrho(u^A)=p}} \alpha(u^A) \cdot \gamma(u^A) \cdot \Phi_i(u^A) \cdot F(u^A)(x_{l,0}, x_{a,0}) \end{aligned}$$

mit $\gamma(u)$:

- $\gamma(\tau_{x_l}) = \gamma(\tau_{x_a}) = \gamma(\tau_{y_l}) = \gamma(\tau_{y_a}) = 1$
- $\gamma(t) = \varrho(t) \cdot \gamma(u_1^L) \cdot \dots \cdot \gamma(u_l^L) \cdot \gamma(u_1^A) \cdot \dots \cdot \gamma(u_n^A)$
falls $t = [u_1^L, \dots, u_l^L, u_1^A, \dots, u_n^A]_{y_l} \in \text{MDAT}_{y_l}$
bzw. $t = [u_1^L, \dots, u_l^L, u_1^A, \dots, u_n^A]_{y_a} \in \text{MDAT}_{y_a}$
- $\gamma(u^L) = \gamma(t_1^L)$
falls $u^L = [t_1^L]_{x_a} \in \text{MDAT}_{x_a}$
- $\gamma(u^A) = \gamma(t_1^A)$
falls $u = [t_1^L]_{x_l} \in \text{MDAT}_{x_l}$
- $\gamma(u^L) = \gamma(u_1^L) \cdot \dots \cdot \gamma(u_n^L)$
falls $u^L = [u_1^L, \dots, u_n^L]_{x_l} \in \text{MDAT}_{x_l}$
- $\gamma(u^A) = \gamma(u_1^A) \cdot \dots \cdot \gamma(u_n^A)$
falls $u^A = [u_1^A, \dots, u_n^A]_{x_a} \in \text{MDAT}_{x_a}$

mit $\Phi_i(u)$

- $\Phi_i(\tau_{x_l}) = \Phi_i(\tau_{x_a}) = \Phi_i(\tau_{y_l}) = \Phi_i(\tau_{y_a}) = 1$
- $\Phi_i(t^L) = \sum_j \beta_{ij}^L \cdot \Phi_i(u_1^L)$
falls $t^L = [u_1^L]_{y_l} \in \text{MDAT}_{y_l}$

- $\Phi_i(t^A) = \sum_j \beta_{ij}^A \cdot \Phi_i(u_1^A)$
 $\text{falls } t^A = [u_1^A]_{y_a} \in \text{MDAT}_{y_a}$
- $\Phi_i(t^L) = m^{n-(\varrho(u_1^A)+\dots+\varrho(u_n^A))}$
 $\sum_{\substack{\mu_1, \dots, \mu_l \\ \nu_1, \dots, \nu_n}} \alpha_{i\mu_1}^L \cdot \dots \cdot \alpha_{i\mu_l}^L \cdot \delta_{i\nu_1}^L \cdot \dots \cdot \delta_{i\nu_n}^L$
 $\cdot \Phi_{\mu_1}(u_1^L) \cdot \dots \cdot \Phi_{\mu_l}(u_l^L)$
 $\cdot \Phi_{\nu_1}(u_1^A) \cdot \dots \cdot \Phi_{\nu_n}(u_n^A)$
 $\text{falls } t^L = [u_1^L, \dots, u_l^L, u_1^A, \dots, u_n^A]_{y_l} \in \text{MDAT}_{y_l}$
 $\text{und } (n+l > 1) \text{ oder } (l=0 \text{ und } n \geq 1)$
- $\Phi_i(t^A) = \left(\frac{1}{m}\right)^{l-(\varrho(u_1^L)+\dots+\varrho(u_l^L))}$
 $\sum_{\substack{\mu_1, \dots, \mu_l \\ \nu_1, \dots, \nu_n}} \delta_{i\mu_1}^A \cdot \dots \cdot \delta_{i\mu_l}^A \cdot \alpha_{i\nu_1}^A \cdot \dots \cdot \alpha_{i\nu_n}^A$
 $\cdot \Phi_{\mu_1}(u_1^L) \cdot \dots \cdot \Phi_{\mu_l}(u_l^L)$
 $\cdot \Phi_{\nu_1}(u_1^A) \cdot \dots \cdot \Phi_{\nu_n}(u_n^A)$
 $\text{falls } t^A = [u_1^L, \dots, u_l^L, u_1^A, \dots, u_n^A]_{y_a} \in \text{MDAT}_{y_a}$
 $\text{und } (n+l > 1) \text{ oder } (l \geq 1 \text{ und } n=0)$
- $\Phi_i(u^L) = \Phi_i(t_1^L)$
 $\text{falls } u^L = [t_1^L]_{x_l} \in \text{MDAT}_{x_l}$
- $\Phi_i(u^A) = \Phi_i(t_1^A)$
 $\text{falls } u^A = [t_1^A]_{x_a} \in \text{MDAT}_{x_a}$
- $\Phi_i(u^L) = \sum_{j, \mu_1, \dots, \mu_n} \omega_{ij}^L \cdot \alpha_{j\mu_1}^L \cdot \dots \cdot \alpha_{j\mu_n}^L$
 $\Phi_{\mu_1}(u_1^L) \cdot \dots \cdot \Phi_{\mu_n}(u_n^L)$
 $\text{falls } u^L = [u_1^L, \dots, u_n^L]_{x_l} \in \text{MDAT}_{x_l}$
- $\Phi_i(u^A) = \sum_{j, \mu_1, \dots, \mu_n} \omega_{ij}^A \cdot \alpha_{j\mu_1}^A \cdot \dots \cdot \alpha_{j\mu_n}^A$
 $\Phi_{\mu_1}(u_1^A) \cdot \dots \cdot \Phi_{\mu_n}(u_n^A)$
 $\text{falls } u^A = [u_1^A, \dots, u_n^A]_{x_a} \in \text{MDAT}_{x_a}$

Wobei die Zugehörigkeiten

$$\begin{array}{ll} u_i^L \in \text{MDAT}_{x_l} & u_i^A \in \text{MDAT}_{x_a} \\ y_i^L \in \text{MDAT}_{y_l} & y_i^A \in \text{MDAT}_{y_a} \end{array}$$

gelten.

Beweis 3.13 *Ein Beweis kann mittels vollständiger Induktion nach der Ableitungsordnung p geführt werden. Hier wollen wir nur einen **groben Abriss der Vorgehensweise** geben. Der vollständige Beweis findet sich im Anhang.*

Wir gehen zunächst von der Darstellung der p -ten Ableitung der Zuwachsfunktion $k_i^L(h)$ gemäß (3.11) aus. Nach (3.8) sind die Ableitungen der Zwischenwerte $\tilde{a}_i^L(h), \tilde{d}_i^L$ Linearkombinationen der entsprechenden Ableitungen von $k_i^L(h)$ bzw. $k_i^A(h)$.

Damit können wir eine Reihe von konstanten Faktoren aus den Multilinear-Formen aus (3.11) herausziehen und gelangen nun zu Multilinear-Formen der Bauart:

$$\left(\frac{\partial q_l}{\partial x_l}\right)^{-1} \frac{\partial^l q_l}{\partial x_l^l} \left(\left(k_{\nu_1}^L\right)^{(\mu_1)}, \dots, \left(k_{\nu_l}^L\right)^{(\mu_l)} \right) \text{ und } \left(\frac{\partial q_l}{\partial x_l}\right)^{-1} \frac{\partial^{l+n} w_l}{\partial x_l^l \partial x_a^n} \left(\left(k_{\lambda_1}^L\right)^{(\mu_1)}, \dots, \left(k_{\xi_n}^A\right)^{(\nu_n)} \right)$$

sowie zu Differentialen der Form:

$$\left(\frac{\partial q_l}{\partial x_l}\right)^{-1} \frac{\partial w_l}{\partial x_l} \left(k_j^L\right)^{(p-1)} \quad \text{und} \quad \left(\frac{\partial q_l}{\partial x_l}\right)^{-1} \frac{\partial w_l}{\partial x_l} \left(k_j^A\right)^{(p-1)}$$

Hierauf wenden wir die Induktionshypothese an, und nach neuerlichem Ausklammern aller konstanten Werte aus den Multilinearformen und Differentialen ergeben sich Multilinearformen und Differentiale der Form:

$$\begin{aligned} &\left(\frac{\partial q_l}{\partial x_l}\right)^{-1} \frac{\partial^l q_l}{\partial x_l^l} \left(F(u_1^L), \dots, F(u_l^L)\right) \quad , \quad \left(\frac{\partial q_l}{\partial x_l}\right)^{-1} \frac{\partial^{l+n} w_l}{\partial x_l^l \partial x_a^n} \left(F(u_1^L), \dots, F(u_n^A)\right) \\ &\left(\frac{\partial q_l}{\partial x_l}\right)^{-1} \frac{\partial w_l}{\partial x_l} F(u_1^L)(x_{l,0}, x_{a,0}) \quad , \quad \left(\frac{\partial q_l}{\partial x_l}\right)^{-1} \frac{\partial w_l}{\partial x_a} F(u_1^A)(x_{l,0}, x_{a,0}) \end{aligned}$$

Nun korrespondieren diese mit den rekursiv definierten MDAT_{x_l} - bzw. LMDAT_{x_l} -Bäumen des letzten Abschnitts:

$$\begin{aligned} &[u_1^L, \dots, u_l^L]_{x_l}, \quad [u_1^L, \dots, u_n^A]_{y_l} \\ &[[u_1^L]_{y_l}]_{x_l}, \quad [[u_1^A]_{y_l}]_{x_l} \end{aligned}$$

Es bleibt nun zu überlegen, dass hierdurch alle Bäume bis zur Ordnung p gebildet werden können und, dass sich diese Beschreibungen nicht überschneiden, jeder Baum ($\in \text{LMDAT}$) also genau einmal entsteht.

Weiter kann man sich klarmachen, dass die jeweils ausgeklammerten konstanten Werte gerade die Koeffizienten $\gamma(\cdot) \cdot \Phi(\cdot)$ ergeben.

□

Damit sind wir nun in der Lage, auch die Ableitungen der numerischen Approximation des compound-Schrittes mithilfe der MDAT -Bäume anzugeben:

Theorem 3.14 (Ableitungen der Approximation des compound-Schrittes)

Für die Ableitungen von $x_{l,1}(h)$ und $x_{a,1}(h)$ an der Stelle $h = 0$ gilt:

$$\begin{aligned} x_{l,1}^{(p)}(0) &= \sum_{\substack{u^L \in \text{MDAT}_{x_l} \\ \varrho(u^L)=p}} \sum_{i=1}^s b_i^L \cdot \gamma(u^L) \cdot \Phi_i(u^L) \cdot \alpha(u^L) \cdot F(u^L)(x_{l,0}, x_{a,0}) \\ x_{a,1}^{(p)}(0) &= \sum_{\substack{u^A \in \text{MDAT}_{x_a} \\ \varrho(u^A)=p}} \sum_{i=1}^s b_i^A \cdot \gamma(u^A) \cdot \Phi_i(u^A) \cdot \alpha(u^A) \cdot F(u^A)(x_{l,0}, x_{a,0}) \end{aligned}$$

Dies führt nun direkt auf die Taylorentwicklung der Näherungen $x_{l,1}, x_{a,1}$ des compound-Schrittes:

Lemma 3.15 (Taylorentwicklung der Approximationen) Für die Taylorentwicklungen der numerischen compound-Schritt-Approximationen $x_{l,1}, x_{a,1}$ durch das nichttransformierte Multirate-Verfahren (2.18) gilt:

$$\begin{aligned} x_{l,1} &= \sum_{\substack{u^L \in \text{MDAT} \\ \varrho(u^L) \leq q}} \sum_{i=1}^s b_i^L \cdot \gamma(u^L) \cdot \Phi_i(u^L) \cdot \alpha(u^L) \cdot F(u^L)(x_{l,0}, x_{a,0}) \cdot \frac{h_l^{\varrho(u^L)}}{\varrho(u^L)!} \\ &\quad + \tilde{R}_q^L(h_l), \end{aligned} \tag{3.12a}$$

$$\begin{aligned} x_{a,1} &= \sum_{\substack{u^A \in \text{MDAT} \\ \varrho(u^A) \leq q}} \sum_{i=1}^s b_i^A \cdot \gamma(u^A) \cdot \Phi_i(u^A) \cdot \alpha(u^A) \cdot F(u^A)(x_{l,0}, x_{a,0}) \cdot \frac{h_a^{\varrho(u^A)}}{\varrho(u^A)!} \\ &\quad + \tilde{R}_q^A(h_a) \end{aligned} \tag{3.12b}$$

mit Resten $\|\tilde{R}_q^L(h_l)\| = \mathcal{O}(h_l^{q+1})$ und $\|\tilde{R}_q^A(h_a)\| = \mathcal{O}(h_a^{q+1})$.

Durch Vergleich der Taylorreihe (3.11) der exakter Lösung und der Taylorentwicklung (3.15) der numerischen Approximation erhalten wir die **zentrale Konsistenzaussage**:

Theorem 3.16 (Konsistenzbedingungen im compound-Schritt)

Das nichttransformierte Multirate-Verfahren (2.18) besitzt im compound-Schritt die Konsistenzordnung p (s. Def 3.1) genau dann, wenn gilt:

$$\begin{aligned} \sum_{i=1}^s b_i^L \cdot \Phi_i(u^L) &= \frac{1}{\gamma(u^L)} & \forall u^L \in \text{MDAT}_{x_l} \quad \text{mit} \quad \varrho(u^L) \leq q \\ \sum_{i=1}^s b_i^A \cdot \Phi_i(u^A) &= \frac{1}{\gamma(u^A)} & \forall u^A \in \text{MDAT}_{x_a} \quad \text{mit} \quad \varrho(u^A) \leq q \end{aligned}$$

Die Werte $\Phi_i(\cdot)$ und $\gamma(\cdot)$ sind nach dem Lemma 3.12 rekursiv definiert. Bei einer genauen Betrachtung der Rekursionsvorschriften stellen wir jedoch fest, dass wir $\Phi_i(\cdot)$ und $\gamma(\cdot)$ auch direkt am zugehörigen Baum „ablesen“ können.

Den Wert $\gamma(u)$ für einen Baum $u \in \text{MDAT}$ erhalten wir durch eine schrittweise Zerlegung des Baumes u „von unten“. D. h., wir nehmen von der Wurzel her aufsteigend jeweils einen Knoten (also jeweils eine Wurzel) weg und betrachten rekursiv die neu entstandenen kleineren Bäume. Gingen vom entfernten Knoten mehrere Äste ab, so erhalten wir durch seine Löschung mehrere Einzelbäume, auf denen wir jeweils einen Wert $\gamma(\cdot)$ nach gleichem Muster berechnen. $\gamma(u)$ ergibt sich dann aus folgendem rekursiven Vorgehen:

- Ist die entfernte Wurzel ein leerer Knoten (Kreis oder Quadrat), so ist das $\gamma(\cdot)$ des Baumes, von dem wir diese Wurzel entfernt haben, das Produkt der $\gamma(\cdot)$'s der nun entstandenen Bäume.
- Ist die entfernte Wurzel ein voller Knoten (Kreis oder Quadrat), so ist das $\gamma(\cdot)$ des Baumes, von dem wir diese Wurzel entfernt haben, das Produkt der $\gamma(\cdot)$'s der nun entstandenen Bäume und der Ordnung $\varrho(\cdot)$ des Baumes vor der Entfernung der Wurzel. (Diese Ordnung ist gerade um 1 größer als die Summe der Ordnungen der neu entstandenen Bäume.)
- $\gamma(\tau_{x_l}) = \gamma(\tau_{x_a}) = \gamma(\tau_{y_l}) = \gamma(\tau_{y_a}) = 1$.
Die „trivialen“ Bäume sind das Rekursionsende.

Dies sei nun noch an zwei kleinen Beispielen aufgezeigt:

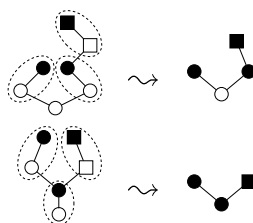
$$\begin{aligned} \gamma(\text{Diagram 1}) &= \gamma(\text{Diagram 2}) = 3 \cdot \gamma(\text{Diagram 3}) = 3 \cdot \gamma(\text{Diagram 4}) \cdot \gamma(\text{Diagram 5}) = 3 \cdot 1 \cdot 1 = 3 \\ \gamma(\text{Diagram 6}) &= \gamma(\text{Diagram 7}) = 3 \cdot \gamma(\text{Diagram 8}) = 3 \cdot \gamma(\text{Diagram 9}) = 3 \cdot 2 \cdot \gamma(\text{Diagram 10}) = 3 \cdot 2 \cdot 1 = 6 \end{aligned}$$

Und auch die Werte $\Phi_i(u)$ lassen sich nach einer kleinen „Reduktion“ der Bäume direkt ablesen.

Hierzu wird ausgenutzt, dass $\Omega^\lambda = (\mathcal{B})^{-1}$ ist. Dies führt bei der rekursiven Berechnung von $\Phi_i(u)$ nach Lemma 3.12 dazu, dass sich ω 's und β 's teilweise auslöschen. Diese Auslöschung tritt genau dann auf, wenn im Baum auf einen leeren Knoten ein voller Knoten derselben Gestalt folgt (an einen leeren Knoten schliesst entweder genau ein voller Knoten derselben Gestalt an oder mehrere leere Knoten beliebiger Gestalt).

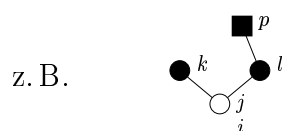
Um nun am Baum direkt die Gestalt von $\Phi_i(u)$ erkennen zu können, transformieren wir diesen in einen „reduzierten Baum“ $\tilde{u}$, der in Bezug auf Φ_i gleichwertig zum Ursprungsbaum u ist (nicht jedoch bezgl. der Differentiale!). Die Reduktion

ist eine einfache Verschmelzung der Kombination leerer–voller Knoten zu einem vollen Knoten, wie die folgenden Beispiele zeigen:



Aus den reduzierten Bäumen lassen sich die Werte Φ_i nun durch folgende Schritte einfach ableiten:

1. Von unten nach oben im Baum werden Indizes $(i, j, \dots)$, beginnend mit i , vergeben. Dabei erhält jeder leere Knoten zwei aufeinanderfolgende Indizes und jeder volle einen.



2. Wird nun jeweils zwei Indizes, die entweder am selben Knoten stehen oder über einen Ast verbunden sind (hier ist vom unteren Knoten der höhere und vom oberen Knoten der niedrigere Index gemeint, falls an einem Knoten zwei Indizes stehen), ein Faktor zugeordnet, so ist $\Phi_i(u)$ die Summe über alle Indizes des Produktes dieser Faktoren.

Diese Faktoren sind:

α_{ij}^L	falls „i,j“ die Indizes zweier runder Knoten sind und vom unteren der beiden zwei oder mehr Äste abgehen
α_{ij}^A	falls „i,j“ die Indizes zweier quadratischer Knoten sind und vom unteren der beiden zwei oder mehr Äste abgehen
β_{ij}^L	falls „i,j“ die Indizes zweier runder Knoten sind und vom unteren der beiden nur ein Ast abgeht
β_{ij}^A	falls „i,j“ die Indizes zweier quadratischer Knoten sind und vom unteren der beiden nur ein Ast abgeht
$\delta_{ij}^L \cdot m^{n-\varrho}$	falls „i“ der Index eines runden Knoten und „j“ der Index eines quadratischen Knotens ist. Hier ist n die Anzahl der Äste, die vom unteren runden Knoten zu höheren quadratischen Knoten abgehen, und ϱ die Summe der Ordnungen der Teilbäume, die jene als Wurzel besitzen

$\delta_{ij}^L \cdot \left(\frac{1}{m}\right)^{n-\varrho}$	falls „i“ der Index eines quadratischen Knoten und „j“ der Index eines runden Knotens ist. Hier ist n die Anzahl der Äste, die vom unteren quadratischen Knoten zu höheren runden Knoten abgehen, und ϱ die Summe der Ordnungen der Teilbäume, die jene als Wurzel besitzen
ω_{ij}^L	falls „i,j“ die beiden Indizes eines leeren runden Knotens sind
ω_{ij}^A	falls „i,j“ die beiden Indizes eines leeren quadratischen Knotens sind

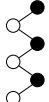
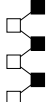
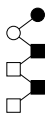
Aus dem Aufbau eines Baumes u erhalten wir nun also auf einfache Weise die zugehörige Bedingungsgleichung:

1. Bestimme mit dem beschriebenen rekursiven Vorgehen zum Baum u den Wert $\gamma(u)$.
2. Nach der Reduktion des Baumes u zum Baum $\tilde{u}$ kann an diesem reduzierten Baum der Wert Φ_i unschwer „abgelesen“ werden.
3. Die rechte Seite der zum Baum gehörenden Bedingungsgleichung ist nun die Summe aller Produkte $\Phi_i(u)$ mit dem Vorfaktor b_i^L , falls der Wurzelknoten rund ist, und b_i^A , falls er quadratisch ist. Die Summe läuft über alle in Phi_i auftauchenden Indizes $i, j, \dots$
4. Die linke Seite der Bedingungsgleichung an die Verfahrenskoeffizienten ist $\gamma(u)^{-1}$.

Wir erhalten beispielsweise zum Baum $u^L =$ 

$$\begin{array}{c}
 \text{Diagram of } u^L \quad \rightsquigarrow \quad \text{Diagram of } \tilde{u} \quad \rightsquigarrow \quad \sum_{i,j,k,l,p} b_i^L \omega_{ij}^L \alpha_{jk}^L \alpha_{jl}^L \delta_{lp}^L \cdot m^{1-1} = \frac{1}{2} \\
 \gamma(u^L) = 2
 \end{array}$$

Tabelle (3.3) stellt mit, den Abkürzungen $\alpha_j^L = \sum_{k=1}^s \alpha_{jk}^L, \dots$, schließlich die Bäume und zugehörigen Bedingungsgleichungen für den latenten und aktiven Teil bis zur Ordnung 3 gegenüber.

	$\sum_i b_i^L = 1$		$\sum_i b_i^A = 1$
	$\sum_{i,j} b_i^L \omega_{ij}^L (\alpha_j^L)^2 = 1$		$\sum_{i,j} b_i^A \omega_{ij}^A (\alpha_j^A)^2 = 1$
	$\sum_{i,j} b_i^L \beta_i^L \beta_j^L = \frac{1}{2}$		$\sum_{i,j} b_i^A \beta_i^A \beta_j^A = \frac{1}{2}$
	$\sum_{i,j} b_i^L \delta_i^L = \frac{1}{2}$		$\sum_{i,j} b_i^A \delta_i^A = \frac{1}{2}$
	$\sum_{i,j} b_i^L (\alpha_i^L)^2 = \frac{1}{3}$		$\sum_{i,j} b_i^A (\alpha_i^A)^2 = \frac{1}{3}$
	$\sum_{i,j} b_i^L \alpha_i^L \delta_i^L = \frac{1}{3}$		$\sum_{i,j} b_i^A \alpha_i^A \delta_i^A = \frac{1}{3}$
	$\sum_{i,j} b_i^L (\delta_i^L)^2 = \frac{1}{3}$		$\sum_{i,j} b_i^A (\delta_i^A)^2 = \frac{1}{3}$
	$\sum_{i,j} b_i^L \beta_{ij}^L \beta_j^L = \frac{1}{6}$		$\sum_{i,j} b_i^A \beta_{ij}^A \beta_j^A = \frac{1}{6}$
	$\sum_{i,j} b_i^L \beta_{ij}^L \delta_j^L = \frac{1}{6}$		$\sum_{i,j} b_i^A \beta_{ij}^A \delta_j^A = \frac{1}{6}$

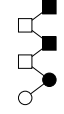
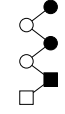
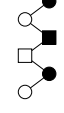
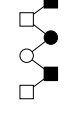
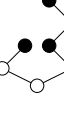
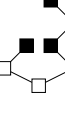
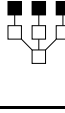
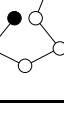
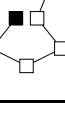
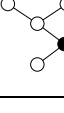
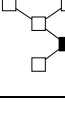
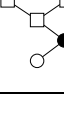
	$\sum_{i,j} b_i^L \cdot \frac{1}{m} \cdot \delta_{ij}^L \beta_j^A = \frac{1}{6}$		$\sum_{i,j} b_i^A \cdot m \cdot \delta_{ij}^A \beta_j^L = \frac{1}{6}$
	$\sum_{i,j} b_i^L \cdot \frac{1}{m} \cdot \delta_{ij}^L \delta_j^A = \frac{1}{6}$		$\sum_{i,j} b_i^A \cdot m \cdot \delta_{ij}^A \delta_j^L = \frac{1}{6}$
	$\sum_{i,j,l} b_i^L \omega_{ij}^L \alpha_j^L \alpha_{jl}^L \beta_l^L = \frac{1}{2}$		$\sum_{i,j,l} b_i^A \omega_{ij}^A \alpha_j^A \alpha_{jl}^A \beta_l^A = \frac{1}{2}$
	$\sum_{i,j,l} b_i^L \omega_{ij}^L \alpha_j^L \alpha_{jl}^L \delta_l^L = \frac{1}{2}$		$\sum_{i,j,l} b_i^A \omega_{ij}^A \alpha_j^A \alpha_{jl}^A \delta_l^A = \frac{1}{2}$
	$\sum_{i,j} b_i^L \omega_{ij}^L (\alpha_j^L)^3 = 1$		$\sum_{i,j} b_i^A \omega_{ij}^A (\alpha_j^A)^3 = 1$
	$\sum_{i,j,k,l} b_i^L \omega_{ij}^L \alpha_j^L \alpha_{jl}^L \omega_{lk}^L (\alpha_k^L)^2 = 1$		$\sum_{i,j,k,l} b_i^A \omega_{ij}^A \alpha_j^A \alpha_{jl}^A \omega_{lk}^A (\alpha_k^A)^2 = 1$
	$\sum_{i,j,k} b_i^L \beta_{ij}^L \omega_{jk}^L (\alpha_k^L)^2 = \frac{1}{3}$		$\sum_{i,j,k} b_i^A \beta_{ij}^A \omega_{jk}^A (\alpha_k^A)^2 = \frac{1}{3}$
	$\sum_{i,j,k} b_i^L \cdot \frac{1}{m} \cdot \delta_{ij}^L \omega_{jk}^A (\alpha_k^A)^2 = \frac{1}{3}$		$\sum_{i,j,k} b_i^A \cdot m \cdot \delta_{ij}^A \omega_{jk}^L (\alpha_k^L)^2 = \frac{1}{3}$

Tabelle 3.3: Bäume und Bedingungsgleichungen bis Ordnung $p = 3$

Wir sind nun in der Lage, Koeffiziente für das nichttransformierte Multirate-Verfahren (2.18) so zu bestimmen, dass dieses im compound-Schritt eine geforderte Konsistenzordnung besitzt. Hierfür müssen wir die Koeffizienten $b_i^\lambda, \alpha_{ij}^\lambda, \beta_{ij}^\lambda, \delta_{ij}^\lambda$ so wählen, dass die Bedingungsgleichungen bis zur gewünschten Ordnung alle bestimmt sind. Dabei ist darauf zu achten, dass die Koeffizienten ω_{ij}^L und ω_{ij}^A nur

als „Platzhalter“ für die Matrixeinträge aus $\Omega^L = (\mathcal{B}^L)^{-1}$ bzw. $\Omega^A = (\mathcal{B}^A)^{-1}$ fungieren.

Außerdem ist zu beachten, dass der Wert m keine Unbekannte, sondern ein freier Parameter ist. Er beschreibt das Verhältnis von großer zu kleiner Schrittweite im Compound-Schritt. Da m also ein sich ändernder Parameter ist, werden unsere Koeffizienten von m abhängig sein. Dies wiederum heißt, dass in einer späteren Implementierung die Verfahrenskoeffizienten in jedem compound-Schritt neu zu berechnen sind.

Wir erinnern uns daran, dass wir in einer Implementierung mit einem **eingebetteten Verfahren**

$$\hat{x}_{\lambda,1} = x_{\lambda,0} + \sum_{i=1}^{\hat{s}} b_i^\lambda \cdot k_i^\lambda, \quad \lambda = L, A$$

die Schrittweiten steuern wollen. Also müssen wir auch für dieses Konsistenzforderungen aufstellen.

Das eingebettete Verfahren benutzt dieselben Zuwächse wie das übergeordnete Verfahren, gewichtet diese lediglich anders. Die Verfahrenskoeffizienten, die Einfluss auf die Zuwächse haben, sind die α 's, β ' und δ 's. Die Gewichte sind die b 's bzw. $\hat{b}$'s.

Für die zu erfüllenden Bedingungsgleichungen bedeutet dies Folgendes:

Zu den Koeffizientengleichungen des übergeordneten Verfahrens der Konsistenzordnung p kommen die Bedingungsgleichungen des eingebetteten Verfahrens der Ordnung $\hat{p}$ hinzu. Der einzige Unterschied in diesen zusätzlichen Forderungen ist, dass in diesen statt b_i^L und b_i^A nun $\hat{b}_i^L$ und $\hat{b}_i^A$ steht.

Im späteren Fall eines Multirate-Verfahrens der Ordnung 3 mit eingebettetem Verfahren der Ordnung 2 haben wir also zu den $(2 \cdot 17)$ Bedingungsgleichungen zusätzlich $(2 \cdot 4)$ Forderungen zu erfüllen.

3.4 Konsistenz der späteren Mikroschritte und dense-output

Nach den Ausführungen zur Konsistenz des compound-Schrittes wenden wir uns nun der **Konsistenz der späteren Mikroschritte** zu.

Dies gestaltet sich wesentlich einfacher, da wir einen Index-1-Löser für nicht-gekoppelte DAEs verwenden und die Einkopplung des latenten Teils über ein dense-output-Schema realisieren werden.

Naiv wird man sich vorstellen, dass die „Kombination“ von Index-1-Löser und dense-output konsistent sind, wenn Index-1-Löser und dense-output jeweils für sich genommen entsprechend „genau“ sind. Dass dies tatsächlich zutrifft, werden wir im Folgenden nachweisen.

Damit wird sich der Aufwand, die Konsistenz der späteren Mikroschritte zu gewährleisten, auf die Betrachtung des dense-output-Schemas reduzieren. Denn vom übernommenen Index-1-Löser kennen wir die Konsistenzordnung,

Wie wir in der Verfahrensentwicklung bereits gesehen haben, würde sich das gekoppelte Hessenbergsystem (2.11) in den späteren Mikroschritten als ein nicht-autonomes Index-1-System darstellen, wenn wir den genauen Lösungsverlauf der latenten Funktion $x_l(t)$ kennen würden:

$$\begin{aligned} \dot{y}_a &= \tilde{w}_a(x_a, t) \quad (\stackrel{\text{def}}{=} w_a(x_l(t), x_a)) \\ 0 &= y_a - g_a(x_a) \\ t &\in [t_0 + h_a, t_0 + h_l] \end{aligned} \quad (3.13)$$

Dieses nichtgekoppelte System könnten wir nun mit einem Index-1-Verfahren ϕ approximativ in der Unbekannten x_a bestimmen.

Nun kennen wir aber den Verlauf von $x_l(t)$ im betrachteten Integrationsintervall $[t_0 + h_a, t_0 + h_l]$ nicht genau, sondern müssen uns mit einer Näherung begnügen. Hier setzten wir ein **dense-output-Schema** an:

$$\begin{aligned} x_l(t_0 + \theta h_l) &\approx x_l^{\text{ds}}(\theta) = x_{l,0} + \sum_{i=1}^s b_i^L(\theta) \cdot k_i^L \\ \theta &\in [0, 1], \end{aligned} \quad (3.14)$$

Der Wunsch, dennoch den Index-1-Löser ϕ zu benutzen, führt zu folgender Idee:

i) Ersetze in (3.13) $\tilde{w}_a(x_a, t) = w_a(x_l(t), x_a)$ durch

$$\begin{aligned} w_a^{\text{ds}}(x_a, t) &\stackrel{\text{def}}{=} w_a(x_l^{\text{ds}}(\theta), x_a) \\ t &= t_0 + \theta \cdot h_l, \quad \theta \in [0, 1] \end{aligned} \quad (3.15)$$

ii) Wende auf das so entstandene neue Index-1-Problem

$$\begin{aligned} \dot{y}_a &= w_a^{\text{ds}}(x_a, t) \\ 0 &= y_a - g_a(x_a) \\ t &\in [t_0 + h_a, t_0 + h_l] \end{aligned} \quad (3.16)$$

das Index-1-Verfahren ϕ an.

Dieses Vorgehen wollen wir von nun an als

Index-1-dense-Verfahren ϕ^{ds}

bezeichnen.

Eine Angabe zur Konsistenzordnung dieses Index-1-dense-Verfahrens ϕ^{ds} liefert das folgende Lemma.

Lemma 3.17 *Besitzt ein Index-1-Löser ϕ die Konsistenzordnung p für das Index-1-Problem (3.13), erfüllt ein dense-output-Schema x_l^{ds} (3.14) die Genauigkeitsbedingung*

$$\|x_l(t_0 + \theta h_l) - x_l^{\text{ds}}(\theta)\| \leq c_1 \cdot h_l^p \quad \forall \theta \in [0, 1]$$

und werden die Mikroschrittweiten h nicht „zu klein“, dann besitzt das Index-1-dense-Verfahren ϕ^{ds} die Konsistenzordnung p .

Beweis 3.18

Es gelten zunächst folgende Benennungen:

- i) $x_a(t)$ ist die exakte Lösung des Ausgangsproblem (3.13).
- ii) $\tilde{x}_a(t)$ ist die exakte Lösung des abgeänderten Problems (3.16).
- iii) $x_a^{\text{ds}}(h)$ ist das Ergebnis des Index-1-dense-Verfahrens ϕ^{ds} nach einem Schritt mit der Schrittweite h .

Mit der Index-1-Hypothese ($\frac{\partial g_a}{\partial x_a}$ ist in einer Umgebung der Lösung invertierbar) gilt damit für $x_a(t), \tilde{x}_a(t)$

$$\begin{aligned} \dot{x}_a &= \left(\frac{\partial g_a}{\partial x_a} \right)^{-1} (x_a) \cdot w_a(x_l(t), x_a) \\ &\stackrel{\text{def}}{=} \mathcal{W}_a(x_l(t), x_a) \end{aligned}$$

$$\dot{\tilde{x}}_a = \mathcal{W}_a(x_l^{\text{ds}}(\theta), \tilde{x}_a), \quad t = t_0 + \theta h_l$$

Da $\frac{\partial g_a}{\partial x_a}$ in einer Umgebung von $x_a(t)$ invertierbar ist und $w_a(x_l, x_a)$ in beiden Parametern eine Lipschitzbedingung erfüllt, erfüllt auch $\mathcal{W}_a$ zwei solche Lipschitzbedingungen:

$$\begin{aligned} \|\mathcal{W}_a(u_1, v) - \mathcal{W}_a(u_2, v)\| &\leq \tilde{L}_1 \|u_1 - u_2\| \\ \|\mathcal{W}_a(u, v_1) - \mathcal{W}_a(u, v_2)\| &\leq \tilde{L}_2 \|v_1 - v_2\| \end{aligned} \tag{3.17}$$

Nun ist zu zeigen

$$\|x_a(t_0 + h) - x_a^{\text{ds}}(h)\| \leq C \cdot h^{p+1} \quad \forall h \in [h_a, h_l].$$

Mit der Dreiecksungleichung gilt:

$$\|x_a(t_0 + h) - x_a^{\text{ds}}(h)\| \leq \|x_a(t_0 + h) - \tilde{x}_a(t_0 + h)\| + \|\tilde{x}_a(t_0 + h) - x_a^{\text{ds}}(h)\|$$

mit

$$\begin{aligned}
\|x_a(t_0 + h) - \tilde{x}_a(t_0 + h)\| &\leq \int_{t_0}^{t_0+h} \|\mathcal{W}_a(x_l(\tau), x_a) - \mathcal{W}_a(x_l^{\text{ds}}(\theta_\tau), \tilde{x}_a)\| d\tau \\
&\stackrel{(3.17)}{\leq} \int_{t_0}^{t_0+h} \tilde{L}_1 \underbrace{\|x_l(\tau) - x_l^{\text{ds}}(\theta_\tau)\|}_{\leq c_1 \cdot h_l} d\tau + \int_{t_0}^{t_0+h} \tilde{L}_2 \|x_a(\tau) - \tilde{x}_a(\tau)\| d\tau \\
&\leq \tilde{L}_1 \cdot c_1 \cdot h \cdot h_l^p + \int_0^h \tilde{L}_2 \|x_a(t_0 + \tau) - \tilde{x}_a(t_0 + \tau)\| d\tau \\
&\leq \tilde{L}_1 \cdot c_1 \cdot h \cdot h_l^p \cdot e^{\tilde{L}_2 \cdot h} \quad \text{nach dem Lemma von Gronwall}
\end{aligned}$$

und

$$\|\tilde{x}_a(t_0 + h) - x_a^{\text{ds}}(h)\| \leq c_2 \cdot h^{p+1}$$

Da $x_a^{\text{ds}}(h)$ das Ergebnis eines Schrittes h des Index-1-Verfahrens ϕ , angewandt auf (3.16) ist

Wir haben vorausgesetzt, dass die Mikroschrittweiten nicht zu klein werden. Dies bedeutet, dass das Verhältnis von Schrittweite h_l zu allen Mikroschrittweiten h beschränkt bleibt, bzw.

$$h_l \leq m_{\max} \cdot h, \quad m_{\max} \text{ nicht zu groß}$$

Mit dieser Einschränkung der Schrittweiten haben wir also insgesamt:

$$\begin{aligned}
\|x_a(t_0 + h) - x_a^{\text{ds}}(h)\| &\leq h^{p+1} \cdot (\tilde{L}_1 \cdot c_1 \cdot e^{\tilde{L}_2 \cdot h_l} + c_2) \\
&\leq C \cdot h^{p+1}.
\end{aligned}$$

□

Bem.:

Der vorangegangene Beweis zeigt sehr deutlich, dass ein Multirate-Verfahren nicht zu „übermütig“ angegangen werden darf. Arbeitet man mit zu großen relativen Schrittweitenunterschieden, dann ist immerhin zu befürchten, dass Probleme in der Genauigkeit auftreten. In diesem Fall hielte nämlich die letzte Abschätzung nicht mehr Stand, und es wäre mit dem Verlust mindestens einer Ordnung bzgl. h zu rechnen. Probleme mit der Genauigkeit können auch auftreten, wenn die Kopplung zwischen latenten und aktiven Teilen der Netzwerkgleichung zu stark sind. Diese schläge sich in den Lipschitzkonstanten $\tilde{L}_1, \tilde{L}_2$ nieder, wiederum mit der Gefahr, Konsistenzordnungen zu verlieren.

Welche maximalen Schrittweitenverhältnisse benutzt werden können, also wie „stark“ das Multirate betrieben werden kann, hängt mit Sicherheit vom Einzelfall ab.

In den späteren Mikroschritten des Multirate-Verfahrens sind wir an einer gewissen Konsistenzordnung p interessiert.

Um diese zu erreichen, liefert uns Lemma 3.17 also folgendes Vorgehen:

- i) Wahl eines Index-1-Lösers mit Schrittweitensteuerung über die algebraischen Unbekannten (die Spannungen). Der lokale Diskretisierungsfehler muss von der Ordnung $\mathcal{O}(h^{p+1})$ sein.
- ii) Anpassung des dense-output-Schemas (3.14) so, dass der Fehler von der Größenordnung $\mathcal{O}(h_l^p)$ ist.

Schritt 1: Index-1-Integrationsschema

Da unsere Index-1-Systeme elektrische Netzwerke darstellen, bietet sich hier das CHORAL-Verfahren von Günther aus [G98] an. CHORAL ist, wie unser Verfahren im compound-Schritt, ROW-basiert mit einer spannungsorientierter Schrittweitensteuerung. Die Konvergenzordnung dieses Verfahrens ist 3, der lokale Diskretisierungsfehler also von der Ordnung $\mathcal{O}(h^4)$. Zur Schrittweitensteuerung ist ein Verfahren der Ordnung 2 eingebettet (lokaler Fehler also von der Größenordnung $\mathcal{O}(h^3)$).

Schritt 2: Dense-output

Wir haben nun das dense-output-Schema (3.14):

$$\begin{aligned} x_l^{\text{ds}}(\theta) &= x_{l,0} + \sum_{i=1}^s b_i^L(\theta) \cdot k_i^L \\ \theta &\in [0, 1], \end{aligned}$$

so anzupassen, dass der Fehler in der Größenordnung $\mathcal{O}(h_l^p)$ liegt (bei Verwendung von CHORAL $p = 3$):

$$\|x_l(t_0 + \theta h_l) - x_l^{\text{ds}}(\theta)\| \leq c_1 \cdot h_l^p, \quad \forall \theta \in [0, 1] \quad (3.18)$$

Wie im Falle des eingebetteten Verfahrens $\hat{x}_l$ im compound-Schritt, benutzt der dense-output dieselben Zuwächse wie das Verfahren zur Bestimmung von $x_{l,1}$, gewichtet diese jedoch anders.

Die Verfahrenskoeffizienten (α 's, β 's, δ 's) stehen bereits aus der Konsistenz-Analyse des compound-Schrittes fest. Es bleibt also nur noch die Aufgabe, die neuen Gewichte $b_i^L(\theta)$ zu bestimmen. Diese sind Funktionen in θ , die wir als Polynome vom Grade r schreiben (der Grad r hängt ab von der geforderten Ordnung p):

$$b_i^L(\theta) = \sum_{j=1}^r b_{ij}^L \cdot \theta^j \quad (3.19)$$

Um die Polynome $b_i^L(\theta)$ so zu bestimmen, dass die Genauigkeitsforderung (3.18) erfüllt ist, müssen wir nun eine neuerlich Taylorentwicklung anstrengen. $x_l(t_0 + \theta h_l)$ muss um den Entwicklungspunkt t_0 und $x_l^{\text{ds}}(\theta)$ um 0 entwickelt werden.

Hier hilft uns jedoch die Entwicklung aus der Analyse des compound-Schrittes:

- Die Entwicklung von $x_l(t_0 + \theta h_l)$ stimmt mit derjenigen von $x_l(t_0 + h_l)$ überein, mit der Ausnahme, dass wegen der Nachdifferentiation statt h_l^ν nun $\theta^\nu \cdot h_l^\nu$ steht.
- Die Entwicklung von $x_l^{\text{ds}}(\theta)$ stimmt mit derjenigen von $x_{l,1}$ überein, mit der Ausnahme, dass statt b_i^L nun $b_i^L(\theta)$ steht.

Deshalb können wir die zentrale Konsistenzaussage des compound-Schrittes (Theorem 3.16) auf den dense-output übertragen:

Theorem 3.19 *Das dense-output-Schema (3.14) ist von der Ordnung $\mathcal{O}(h_l^p)$, d.h. x_l^{ds} erfüllt (3.18) genau dann, wenn gilt:*

$$\sum_{i=1}^s b_i^L(\theta) \cdot \Phi_i(u^L) = \frac{\theta \varrho(u^L)}{\gamma(u^L)} \quad \forall \theta \in [0, 1] \quad (3.20)$$

$$\forall u^L \in \text{MDAT}_{x_l} \quad \text{mit} \quad \varrho(u^L) \leq p - 1$$

mit $\gamma(u^L), \Phi_i(u^L)$ aus Theorem 3.16.

Bem.:

Das besondere an dieser Bedingung ist, dass wir im dense-output eine Ordnung „nach unten“ gehen können. Wir müssen nicht wie im compound-Schritt, die Bedingungsgleichungen aller Bäume der Ordnungen $1, \dots, p$ erfüllen, sondern nur die der Bäume bis zur Ordnung $p - 1$.

Der Grund hierfür liegt in der Lipschitzbedingung der Funktion $\mathcal{W}(x_l, x_a)$ aus dem Beweis 3.18. Diese Lipschitzbedingung hängt wiederum ab von $\frac{\partial q_a}{\partial x_a}$ und dem Erfüllen einer Lipschitzbedingung von $w_a(x_l, x_a)$. Und dies unterstreicht einmal mehr die Forderung nach einer **schwachen Kopplung** zwischen latenten und aktiven Größen und einer **glatten Modellierung der Ströme**.

Wie man sich nun leicht klarmacht, führt Theorem 3.19 mit der speziellen Wahl der $b_i^L(\theta)$ als Polynome (3.19) durch einen Koeffizientenvergleich auf ein lineares Gleichungssystem:

$$\begin{pmatrix} \Phi_1(\bullet) & \dots & \Phi_s(\bullet) \\ \Phi_1(\bullet) & \dots & \Phi_s(\bullet) \\ \vdots & & \vdots \end{pmatrix} \cdot \begin{pmatrix} b_{11}^L & \dots & b_{1r}^L \\ b_{21}^L & \dots & b_{2r}^L \\ \vdots & & \vdots \\ b_{s1}^L & \dots & b_{sr}^L \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots \\ 0 & \frac{1}{2} & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix} \quad (3.21)$$

Dieses Gleichungssystem stellt nun die Bedingungsgleichungen für die dense-output-Gewichte $b_i^L(\theta)$ dar.

Wollen wir ein zum CHORAL-Verfahren passenden dense-output so konstruieren, dass die „Kombination“, das entstehende Index-1-dense-Verfahren also, die Konsistenzordnung 3 besitzt, so haben wir in obigem linearen Gleichungssystem nur die Bäume der Ordnungen 1 und 2 einzubeziehen (dies sind 4 Stück).

3.5 Konvergenz und Stabilität

Nach der Untersuchung der Konsistenz wenden wir uns nun der **Konvergenz** und der **Stabilität** des nichttransformierten Multirate-Verfahrens (2.18) zu. Die Konsistenzuntersuchung beschäftigte sich mit dem lokalen Diskretisierungsfehler, d. h. dem Fehler, den wir machen, wenn wir von einem exakten Anfangswert ausgehen und jeweils einen Schritt mit unserem Verfahren durchführen. Die Untersuchung der **Konvergenz** stellt die Frage, welchen Fehler wir am Ende des Integrationsintervalls, t_{end} gemacht haben werden. Denn das Integrationsintervall $[t_0, t_{end}]$ besteht mit Sicherheit aus mehreren Mikroschritten für den aktiven Teil und i. A. aus mehr als einem Makroschritt für den latenten Teil. In der **Stabilitätsanalyse** schliesslich untersuchen wir das Verhalten des konstruierten Multirate-Verfahrens für steife Probleme (zum Begriff der „Steifheit“ von Systemen siehe [HW96]).

Konvergenz

In der Konsistenzanalyse konnten wir Bedingungsgleichungen an die Verfahrenskoeffizienten herleiten, so dass der Fehler nach jeweils einem Schritt (lokaler Diskretisierungsfehler) in gewünschten Grenzen bleibt und bei Verfeinerung der Zeitdiskretisierung mit der gewünschten „Geschwindigkeit“ (Konsistenzordnung) gegen 0 strebt. Wir gingen hierzu jeweils von exakten Anfangswerten für jeden compound-Schritt und jeden späteren Mikroschritt aus.

Nun arbeitet aber bereits der erste der späteren Mikroschritte nicht mit dem exakten Anfangswert $x_a(t_0 + h_a)$ sondern mit der Näherung $x_{a,1}$. Es findet also eine Fehlerfortpflanzung von Schritt zu Schritt statt. Analoges gilt für die compound-Schritte: die Anfangswerte eines solchen Schrittes sind für den latenten Teil durch die Näherung des vorhergehenden compound-Schrittes und für den aktiven Teil durch den letzten der vorhergehenden späteren Mikroschritte gegeben.

Da diese jeweils bereits fehlerbehaftet sind, untersuchen wir im Folgenden den Fehler, den wir am Ende des Integrationsintervalls $[t_0, t_{end}]$ gemacht haben werden:

$$\|x_l(t_{end}) - x_{l,\nu}\| \quad \text{und} \quad \|x_a(t_{end}) - x_{a,\mu}\|$$

wobei wir von ν Schritten für den latenten Teil und μ Schritten für den aktiven Teil bis zum Erreichen von t_{end} ausgehen ($\nu = \#$ compound-Schritte, $\mu = \nu + \sum (\# \text{ spätere Mikroschritte pro compound-step})$).

Um nun nicht mehr ständig zwischen späteren Mikroschritten und compound-Schritten unterscheiden zu müssen, wollen wir alle Schritte, die für den aktiven Teil gemacht werden, als **Mikroschritte** bezeichnen und alle für den latenten als **Makroschritte**.

So wie wir den Begriff der Konsistenz für Multirate-Verfahren (s. S. 43, Def. 3.1) etwas anders definiert haben als für ODE-Verfahren, werden wir auch den Begriff der Konvergenz anpassen.

Definition 3.20 (Konvergenzordnung in der Schaltungssimulation)

Ein Multirate-Verfahren⁴ in der Schaltungssimulation besitzt die Konvergenzordnung p , falls gilt:

$$\begin{aligned} \|x_l(t_{end}) - x_{l,\nu}\| &= \mathcal{O}(h_{l,\max}^p) \quad \text{und} \\ \|x_a(t_{end}) - x_{a,\mu}\| &= \mathcal{O}(h_{a,\max}^p) \end{aligned}$$

mit $h_{l,\max} = \max\{\text{Makroschrittweiten } h_l\}$,
 $h_{a,\max} = \max\{\text{Mikroschrittweiten } h_a\}$,
 $\nu = \#$ (nötige Makroschritte) und
 $\mu = \#$ (nötige Mikroschritte).

Wir werden nun folgendes Lemma zeigen, das besagt, dass aus der Konsistenz die Konvergenz folgt

Lemma 3.21

Es gilt

$$\text{Konsistenz (Def. 3.1)} \Rightarrow \text{Konvergenz (Def. 3.20)}$$

falls das Schrittweitenverhältnis von Makroschrittweite zu Mikroschrittweiten in jedem compound-Schritt nicht zu groß wird.

Dieses Lemma werden wir nicht formal beweisen.

Wir werden lediglich auf die „Verwandtschaft“ zum entsprechenden Satz aus der Theorie der Einschrittverfahren für gewöhnliche Differentialgleichungen (Konsistenz impliziert Konvergenz) hinarbeiten.

Nach einer Aufteilung des Integrationsintervalls $[t_0, t_{end}]$ in mehrere compound-Schritte zeigen wir, dass der Fehler am Ende eines jeden compound-Schrittes im Latenten von der Größenordnung $\mathcal{O}(h_l^p)$ und im Aktiven von der Größenordnung $\mathcal{O}(h_{a,\max_{cp}})$ ist ($h_{a,\max_{cp}}$ ist die maximale Mikroschrittweite innerhalb eines Makroschrittes).

Da „unter“ einem Makroschritt mehrere Mikroschritte (einer im compound-Schritt und die aus den späteren Mikroschritten) liegen, entspricht dies einer

⁴mit einer Zweiteilung der Netzwerkgleichungen in aktive und latente Komponenten, sowie compound-Schritte und spätere Mikroschritte

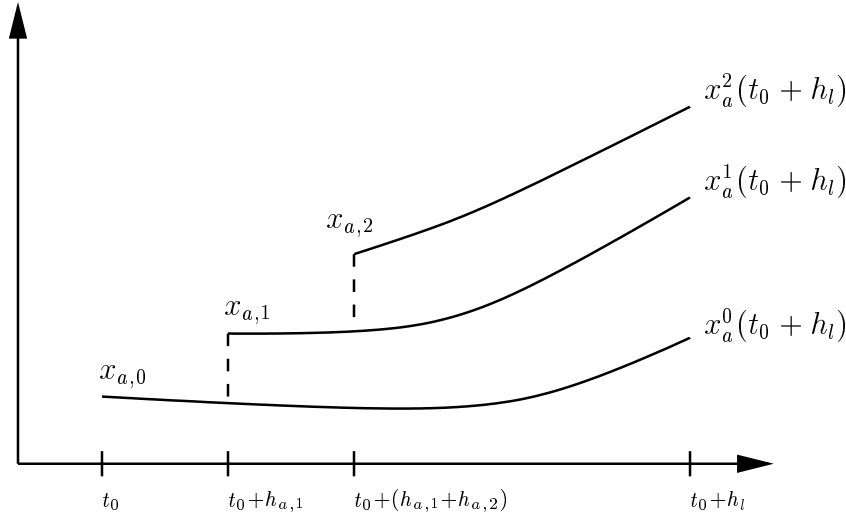


Abbildung 3.2: Fehlerfortpflanzung während der Simulation

Konvergenzordnung von $p + 1$ der Mikroschritte während eines Makroschrittes. D. h. in jedem einzelnen Makroschritt verlieren wir keine Ordnung im Aktiven.

Ist dies gezeigt, befinden wir uns in der gleichen Situation wie im ODE-Fall: wir haben statt zweier Diskretisierungen des Intervalls $[t_0, t_{end}]$ nur eine (durch die compound-Schritte). Hier werden wir abbrechen, denn wie bei Einschrittverfahren für gewöhnliche Differentialgleichungen (s. [HNW00]) könnten wir die entsprechenden Abschätzungen aus Lemma 3.21 zeigen.

Zerlegen wir also das Integrationsintervall in M Makroschritte der jeweiligen Länge $h_{l,j} = t_j - t_{j-1}$:

$$[t_0, t_{end}] = [t_0, t_1] \cup [t_1, t_2] \cup \dots \cup [t_{M-1}, t_M],$$

und greifen uns ein beliebiges dieser Intervalle heraus, oBdA. das erste $[t_0, t_1]$ mit der Länge $h_l \stackrel{\text{def}}{=} h_{l,1}$.

Dieses Makrointervall können wir nun wiederum zerlegen in η Mikrointervalle der jeweiligen Länge $h_{a,j} = \hat{t}_j - \hat{t}_{j-1}$:

$$[t_0, t_1] = [\hat{t}_0, \hat{t}_1] \cup [\hat{t}_1, \hat{t}_2] \cup \dots \cup [\hat{t}_{\eta-1}, \hat{t}_\eta]$$

Nun betrachten wir die Fehler am Ende des Makroschrittes:

- **für den latenten Teil:**

gerade der lokale Diskretisierungsfehler e_l mit

$$\|e_l\| = \|x_l(t_0 + h_l) - x_{l,1}\| \leq c_l \cdot h_l^{p+1}$$

• **für den aktiven Teil:**

Der Fehler e_a ergibt sich als „Summe“ der lokalen Diskretisierungsfehler der einzelnen Mikroschritte im Sinne von Abbildung 3.2.

Mit

- i) $x_{a,1}, x_{a,2}, \dots$ sind die aufeinanderfolgenden Approximationen der Mikroschritte.
- ii) $x_a = x_a^0$ ist die exakte Lösung mit Anfangswert $x_{a,0}$
- iii) $x_a^1(t), x_a^2(t), \dots$ sind die exakten Lösungen (deren Existenz vorausgesetzt !) zu den Anfangswerten $x_{a,1}, x_{a,2}, \dots$

Dann gilt für e_a mit $x_a^\eta(t_1) = x_{a,\eta}$

$$\|e_a\| = \|x_a(t_1) - x_{a,\eta}\| \leq \sum_{i=0}^{\eta-1} \|x_a^i(t_1) - x_a^{i+1}(t_1)\|$$

Für die „benachbarten“ Lösungen $x_a^i(t_1)$ und $x_a^{i+1}(t_1)$, sie unterscheiden sich nur im zugehörigen Anfangswert, liefert das „Fundamentallemma“ [HNW00, I.10, Theorem 10.2] für Differentialungleichungen:

$$\|x_a^i(t_1) - x_a^{i+1}(t_1)\| \leq \|x_a^i(\hat{t}_{i+1}) - x_{a,i+1}\| \cdot e^{\tilde{L}_2 \cdot (t_1 - \hat{t}_{i+1})}$$

($\tilde{L}_2$ ist die Lipschitzkonstante aus dem Beweis 3.18 zum Lemma über die Konsistenzordnung des Index-1-dense-Verfahrens.)

Nun ist $\|x_a^i(\hat{t}_{i+1}) - x_{a,i+1}\|$ gerade der lokale Diskretisierungsfehler im Mikroschritt mit der Schrittweite $h_{a,i+1}$, deshalb also:

$$\|x_a^i(\hat{t}_{i+1}) - x_{a,i+1}\| \leq c_{a,i+1} \cdot h_{a,i+1}^{p+1}$$

Dies führt nun für e_a auf:

$$\begin{aligned} \|e_a\| &\leq \sum_{i=0}^{\eta} c_{a,i+1} \cdot h_{a,i+1}^{p+1} \cdot e^{\tilde{L}_2 \cdot (t_1 - \hat{t}_{i+1})} \\ &\leq \left(\sum_{i=0}^{\eta} c_{a,i+1} \cdot e^{\tilde{L}_2 \cdot h_l} \right) \cdot h_{a,\max_{cp}}^{p+1} \\ &\leq c_a \cdot h_{a,\max_{cp}}^{p+1} \end{aligned}$$

Die letzte Abschätzung gilt nur dann, wenn zum einen $\tilde{L}_2$ genügend klein ist (schwache Kopplung !) und zum anderen das Verhältnis der Makroschrittweite h_l zu den Mikroschrittweiten $h_{a,j}$ nicht zu groß ist. Ansonsten könnte in dieser Abschätzung eine Ordnung verloren gehen.

An dieser Stelle sind wir in der gleichen Situation wie im Falle von ODE-Lösern. Mit den lokale Diskretisierungsfehlern nach einem Makro- bzw. compound-Schritt (e_a kann als solcher aufgefasst werden) verläuft der weitere Beweis wie im Falle von gewöhnlichen ODE-Lösern.

Stabilität

Im Gegensatz zur Konsistenz- und Konvergenzuntersuchung können wir für die **Stabilität** keine derart genauen Angaben machen.

Der Grund hierfür liegt darin, dass es noch keine der *Dahlquist'sche Testgleichung* entsprechende Referenzfunktion für differential-algebraische Probleme gibt.

Von Kværnø ([Kv]) stammt immerhin eine entsprechendes Referenzproblem zur Untersuchung von Multirate-Verfahren für ODE-Systeme (z. B. Netzwerkgleichungen in der Zustandsform). Dies und einige Grundbegriffe aus der Stabilitätsanalyse sollen hier kurz aufgeführt werden.

In der Theorie der gewöhnlichen Differentialgleichungen werden Probleme als **steif** bezeichnet, wenn explizite Verfahren (z. B. expliziter Euler), die darauf angesetzt werden, extrem kleine Schrittweiten benutzen, obwohl sich die Änderung kaum ändert, implizite Verfahren hier jedoch wesentlich zufriedenstellendere Ergebnisse liefern.

Wie man schnell sieht, ist die *Dahlquist'sche Testgleichung* :

$$\dot{y} = \lambda \cdot y \quad \lambda \in \mathbb{C}^- = \{z \in \mathbb{C} | \operatorname{Re}(z) \leq 0\} \quad (3.22)$$

für im Realteil stark negative λ

Die Stabilitätseigenschaften eines gewöhnlichen ODE-Lösers beschreiben seine „Eignung“ für seine Anwendung auf steife Probleme. Sie werden untersucht, indem man ihn auf die *Dahlquist'sche Testgleichung* (3.22) anwendet und nach einem Schritt mit der Schrittweite h eine numerische Approximation y_1 erhält:

$$y_1 = R(h\lambda) \cdot y_0$$

Die Funktion $R(h\lambda)$ wird **Stabilitätsfunktion** genannt. Aus mehreren Stabilitätsbegriffen seien nun zwei besonders wichtige herausgegriffen:

Definition 3.22 (A-Stabilität) Ein Verfahren heißt *A-stabil*, falls für seine Stabilitätsfunktion R gilt:

$$|R(z)| \leq 1 \quad \forall z \in \mathbb{C}^-$$

Definition 3.23 (L-Stabilität) Ein Verfahren heißt *L-stabil*, falls es *A-stabil* ist und zusätzlich gilt

$$\lim_{x \rightarrow \infty} R(z) = 0$$

Das beschriebene Vorgehen führt beispielsweise bei einem ROW-Verfahren auf die Stabilitätsfunktion (s. [HW96, IV.7, VI.4])

$$\begin{aligned} R(z) &= 1 + zb^T (I - zB)^{-1} \mathbb{1} \\ R(\infty) &= 1 - b^T B^{-1} \mathbb{1} \end{aligned}$$

Ein Multirate-Verfahren für ODEs, kann verständlicherweise nicht mit (3.22) hinsichtlich seiner Stabilität untersucht werden, sondern benötigt ein angepasstes Problem. Bei einer angenommenen Zweiteilung in latente und aktive Komponenten liegt eine Testfunktion der folgenden Form nahe:

$$\begin{pmatrix} \dot{y}_l \\ \dot{y}_a \end{pmatrix} = \Lambda \begin{pmatrix} y_l \\ y_a \end{pmatrix} \quad \Lambda = \begin{pmatrix} \lambda_{1,1} & \lambda_{1,2} \\ \lambda_{2,1} & \lambda_{2,2} \end{pmatrix} \in \mathbb{R}^{2 \times 2} \quad (3.23)$$

mit

$$\lambda_{1,1}, \lambda_{2,2} < 0 \quad \frac{\lambda_{1,2}\lambda_{2,1}}{\lambda_{1,1}\lambda_{2,2}} < 1$$

Diese Idee und das weitere Vorgehen, das hier nicht in aller Ausführlichkeit ausgearbeitet wird, ist zu finden bei Kværnø ([Kv]). Wenden wir ein Multirate-Verfahren für ODEs an auf die kopplungsfreie Testgleichung

$$\begin{pmatrix} \dot{y}_1 \\ \dot{y}_2 \end{pmatrix} = \begin{pmatrix} \lambda_{1,1} & 0 \\ 0 & \lambda_{2,2} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix},$$

so erhalten wir nach einem compound-Schritt

$$\begin{pmatrix} y_{l,1} \\ y_{a,1} \end{pmatrix} = \begin{pmatrix} R_l(\lambda_{1,1} \cdot h_l) \cdot y_{l,0} \\ R_a(\lambda_{2,2} \cdot h_a) \cdot y_{a,0} \end{pmatrix}.$$

R_l und R_a sind gerade die Stabilitätsfunktionen des Makro- und des Mikroschritt-Verfahrens.

Nach der Anwendung eines Multirate-Verfahrens auf die allgemeine Testgleichung (3.23) kann die hieraus gewonnene Stabilitätsfunktion durch die Stabilitätsfunktionen R_l und R_a der zugrundeliegenden Verfahren des latenten und aktiven Teils sowie den Größen $\lambda_{i,j}$ ausgedrückt werden. Es liegt hier die Vermutung nahe, dass sich das Gesamtverfahren bei schwacher Kopplung **wie die Einzelverfahren verhält**.

Wir werden also bei der Wahl der Koeffizienten darauf achten, dass die Stabilitätsfunktionen R_l und R_a der Einzelverfahren die Bedingung der Definition 3.22 (A-Stabilität) erfüllen. Tun sie dies, so sagen wir, das Multirate-Verfahren ist A-stabil (analog für L-Stabilität).

Kapitel 4

MDAE— ein Multirate-DAE-Verfahren

Nach den theoretischen Betrachtungen der Kapitel 2 und 3 darüber, wie ein Multirate-Verfahren für differential-algebraische Netzwerkgleichungen aussehen kann, und welche theoretischen Eigenschaften es besitzt, wollen wir in diesem Kapitel nun ein Multirate-Verfahren implementieren und testen.

Wie wir bereits festgehalten haben, führten wir die Untersuchungen am nicht-transformierten Multirate-Verfahren (2.18) durch, wollen aber das durch lineare Transformationen daraus gewonnene **Multirate-Verfahren für die partitionierte Netzwerkgleichung** (2.23) implementieren.

Dieses besitzt dieselben theoretischen Eigenschaften (Konsistenz, Konvergenz, Stabilität) wie das nichttransformierte, behandelt aber die allgemeinere Form der partitionierten Netzwerkgleichungen (2.10).

Ein letztes Mal benötigen wir das nichttransformierte Multirate-Schema (2.18), um die Koeffizienten für ein Multirate-Verfahren MDAE der Ordnung 3, mit eingebettetem Verfahren der Ordnung 2 zur Schrittweitensteuerung zu finden. Wir lösen die Bedingungsgleichungen der Tabelle 3.3, erhalten hieraus $b^\lambda, \hat{b}^\lambda, \mathcal{B}^\lambda, \mathcal{A}^\lambda, \mathcal{D}^\lambda$ und berechnen durch die Transformationen (2.19) die gesuchten Verfahrenskoeffizienten $d^\lambda, \hat{d}^\lambda, \mathcal{B}^\lambda, \sigma^\lambda, \varrho^\lambda$ ($\lambda = L, A$).

Nachdem wir **Koeffizienten** gefunden haben, verändern wir unser Verfahren derart, dass die Behandlung **nichtautonomer Netzwerkgleichungen** ermöglicht wird. Daraufhin wird die **Schrittweitensteuerung** und die damit verbundene Fehlerschätzung kurz erläutert. Schließlich führen wir einen **numerischen Test** durch, um zu sehen, ob sich die theoretischen Überlegungen „gelohnt“ haben, die Multirate-Eigenschaft also tatsächlich angemessen umgesetzt wird.

4.1 MDAE-Koeffizientensatz

Koeffizienten für den compound-Schritt

Es soll nun ein Verfahren der Ordnung 3 mit einem eingebetteten Verfahren der Ordnung 2 zur Schrittweitensteuerung und Fehlerkontrolle konstruiert werden. Wie beschrieben, berechnen wir zunächst Koeffizienten für das nichttransformierte Multirate-Verfahren (2.18) und erhalten durch die linearen Transformationen (2.19) Koeffizienten für das Multirate-Verfahren für die allgemeinere partitionierte Netzwerkgleichung (2.10).

Um die Bedingungsgleichungen aus Tabelle 3.3 zu erfüllen, würden für das übergeordnete Verfahren 4 Stufen und für das eingebettete Verfahren 3 Stufen reichen. Dies führt jedoch auf Verfahrenskoeffizienten, die nicht praktikabel sind (s. Anhang).

Deshalb seien für das übergeordnete Verfahren fünf, für das eingebettete (zur Schrittweitensteuerung) vier Stufen angesetzt:

$$\begin{aligned} x_{l,1} &= x_{l,0} + \sum_{i=1}^5 b_i^L \cdot k_i^L \\ x_{a,1} &= x_{a,0} + \sum_{i=1}^5 b_i^A \cdot k_i^A \\ \hat{x}_{l,1} &= x_{l,0} + \sum_{i=1}^4 \hat{b}_i^L \cdot k_i^L \\ \hat{x}_{a,1} &= x_{a,0} + \sum_{i=1}^4 \hat{b}_i^A \cdot k_i^A \end{aligned}$$

Nach Tabelle 3.3 stehen somit (das eingebettete Verfahren mit 8 zusätzlichen Unbekannten bei 8 Gleichungen mitgerechnet) 42 Gleichungen 80 Unbekannte gegenüber¹.

Diese Zahlen lassen sich durch einige Vereinfachungen und Überlegungen deutlich reduzieren.

- Zwischen latentem und aktivem Teil bestehe weitgehende Symmetrie:

$$\begin{aligned} b^L &= b^A \\ \hat{b}^L &= \hat{b}^A \\ \mathcal{A}^L &= \mathcal{A}^A \\ \mathcal{B}^L &= \mathcal{B}^A \\ \delta_i^L &= \delta_i^A \quad (i = 2, \dots, 5) \end{aligned}$$

¹die Abkürzungen $\alpha^L, \dots$ ausgeschrieben

- übergeordnetes und eingebettetes Verfahren seien *stiffly accurate*²:

$$\begin{aligned} b_i &= \beta_{5i} \quad (i = 1, \dots, 5) \\ \hat{b}_i &= \alpha_{5i} = \beta_{4i} \quad (i = 1, \dots, 4) \\ \alpha_5 &= \alpha_4 = 1 \end{aligned}$$

Diese Wahl bewirkt, dass die Stabilitätsfunktionen R_l, R_a des übergeordneten und $\hat{R}_l, \hat{R}_a$ des eingebetteten Verfahrens im Unendlichen verschwinden:

$$R_l(\infty) = R_a(\infty) = \hat{R}_l(\infty) = \hat{R}_a(\infty) = 0,$$

d. h. aus der A-Stabilität folgt L-Stabilität ([HW96, IV.3, Prop. 3.8, VI.4, Def. 4.11]).

Als zusätzlichen Effekt erhalten wir durch diese Wahl eine Reduktion der zu lösenden Bedingungsgleichungen, denn einige davon sind nun automatisch erfüllt.

- Für die Einkopplung der jeweils anderen Diskretisierungsebene gelte:

$$\begin{aligned} \delta_i^L &= \alpha_i^L \quad (i = 2, \dots, 5) \\ \delta_i^A &= \alpha_i^A \quad (i = 2, \dots, 5) \end{aligned}$$

Somit sind nurmehr folgende **Gleichungen** zu lösen:

(Bezeichnung: $\beta'_i = \beta_i - \gamma \quad (i = 1, \dots, 5)$)

$$\beta_{42}\beta'_2 + \beta_{43}\beta'_3 = \gamma^2 - 2\gamma + \frac{1}{2} \quad (4.1a)$$

$$\beta_{42}\alpha_2 + \beta_{43}\alpha_3 = \frac{1}{2} - \gamma \quad (4.1b)$$

$$\beta_{52}\beta'_2 + \beta_{53}\beta'_3 + \beta_{54}(1 - \gamma) = \gamma^2 - 2\gamma + \frac{1}{2} \quad (4.1c)$$

$$\beta_{52}\alpha_2 + \beta_{53}\alpha_3 + \beta_{54} = \frac{1}{2} - \gamma \quad (4.1d)$$

$$\beta_{52}\alpha_2^2 + \beta_{53}\alpha_3^2 + \beta_{54} = \frac{1}{3} - \gamma \quad (4.1e)$$

$$\beta_{54}\beta_{32}\beta'_2 + \beta_{54}(\gamma^2 - 2\gamma + \frac{1}{2}) = -\gamma^3 + 3\gamma^2 - \frac{3}{2}\gamma + \frac{1}{6} \quad (4.1f)$$

$$(\beta_{52}\gamma + \beta_{53}\beta_{32})\alpha_2 + \beta_{53}\gamma\alpha_3 + \frac{1}{2}\beta_{54} = \frac{1}{6} - \frac{1}{2}\gamma \quad (4.1g)$$

$$\begin{aligned} &\beta_{53}\beta'_2\delta_{32}^L + \beta_{54}\beta'_2\delta_{42}^L + \beta_{54}\beta'_3\delta_{43}^L \\ &+ \gamma\beta'_2\delta_{52}^L + \gamma\beta'_3\delta_{53}^L + (\gamma - \gamma^2)\delta_{54}^L = \frac{1}{6}m - \frac{1}{2}\gamma \end{aligned} \quad (4.1h)$$

²engl.: steif genau

$$\begin{aligned} & \beta_{53}\beta'_2\delta_{32}^A + \beta_{54}\beta'_2\delta_{42}^A + \beta_{54}\beta'_3\delta_{43}^A \\ & + \gamma\beta'_2\delta_{52}^A + \gamma\beta'_3\delta_{53}^A + (\gamma - \gamma^2)\delta_{54}^A = \frac{1}{6m} - \frac{1}{2}\gamma \end{aligned} \quad (4.1i)$$

$$\beta_{53}\alpha_2\delta_{32}^L + \beta_{54}\alpha_2\delta_{42}^L + \beta_{54}\alpha_3\delta_{43}^L + \gamma\alpha_2\delta_{52}^L + \gamma\delta_{54}^L = \frac{1}{6}m \quad (4.1j)$$

$$\beta_{53}\alpha_2\delta_{32}^A + \beta_{54}\alpha_2\delta_{42}^A + \beta_{54}\alpha_3\delta_{43}^A + \gamma\alpha_2\delta_{52}^A + \gamma\delta_{54}^A = \frac{1}{6m} \quad (4.1k)$$

$$\begin{aligned} & \beta_{53}\gamma\alpha_2^2\delta_{32}^L + \beta_{54}\gamma\alpha_2^2\delta_{42}^L + (\beta_{54}\gamma\alpha_3^2 - \beta_{32}\beta_{54}\alpha_2^2)\delta_{43}^L \\ & + \gamma^2\alpha_2^2\delta_{52}^L + (\gamma\alpha_3^2 - \beta_{32}\gamma\alpha_2^2)\delta_{53}^L \\ & + ((\beta_{32}\beta_{43} - \beta_{42}\gamma)\alpha_2^2 - \beta_{43}\gamma\alpha_3^2 + \gamma^2)\delta_{54}^L = \frac{1}{3}m\gamma^2 \end{aligned} \quad (4.1l)$$

$$\begin{aligned} & \beta_{53}\gamma\alpha_2^2\delta_{32}^A + \beta_{54}\gamma\alpha_2^2\delta_{42}^A + (\beta_{54}\gamma\alpha_3^2 - \beta_{32}\beta_{54}\alpha_2^2)\delta_{43}^A + \gamma^2\alpha_2^2\delta_{52}^A \\ & + (\gamma\alpha_3^2 - \beta_{32}\gamma\alpha_2^2)\delta_{53}^A \\ & + ((\beta_{32}\beta_{43} - \beta_{42}\gamma)\alpha_2^2 - \beta_{43}\gamma\alpha_3^2 + \gamma^2)\delta_{54}^A = \frac{1}{3m}\gamma^2 \end{aligned} \quad (4.1m)$$

Aufgrund der Dreiecksgestalt der Koeffizientenmatrix $\mathcal{B}^L = \mathcal{B}^A$ kann der Parameter γ als Faktor verstanden werden, der die Stabilität der „Einzelverfahren“ des latenten und aktiven Teils bestimmt. Setzt man

$$\gamma \stackrel{\text{def}}{=} 0.5728160624821349$$

so sind diese A-stabil(s. Anhang) und es ist darüberhinaus möglich

$$\alpha_2^L = \alpha_2^A = \delta_2^L = \delta_2^A = 0$$

zu setzen, was die Einsparung von Funktionsauswertungen mit sich bringt.

Es gibt einen Koeffizientensatz, der die Bedingungsgleichungen (4.1a- 4.1m) und die speziellen Forderungen an γ erfüllt. Die durch die lineare Transformation (2.19) hieraus entstehenden Koeffizienten $\hat{d}^\lambda, d^\lambda, \sigma^\lambda, \varrho^\lambda, \mathcal{B}^\lambda, (\lambda = L, A)$, die das Verfahren (2.23) bestimmen, sind in Tabelle (4.2) zusammengestellt.

Koeffizienten für die späteren Mikroschritte

Für die Berechnung der späteren Mikroschritte wird das CHORAL-Verfahren ([G98]) verwendet, in welches die latenten Komponenten via dense-output eingekoppelt werden sollen.

CHORAL erzielt mit vier Stufen die Konvergenzordnung 3, d. h. der lokale Diskretisierungsfehler ist von der Ordnung $\mathcal{O}(h^4)$. Nach Lemma 3.17 muss demnach der Fehler des dense-output in der Größenordnung $\mathcal{O}(h^3)$ liegen.

Um dies zu erfüllen stellen wir die Gewichtsfunktionen $b_i^L(\theta)$ als Polynome vom Grade 2 dar:

$$b_i^L(\theta) = b_{i1}^L \cdot \theta + b_{i2}^L \cdot \theta^2$$

Nach dem Theorem 3.19 zur Ordnung des dense-output müssen wir das lineares Gleichungssystem in den Polynomkoeffizienten $b_{11}^L, \dots, b_{52}^L$ zu lösen:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ \beta_1^L & \beta_2^L & \beta_3^L & \beta_4^L & \beta_5^L \\ 0 & \delta_2^L & \delta_3^L & \delta_4^L & \delta_5^L \\ 0 & \Psi_2^L & \Psi_3^L & \Psi_4^L & \Psi_5^L \end{pmatrix} \begin{pmatrix} b_{11}^L & b_{12}^L \\ b_{21}^L & b_{22}^L \\ b_{31}^L & b_{32}^L \\ b_{41}^L & b_{42}^L \\ b_{51}^L & b_{52}^L \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & \frac{1}{2} \\ 0 & \frac{1}{2} \\ 0 & 1 \end{pmatrix}$$

$$\text{mit } \Psi_i^L = \omega_{i2}^L(\alpha_2^L)^2 + \dots + \omega_{ii}^L(\alpha_i^L)^2 \quad (i = 2, \dots, 5)$$

In diesem linearen Gleichungssystem der Form (3.21) gehen die MDAT_{x_l} -Bäume der Ordnung 1 und 2 ein.

Wie man sieht, sind dies acht Gleichungen für zehn Unbekannte. D. h. zwei Unbekannte können frei gewählt werden:

$$b_{21}^L = b_{22}^L = 0$$

Da die Werte des dense-output in die Zuwächse des **CHORAL**-Verfahrens eingehen, benötigen wir keine dense-output-Gewichte $\hat{b}_i^L(\theta)$ für ein eingebettetes Verfahren.

Auch hier existieren Werte die obiges lineares Gleichungssystem erfüllen. Die bereits transformierten Werte und die **CHORAL**-Koeffizienten sind in Tabelle 4.2 aufgestellt.

4.2 Nichtautonome Probleme und Koeffizientensatz

Nun wollen wir das Multirate-Verfahren so anpassen, dass damit auch **nichtautonome Probleme der Form (4.2)** behandelt werden können.

latent: $A_l \cdot \dot{z}_l = f_l(x_l, x_a, t)$ $0 = z_l - q_l(x_l, t)$	aktiv: $A_a \cdot \dot{z}_a = f_a(x_l, x_a, t)$ $0 = z_a - q_a(x_a, t)$	(4.2)
---	--	-------

Dies geschieht durch Hinzunahme der trivialen Differentialgleichung $\dot{t} = 1$ und der Zeit t zum Vektor der Unbekannten. Dies hat natürlich keinen Einfluss auf die Genauigkeitsbedingungen. Das genaue Vorgehen ist im Anhang skizziert.

Durch die Betrachtung des Stabilitätsgebietes können wir zeigen, dass die Einzelverfahren des compound-Schrittes für die Wahl von $\gamma = 0.5728 \dots$ A-stabil und somit auch L-stabil sind.

Somit haben wir:

Theorem 4.1 *MDAE-Verfahren*

Das nachfolgende Multirate-Verfahren **MDAE** (4.3) stellt mit den Koeffizienten aus Tabelle 4.2 ein Multirate-Verfahren zur Lösung nichtautonomer Index-1-DAEs des Typ (4.2) dar.

Dieses Verfahren besitzt die Konsistenzordnung 3 mit einem eingebetteten Verfahren der Ordnung 2 zur Schrittweitensteuerung. Das dem latenten und aktiven Teil zugrundeliegende ROW-Verfahren ist A- und L-stabil.

compound-Schritt:

$$\begin{array}{l|l} x_{l,1} = x_{l,0} + \sum_{i=1}^s d_i^L \cdot \kappa_i^L & x_{a,1} = x_{a,0} + \sum_{i=1}^s d_i^A \cdot \kappa_i^A \\ \hat{x}_{l,1} = x_{l,0} + \sum_{i=1}^s \hat{d}_i^L \cdot \kappa_i^L & \hat{x}_{a,1} = x_{a,0} + \sum_{i=1}^s \hat{d}_i^A \cdot \kappa_i^A \end{array}$$

mit:

$$\begin{aligned} (A_l \frac{\partial q_l}{\partial x_l} - h_l \gamma^L \frac{\partial f_l}{\partial x_l}) \kappa_i^L &= A_l (q_l(x_{l,0}, t_0) - q_l(\tilde{a}_{L,i}, t_0 + \alpha_i^L h_l)) \\ &\quad + h_l \sum_{j=1}^i \beta_{ij}^L f_l(\tilde{a}_{L,j}, \tilde{d}_{L,j}, t_0 + \alpha_j^L h_l) + h_l \sum_{j=1}^{i-1} \beta_{ij}^L \frac{\partial f_l}{\partial x_l} \kappa_j^L \\ &\quad + h_l^2 \tau_i^L \frac{\partial f_l}{\partial t} - h_l \gamma_i^L A_l \frac{\partial q_l}{\partial t} \\ (A_a \frac{\partial q_a}{\partial x_a} - h_a \gamma^A \frac{\partial f_a}{\partial x_a}) \kappa_i^A &= A_a (q_a(x_{a,0}, t_0) - q_a(\tilde{a}_{A,i}, t_0 + \alpha_i^A h_a)) \\ &\quad + h_a \sum_{j=1}^i \beta_{ij}^A f_l(\tilde{d}_{A,i}, \tilde{a}_{A,i}, t_0 + \alpha_i^A h_a) + h_a \sum_{j=1}^{i-1} \beta_{ij}^A \frac{\partial f_a}{\partial x_a} \kappa_j^A \\ &\quad + h_a^2 \tau_i^A \frac{\partial f_a}{\partial t} - h_a \gamma_i^A A_a \frac{\partial q_a}{\partial t} \\ \tilde{a}_{\lambda,i} &= x_{\lambda,0} + \sum_{j=1}^{i-1} \sigma_{ij}^\lambda \kappa_j^\lambda \quad \tilde{d}_{\lambda,i} = x_{\bar{\lambda},0} + m^{\chi(\lambda)} \sum_{j=1}^{i-1} \varrho_{ij}^\lambda \kappa_j^{\bar{\lambda}} \end{aligned}$$

$$\begin{array}{ll} m = \frac{h_l}{h_a} & \{\lambda, \bar{\lambda}\} = \{L, A\} \\ \chi: \{L, A\} \rightarrow \{-1, 1\} & \chi(L) = 1 \quad \chi(A) = -1 \end{array}$$

spätere Mikroschritte:

$$\theta = \frac{1}{h_l} h_a, \frac{1}{h_l} (h_a + h_{\text{mc},1}), \dots, \underbrace{\frac{1}{h_l} (h_a + \dots + h_{\text{mc},\mu})}_1$$

$$x_{a,1+\nu} = x_{a,\nu} + \sum_{i=1}^s d_i^C \cdot \kappa_i^C \quad (\nu = 1, \dots, \mu)$$

$$\hat{x}_{a,1+\nu} = x_{a,\nu} + \sum_{i=1}^s \hat{d}_i^C \cdot \kappa_i^C \quad (\nu = 1, \dots, \mu)$$

mit:

$$\begin{aligned} (A_a \frac{\partial q_a}{\partial x_a} - h_l \gamma^L \frac{\partial f_a}{\partial x_a}) \kappa_i^C &= A_a (q_a(x_{a,\nu-1}, t_0) - q_a(\tilde{a}_{a,\nu}, t_0 +_{\text{mc}} \alpha_i^C)) \\ &+ h_{\text{mc},\nu} \sum_{j=1}^i \beta_{ij}^C f_a(x_l^{ds}(\theta), \tilde{a}_{C,i}, t_0 +_{\text{mc}} \alpha_i^C) + h_{\text{mc},\nu} \sum_{j=1}^{i-1} \beta_{ij}^C \frac{\partial f_a}{\partial x_a} \kappa_j^C \\ &+ h_{\text{mc}}^2 \tau_i^C \frac{\partial f_a}{\partial t} - h_{\text{mc}} \gamma_i^C A_a \frac{\partial q_a}{\partial t} \\ \tilde{a}_{C,i} &= x_{a,\nu} + \sum_{j=1}^{i-1} \sigma_{ij}^C \kappa_j^C \quad x_l^{ds}(\theta) = x_{l,0} + \sum_{j=1}^i d_i^L(\theta) \kappa_L^j \end{aligned}$$

MDAE-Verfahren

(4.3)

Die Approximationen an die exakte Lösung sind

- für den latenten Teil: $x_{l,1}$,
- für den aktiven Teil: $x_{a,1+\nu}$, $\nu = 0, \dots, \mu$

Mit diesen Werten wird jeweils weitergerechnet im nächsten Makro- bzw. Mikroschritt.

Zur Schrittweitensteuerung und Fehlerkontrolle werden die Approximationen der eingebetteten Verfahren benutzt

- für den latenten Teil: $\hat{x}_{l,1}$,
- für den aktiven Teil: $\hat{x}_{a,1+\nu}$, $\nu = 0, \dots, \mu$

Wie die Schrittweitensteuerung angesetzt ist, wird im nächsten Abschnitt erklärt.

Zunächst die Verfahrenskoeffizienten:

Tabelle 4.2: MDAE-Koeffizienten

compound-Schritt	
$\gamma^L = \gamma^A = 0.5728160624821349$	$\beta_{21}^L = \beta_{21}^A = 1$
$d_1^L = d_1^A = \hat{d}_1^L = \hat{d}_1^A = 2.0693903153031802444$	$\beta_{31}^L = \beta_{31}^A = 1.8880306002065536370$
$d_2^L = d_2^A = \hat{d}_2^L = \hat{d}_2^A = 0.20426363125484201319$	$\beta_{32}^L = \beta_{32}^A = -0.44456425453331691349$
$d_3^L = d_3^A = \hat{d}_3^L = \hat{d}_3^A = -0.5819203670527821464$	$\beta_{41}^L = \beta_{41}^A = 0.71651122219527387589$
$d_4^L = d_4^A = \hat{d}_4^L = \hat{d}_4^A = 1.0$	$\beta_{42}^L = \beta_{42}^A = -0.22576755996699538584$
$d_5^L = d_5^A = 1.0$	$\beta_{43}^L = \beta_{43}^A = -0.063559724710413390053$
$\alpha_2^L = \alpha_2^A = 0.0$	$\beta_{51}^L = \beta_{51}^A = -0.31214675842986220340$
$\alpha_3^L = \alpha_3^A = 1.1456321249642698000$	$\beta_{52}^L = \beta_{52}^A = 2/3$
$\alpha_4^L = \alpha_4^A = 1.0$	$\beta_{53}^L = \beta_{53}^A = -0.99895604626306480089$
$\alpha_5^L = \alpha_5^A = 1.0$	$\beta_{54}^L = \beta_{54}^A = 1.0716200755441254375$
$\tau_1^L = \tau_1^A = 0.32811824143753707377$	$\gamma_1^L = \gamma_1^A = \gamma$
$\tau_2^L = \tau_2^A = 1.4737503664018068738$	$\gamma_2^L = \gamma_2^A = 1.5728160624821349$
$\tau_3^L = \tau_3^A = 0.88099892093718541473$	$\gamma_3^L = \gamma_3^A = 0.87065028319110182351$
$\tau_4^L = \tau_4^A = 0.0$	$\gamma_4^L = \gamma_4^A = 0$
$\tau_5^L = \tau_5^A = 0.0$	$\gamma_5^L = \gamma_5^A = -\gamma$
$\sigma_{31}^L = \sigma_{31}^A = -0.3967214617379743755$	
$\sigma_{32}^L = \sigma_{32}^A = 0.87288055057917321998$	
$\sigma_{41}^L = \sigma_{41}^A = \sigma_{51}^L = \sigma_{51}^A = 2.0693903153031802445$	
$\sigma_{42}^L = \sigma_{42}^A = \sigma_{52}^L = \sigma_{52}^A = 0.20426363125484201301$	
$\sigma_{43}^L = \sigma_{43}^A = \sigma_{53}^L = \sigma_{53}^A = -0.58192036705278214648$	
$\sigma_{54}^L = \sigma_{54}^A = 1.0$	
$\varrho_{31}^L = \varrho_{31}^A = -1.1956286156506325007$	
$\varrho_{32}^L = \varrho_{32}^A = 1.1638407341055642933$	
$\varrho_{41}^L = -2.4749831147425456100 - 0.43549629019110712979 \cdot m$	
$\varrho_{41}^A = -2.4749831147425456100 - 0.43549629019110712979/m$	
$\varrho_{42}^L = 1.5896064250307480252 + 0.11874443097228464219 \cdot m$	
$\varrho_{42}^A = 1.5896064250307480252 + 0.11874443097228464219/m$	
$\varrho_{43}^L = -0.094697535173801056646 + 0.072010683301616068636 \cdot m$	
$\varrho_{43}^A = -0.094697535173801056646 + 0.072010683301616068636/m$	
$\varrho_{51}^L = 8.2842818056624845970 - 1.5040856873323504931 \cdot m$	
$\varrho_{51}^A = 8.2842818056624845970 - 1.5040856873323504931/m$	
$\varrho_{52}^L = -1.6136038399240553432 + 0.37692236293441198975 \cdot m$	
$\varrho_{52}^A = -1.6136038399240553432 + 0.37692236293441198975/m$	
$\varrho_{53}^L = -1.3868572374099122996 + 0.30865997463013855963 \cdot m$	
$\varrho_{53}^A = -1.3868572374099122996 + 0.30865997463013855963/m$	
$\varrho_{54}^L = \varrho_{54}^A = 1.1638407341055642933$	

spätere Mikroschritte

$\gamma^C = 0.5728160624821349$	$\beta_{21}^C = -2.0302139317498051$
$d_1^C = \hat{d}_1^C = \sigma_{21}^C = \sigma_{31}^C = \sigma_{41}^C = 1/\gamma$	$\beta_{31}^C = 0.2707896390839690$
$d_2^C = \hat{d}_2^C = \sigma_{32}^C = \sigma_{42}^C = 0.0$	$\beta_{32}^C = 0.1563942984338961$
$d_3^C = \hat{d}_3^C = \sigma_{43}^C = 1.0$	$\beta_{41}^C = 2/3$
$d_4^C = 1.0$	$\beta_{42}^C = 0.08757666432971973$
$\alpha_2^C = 1.0$	$\beta_{43}^C = -0.3270593934785213$
$\alpha_3^C = 1.0$	$\gamma_1^C = \gamma$
$\alpha_4^C = 1.0$	$\gamma_2^C = -2.457397870$

$$\begin{aligned}
\tau_1^C &= 0.3281182414375370 & \gamma_3^C &= 0.0 \\
\tau_2^C &= -2.57057612180719 & \gamma_4^C &= 0.0 \\
\tau_3^C &= -0.229210360916031 \\
\tau_4^C &= 1/6
\end{aligned}$$

dense-output

$$\begin{aligned}
d_1^L(\theta) &= -3.55895607418976124 \cdot \theta^2 + 6.36789149915394148 \cdot \theta \\
d_2^L(\theta) &= 1.07709809132388922 \cdot \theta^2 - 0.7901086922765544185 \cdot \theta \\
d_3^L(\theta) &= 0.395738717040728160 \cdot \theta^2 - 1.61366155722255233 \cdot \theta \\
d_4^L(\theta) &= 0.272488923071582123 \cdot \theta^2 + 1.20078325501518202 \cdot \theta \\
d_5^L(\theta) &= 0.0
\end{aligned}$$

4.3 Schrittweitensteuerung und Implementierung

Wir wollen nun noch einige Details zu Implementierung und insbesondere zur Schrittweitensteuerung von MDAE darlegen.

Eine erste Implementierung des entwickelten MDAE-Algorithmus wurde für `matlab` geschrieben und ist auf der beigefügten CD enthalten.

Um die jeweiligen Schrittweiten anpassen zu können, muss der Fehler abgeschätzt werden, den wir während eines Schrittes gemacht haben.

Zur **Fehlerschätzung** wird die Differenz zwischen den Werten des übergeordneten und eingebetteten Verfahrens benutzt. Diese Differenz wird skaliert mit der Matrix, die im Verfahren selbst die Zuwächse bestimmt. Für den latenten Teil liest sich dies beispielsweise als:

$$\begin{aligned}
\mathbf{err}_l &= (A_l \frac{\partial q_l}{\partial x_l} - h_l \gamma_l \frac{\partial f_l}{\partial x_l})^{-1} (x_{l,1} - \hat{x}_{l,1}) \\
\text{mit } \hat{x}_{l,1} &= x_{l,0} + \sum_{i=1}^4 \hat{d}_i^L \kappa_i^L
\end{aligned}$$

Dies ist ohne großen Mehraufwand zu bestimmen, zumal die LR-Zerlegung der Skalierungsmatrix aus dem Verfahren bereits vorliegt. Die Idee dieser Skalierung stammt von Shampine und wird bei der Behandlung steifer Differentialgleichungen eingesetzt (s. z. B. [HW96, IV.8]).

Der skalierte Fehler wird nun normiert. In diese Normierung gehen relative und absolute Fehlertoleranzen ein (s. [HNW00, II.4]). Mit der normierten Größe wird ein **Schrittweitevorschlag** errechnet.

Ist der normierte Fehler größer als 1, so wird die Approximation verworfen und der Schritt mit der vorgeschlagenen Schrittweite erneut durchgeführt. Andernfalls wird die Näherung akzeptiert und im nächsten Schritt mit der vorgeschlagenen Schrittweite gerechnet. Ein compound-Schritt wird also nur dann abgelehnt, wenn

der Fehler des Makroschrittes oder der des zugehörigen Mikroschrittes die gegebenen Toleranzen übersteigt.

Um ein zu schnelles Wachstum der Schrittweiten zu verhindern, wird mit einem Sicherheitsfaktor $f_{\text{fac}} \leq 1$ die Wachstumsgeschwindigkeit beschränkt.

Wie wir an mehreren Stellen (Konvergenz, spätere Mikroschritte) gesehen haben, müssen wir das **Verhältnis von Makro- zu Mikroschrittweiten** beschränken. Dies ist folgendermaßen realisiert:

- Im compound-Schritt wird die große Schrittweite h_l so gewählt, dass sie nicht mehr als 10-mal so groß wie die Schrittweite h_a wird,
- in den späteren Mikroschritten werden die Mikroschrittweiten h_{mc} gegebenenfalls so nach oben korrigiert, dass sie nicht kleiner als der zweihundertste Teil der Makroschrittweite h_l sind,
- der jeweils letzte Schrittweitemvorschlag der Mikroschritte wird als neue Schrittweite h_a im compound-Schritt verwandt.

Diese Verhältnisse (10 im compound Schritt, 200 in den späteren Mikroschritten) wurden gewählt, da sie sich im numerischen Test am besten bewährt haben (minimale Zahl abgelehnter Schritte). Bei großen Dimensionsunterschieden zwischen latentem und aktivem Teil ist eine Reduktion der Integrationsarbeiten in Bezug auf den latenten Teil um den Faktor 10 bereits recht gut.

Der angesprochene Sicherheitsfaktor f_{fac} wurde aus demselben Grund auf 0.8 gesetzt.

MDAE ist dazu bestimmt worden, nichtautonome partitionierte Netzwerkgleichungen des Typs (4.2) approximativ zu lösen. Also müssen die gestellten Probleme in der technischen Implementierung einige Benennungs-Standards erfüllen.

Als **Eingabeparameter** verlangt vorliegende MDAE-Implementierung einen `matlab`-function-handler auf die zu berechnende Index-1-DAE, die Anfangswerte für latenten und aktiven Teil sowie das Simulationsintervall in der Form $[t_{\text{start}}, t_{\text{end}}]$ oder t_{end} (im letzteren Fall wird t_{start} automatisch auf 0 gesetzt).

Bei der Eingabe-DAE wird von einer festen Partitionierung in aktive und latente Bestandteile ausgegangen. Zur Übergabe an das Verfahren muss sie derart aufbereitet sein, dass sie in Form einer Struktur mit folgenden Elementen vorliegt:

- A_l, A_a : Inzidenzmatrizen des latenten bzw. aktiven Teils
- f_l, f_a : rechte Seiten des differentiellen Teils der DAE (function-handler)
- q_l, q_a : algebraische Bedingungen (function-handler)

Die **Rückgabewerte** von MDAE sind zum einen zwei Vektoren Tl, Ta , in denen die Zeitdiskretisierung des Simulationsintervalls auf latenter bzw. aktiver Ebene

festgehalten ist und zum anderen zwei Felder Xl, Xa deren Spalten der jeweiligen Approximation von latentem und aktivem Anteil entsprechen. Zusätzlich werden in einer Struktur Angaben zum Aufwand der Integration festgehalten.

Die verwendeten Hilfsfunktionen werden im Anhang kurz beschrieben.

4.4 Numerischer Test

Die Funktionalität von MDAE kann zunächst nur an Testgleichungen untersucht werden, bei denen die Aktivitätsgrade während der gesamten Simulationsdauer konstant bleiben, d.h. Lösungskomponenten sind immer „aktiv“ oder immer „latent“.

Hierzu bietet sich die von Kværnø für Multirate-Verfahren angepasste **Prothero-Robinson**-Testgleichung an (s. [BG01]):

$$\dot{y}(t) = \begin{pmatrix} \Upsilon & \varepsilon \\ \varepsilon & -1 \end{pmatrix} \begin{pmatrix} y_1(t) - \sin(t) \\ y_2(t) - \sin(\omega t) \end{pmatrix} + \begin{pmatrix} \cos(t) \\ \omega \cdot \cos(\omega t) \end{pmatrix}, \quad y(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

In der Schreibweise des allgemeinen nichtautonomen Systems (4.2) also:

$$\begin{aligned} 1 \cdot \dot{z}_l(t) &= \Upsilon \cdot (x_l(t) - \sin(t)) + \varepsilon \cdot (x_a(t) - \sin(\omega t)) + \cos(t) \\ 0 &= z_l(t) - x_l(t) \\ 1 \cdot \dot{z}_a(t) &= \varepsilon \cdot (x_l(t) - \sin(t)) - 1 \cdot (x_a(t) - \sin(\omega t)) + \omega \cdot \cos(\omega t) \\ 0 &= z_a(t) - x_a(t) \\ x_l(0) &= 0 \\ x_a(0) &= 0 \end{aligned}$$

Bem.:

- i) Da der Cosinus nicht linear ist, stellt die Prothero-Robinson-Gleichung ein nichtautonomes Problem dar.
- ii) Für stark negative Υ wird dieses System steif, ε steuert die Kopplung (bei betragsmäßig kleinem ε schwach) und ω bestimmt die Aktivität des aktiven Teils im Vergleich zum latenten.

Wie leicht einzusehen ist, ist die eindeutige Lösung gegeben durch

$$\begin{aligned} x_l(t) &= \sin(t) \\ x_a(t) &= \sin(\omega t) \end{aligned}$$

Tabelle (4.3) stellt den Aufwand der Simulation durch das neue Verfahren MDAE dem des, in MATLAB implementierten Verfahrens `ode23s` gegenüber.

Hier wurden die Funktionsparameter wie folgt gewählt:

$$\Upsilon = -10^4, \quad \varepsilon = 0.1, \quad \omega = 100$$

Das Integrationsintervall war $[0, \pi]$. Die Schrittweitensteuerung arbeitete mit den Toleanzen $rtol = 10^{-3}$, $atol = 10^{-6}$ und im Falle von MDAE mit dem Sicherheitsfaktor $fac = 0.8$

Methode	MDAE	ode23s
compound-Schritte	199(0)	
Mikro-Schritte	2693(588)	1499(243)
Funktionsauswertungen		
f_l	796	7984
q_l	796	
f_a	13920	
q_a	13332	
LU-Zerlegungen	3679	1742
Minimale Schrittweiten		
$\min\{h_a\}$	$9.359 \cdot 10^{-6}$	$3.262 \cdot 10^{-4}$
$\min\{h_t\}$	$3.142 \cdot 10^{-4}$	

Tabelle 4.3: Vergleich MDAE mit ode23s

Die Angaben in Klammern entsprechen der Anzahl der jeweils abgelehnten Schritte.

Für die Funktionen f_l und g_l konnte nach Tabelle (4.3) eine sehr große Einsparung an Funktionsauswertungen erzielt werden. Im Falle einer Berechnung mit ode23s wurden diese immerhin rund zehnmal so oft aufgerufen.

Im Prothero-Robinson Beispiel sind die Dimensionen des latenten und aktiven Teil gleich, deshalb schlägt die gestiegene Zahl der Funktionsauswertungen im Aktiven in der Rechenzeit zu Buche.

Bei einem geeigneten Beispiel, einem Beispiel also, das $\dim(x_a) \ll \dim(x_t)$ erfüllt, würde voraussichtlich die Einsparung von ca. 90% der Auswertungen der latenten Funktionen zu einer deutlichen Reduktion der Rechenzeit im Vergleich zum ode23s führen.

Erfreulich ist auch, dass mit den beschriebenen Einschränkungen der Schrittweitenverhältnisse die Anzahl der abgelehnten compound-Schritte auf 0 reduziert werden konnte. Auch die Anzahl abgelehnter Mikroschritte ist zufrieden-

stellend, steht er doch zu den angenommenen im (fast) gleichen Verhältnis wie beim `ode23s`-Löser.

Zusammenfassung und Ausblick

In der industriellen Schaltungsentwicklung ist es unumgänglich, die Funktionsweise elektrischer Schaltwerke vor deren technischer Umsetzung zu überprüfen. Hierzu werden Netzwerke auf ein adäquates mathematisches Modell abgebildet. Die so entstehenden Netzwerkgleichungen ermöglichen die Simulation der zugrundeliegenden Schaltung.

In der vorliegenden Arbeit wurde ein Multirate-Verfahren zur approximativen Berechnung differential-algebraischer Netzwerkgleichung entwickelt.

Zuerst wurden die Grundlagen der mathematischen Modellbildung elektrischer Schaltungen vorgestellt. Es wurde aufgezeigt, dass industrielle Simulationstools elektrische Schaltungen auf differential-algebraische Netzwerkgleichungen abbilden.

Neben dieser analytischen Eigenschaft besitzen Netzwerkgleichungen die Multirate-Eigenschaft, d. h. es gibt sich schnell ändernde Lösungskomponenten und solche, die weniger Dynamik zeigen. Multirate-Verfahren nutzen diese Aktivitätsdifferenzen und berechnen langsamere Komponenten mit größeren und aktivere mit kleineren Schrittweiten.

Das entwickelte Multirate-Verfahren ermöglicht es, die Multirate-Eigenschaft auch in der Berechnung differential-algebraischer Netzwerkgleichungen auszunutzen. Der Schwerpunkt in dieser Verfahrensentwicklung und Beschreibung lag in der Untersuchung der Konsistenz, aus welcher auch auf die Konvergenz geschlossen werden konnte. Daneben wurde ein Weg aufgezeigt, den Begriff der Stabilität numerischer Integrationsschematas auf Multirate-Verfahren für differential-algebraische Systeme zu übertragen.

In der Konsistenzanalyse konnte ein einfacher und anschaulicher Automatismus aufgebaut werden, um Bedingungen an die Verfahrenskoeffizienten aufstellen zu können, die zu einer gewünschten Konsistenzordnung führen. Hierzu war die Idee maßgeblich, Differentiale in der Taylorentwicklung mit graphischen Objekten, „Bäumen“, zu verbinden. Bäume als Repräsentanten von Differentialen zu benutzen, ist eine Idee aus dem Bereich numerischer Integratoren für gewöhnliche Differentialgleichungen. Diese Idee wurde nun durch eine gänzlich neue Klasse von Bäumen an das Problem partitionierter Netzwerkgleichungen angepasst.

Schließlich konnte MDAE, ein Multirate-Verfahren der Konvergenzordnung 3, in

der `matlab`-Umgebung implementiert und getestet werden. Der numerische Test lieferte vielversprechende Ergebnisse. So konnte bereits bei dem recht einfachen Beispiel des *Prothero-Robinson*-Problems die Anzahl der Funktionsauswertungen des weniger aktiven (latenten) Teils um ca. 90 % reduziert werden.

MDAE verfügt noch über keine dynamische Partitionierung, benötigt also Probleme, bei denen die Zuteilung zum aktiven oder latenten Part statisch ist. Aus der Schaltungsentwicklung zeichnet sich jedoch eine Problemklasse ab, die genau dies erfüllt. In Schaltkreisen, die mit Transistoren auf Basis der noch jungen SOI-Technik (**S**ilicon **O**n **I**nsulator³) bestückt sind, gibt es eine solche statische Partitionierung. Bei klassischen MOSFET-Transistoren liegen die Transistorteile direkt auf der Siliciumplatte auf und sind fest verdrahtet. SOI-Transistoren liegen auf einer dünnen Silicium-Dioxid-Schicht, haben somit keinen direkten Kontakt zur tragenden Siliciumplatte. Die Dioxid-Schicht fungiert als Isolator. Sie verhindert das Abdriften von Elektronen in die Siliciumplatte. Sie erhöht somit die Effizienz des Transistors und senkt den Energieverbrauch der gesamten Schaltung. In der SOI-Realisierung von IBM ist der getragene Transistor nicht fest mit seiner Umgebung verbunden, sondern „schwimmt“ gewissermaßen auf der Silicium-Dioxid-Schicht. Dieses Schwimmen und die Leitfähigkeit des Transistors beeinflussen sich gegenseitig. Als Maß für die Beweglichkeit des Transistors auf dem Isolator-Kissen kann seine Temperatur genommen werden, die Leitfähigkeit beschreibt die Beziehung zwischen Spannungsabfall und Strom. Da die Temperatur eine sich langsam ändernde Größe ist, Spannungen jedoch im Mega-Hertz Bereich liegen, hat man hier ein sehr deutliches Multirate-Potential. Dies wird in Zukunft in Zusammenarbeit mit der *infineon technologies AG* untersucht werden.

Für die klassischen Techniken aus der Elektronik wird eine dynamische Partitionierungsstrategie von Interesse sein und eine Übertragung auf differential-algebraische Probleme höherer Ordnung.

³engl.: Insulator=Isolator

Anhang

Im Anhang wollen wir nun einige Sachverhalte zusammenstellen, die den Lesefluss behindert hätten:

A) Beweis zur Differentiation der Zuwächse.....	94
B) Differentiale und Bäume der Ordnungen 1,2,3	99
C) Übertragung auf nicht-autonome Systeme.....	102
D) Koeffizienten für ein 4-stufiges Verfahren.....	103
E) MDAE-Hilfsfunktionen.....	104

A) Beweis zur Differentiation der Zuwächse

Hier sei der vollständige Beweis zum Lemma 3.12 (S. 58) dargestellt.

Beweis 4.2

Durch vollständige Induktion nach p am oBdA. latenten Fall k_i^L

Induktionsstart $p = 1$

Der einzige Baum $u^L \in \text{MDAT}_{x_l}$ mit $\varrho(u) = 1$ ist τ_{x_l} mit

$$\begin{aligned}\gamma(\tau_{x_l}) &= 1 \\ \Phi_i(\tau_{x_l}) &= 1 \\ F(\tau_{x_l}) &= \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l\end{aligned}$$

Es ist also zu zeigen: $(k_i^L)^{(1)}(0) = \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l$

Nach (3.10) gilt: $(k_i^L)^{(1)}(0) = \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} (l_i^L)^{(1)}(0)$

$$\text{mit: } (l_i^L)^{(1)}(0) = w_l + \frac{\partial w_l}{\partial x_l} \sum_{j=1}^i \gamma_{ij}^L \underbrace{k_j^L(0)}_{=0} = w_l$$

Induktionsschritt:

Die Behauptung (auch die bezgl. des aktiven Teils) gelte bereits für alle Ableitungen der Ordnung $\leq p - 1$

Dann ist:

$$\begin{aligned}
& \left(k_i^L\right)^{(p)}(0) = \\
& \sum_{\substack{\mu_1+\dots+\mu_l=p \\ l \geq 2}} \sum_{j, \nu_1, \dots, \nu_l} \omega_{ij}^L \cdot \alpha_{j\nu_1}^L \cdot \dots \cdot \alpha_{j\nu_l}^L \cdot (-1) \cdot \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} \frac{\partial^l g_l}{\partial x_l^L} \left(\left(k_{\nu_1}^L\right)^{(\mu_1)}, \dots, \left(k_{\nu_l}^L\right)^{(\mu_l)} \right) \\
& + p \cdot \sum_{\substack{\mu_1+\dots+\mu_l+\nu_1+\dots+\nu_n \\ =p-1 \\ l+n \geq 2}} \sum_{\substack{\lambda_1, \dots, \lambda_l, \\ \xi_1, \dots, \xi_n}} \alpha_{i\lambda_1}^L \cdot \dots \cdot \alpha_{i\lambda_l}^L \cdot \delta_{i\xi_1}^L \cdot \dots \cdot \delta_{i\xi_n}^L \cdot m^n \left(\frac{1}{m}\right)^{\nu_1+\dots+\nu_n} \\
& \quad \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} \frac{\partial^{l+n} w_l}{\partial x_l^l \partial x_a^n} \left(\left(k_{\lambda_1}^L\right)^{(\mu_1)}, \dots, \left(k_{\xi_n}^A\right)^{(\nu_n)} \right) \\
& + p \cdot \sum_{j=1}^i \beta_{ij}^L \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} \frac{\partial w_l}{\partial x_l} \left(k_j^L\right)^{(p-1)} + m \cdot \left(\frac{1}{m}\right)^{p-1} \cdot p \cdot \sum_{j=1}^i \delta_{ij}^L \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} \frac{\partial w_l}{\partial x_l} \left(k_j^A\right)^{(p-1)}
\end{aligned}$$

$$\begin{aligned}
& \boxed{\text{I.H.}} \\
& \left(k_i^L\right)^{(p)}(0) = \sum_{\substack{\mu_1+\dots+\mu_l=p \\ l \geq 2}} \sum_{j, \nu_1, \dots, \nu_l} \omega_{ij}^L \cdot \alpha_{j\nu_1}^L \cdot \dots \cdot \alpha_{j\nu_l}^L \cdot \\
& \quad (-1) \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} \frac{\partial^l g_l}{\partial x_l^L} \left(\sum_{\substack{u_1^L \in \text{LMDAT}_{x_l} \\ \varrho(u_1^L) = \mu_1}} \gamma(u_1^L) \cdot \Phi_{\nu_1}(u_1^L) \cdot F(u_1^L)(x_{l,0}, x_{a,0}), \dots \right. \\
& \quad \left. \sum_{\substack{u_l^L \in \text{LMDAT}_{x_l} \\ \varrho(u_l^L) = \mu_l}} \gamma(u_l^L) \cdot \Phi_{\nu_l}(u_l^L) \cdot F(u_l^L)(x_{l,0}, x_{a,0}) \right) \\
& + \sum_{\substack{\mu_1+\dots+\mu_l+\nu_1+\dots+\nu_n \\ =p-1 \\ l+n > 2}} p \cdot \sum_{\substack{\lambda_1+\dots+\lambda_l+\xi_1+\dots+\xi_n}} \alpha_{i\lambda_1}^L \cdot \dots \cdot \alpha_{i\lambda_l}^L \cdot \delta_{i\xi_1}^L \cdot \dots \cdot \delta_{i\xi_n}^L \cdot m^n \cdot \left(\frac{1}{m}\right)^{\nu_1+\dots+\nu_n} \\
& \quad \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} \frac{\partial^{l+n} w_l}{\partial x_l^l \partial x_a^n} \left(\sum_{\substack{u_1^L \in \text{LMDAT}_{x_l} \\ \varrho(u_1^L) = \mu_1}} \gamma(u_1^L) \cdot \Phi_{\lambda_1}(u_1^L) \cdot F(u_1^L)(x_{l,0}, x_{a,0}), \dots \right. \\
& \quad \sum_{\substack{u_l^L \in \text{LMDAT}_{x_l} \\ \varrho(u_l^L) = \mu_l}} \gamma(u_l^L) \cdot \Phi_{\lambda_l}(u_l^L) \cdot F(u_l^L)(x_{l,0}, x_{a,0}), \\
& \quad \sum_{\substack{u_1^A \in \text{LMDAT}_{x_a} \\ \varrho(u_1^A) = \nu_1}} \gamma(u_1^A) \cdot \Phi_{\xi_1}(u_1^A) \cdot F(u_1^A)(x_{l,0}, x_{a,0}), \dots \\
& \quad \left. \sum_{\substack{u_n^A \in \text{LMDAT}_{x_a} \\ \varrho(u_n^A) = \nu_n}} \gamma(u_n^A) \cdot \Phi_{\xi_n}(u_n^A) \cdot F(u_n^A)(x_{l,0}, x_{a,0}) \right)
\end{aligned}$$

$$\begin{aligned}
& + p \cdot \sum_{j=1}^i \beta_{ij}^L \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \frac{\partial w_l}{\partial x_l} \sum_{\substack{u_1^L \in \text{LMDAT}_{x_l} \\ \varrho(u_1^L) = p-1}} \gamma(u_1^L) \cdot \Phi_j(u_1^L) \cdot F(u_1^L)(x_{l,0}, x_{a,0}) \\
& + p \cdot m \cdot \left(\frac{1}{m} \right)^{p-1} \sum_{j=1}^i \delta_{ij}^L \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \frac{\partial w_l}{\partial x_a} \sum_{\substack{u_1^A \in \text{LMDAT}_{x_a} \\ \varrho(u_1^A) = p-1}} \gamma(u_1^A) \cdot \Phi_j(u_1^A) \cdot F(u_1^A)(x_{l,0}, x_{a,0}) \\
& \stackrel{\boxed{\Rightarrow}}{\left(k_i^L \right)^{(p)}(0)} = \\
& \sum_{\substack{u_1^L, \dots, u_l^L \in \text{LMDAT}_{x_l} \\ \varrho(u_1^L) + \dots + \varrho(u_l^L) = p \\ l \geq 2}} \gamma(u_1^L) \cdot \dots \cdot \gamma(u_l^L) \sum_{j, \nu_1, \dots, \nu_l} \omega_{ij}^L \cdot \alpha_{j\nu_1}^L \cdot \dots \cdot \alpha_{j\nu_l}^L \cdot \Phi_{\nu_1}(u_1^L) \cdot \dots \cdot \Phi_{\nu_l}(u_l^L) \\
& \quad (-1) \cdot \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \frac{\partial^l g_l}{\partial x_l^l} \left(F(u_1^L), \dots, F(u_l^L) \right) \\
& + \sum_{\substack{u_1^L, \dots, u_l^L \in \text{LMDAT}_{x_l} \\ u_1^A, \dots, u_n^A \in \text{LMDAT}_{x_a} \\ \varrho(u_1^L) + \dots + \varrho(u_l^L) + \\ \varrho(u_1^A) + \dots + \varrho(u_n^A) = p-1 \\ l+n \geq 2}} p \cdot \gamma(u_1^L) \cdot \gamma(u_l^L) \cdot \gamma(u_1^A) \cdot \dots \cdot \gamma(u_n^A) \\
& \quad \sum_{\substack{\lambda_1, \dots, \lambda_l \\ \xi_1, \dots, \xi_n}} \alpha_{i\lambda_1}^L \cdot \dots \cdot \alpha_{i\lambda_l}^L \cdot \delta_{i\xi_1}^L \cdot \dots \cdot \delta_{i\xi_n}^L \cdot m^n \cdot \left(\frac{1}{m} \right)^{\varrho(u_1^A) + \dots + \varrho(u_n^A)} \\
& \quad \cdot \Phi_{\lambda_1}(u_1^L) \cdot \dots \cdot \Phi_{\lambda_l}(u_l^L) \cdot \Phi_{\xi_1}(u_1^A) \cdot \dots \cdot \Phi_{\xi_n}(u_n^A) \\
& \quad \cdot \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \frac{\partial^{l+n} w_l}{\partial x_l^l \partial x_a^n} \left(F(u_1^L), \dots, F(u_n^A) \right) \\
& + \sum_{\substack{u_1^L \in \text{LMDAT}_{x_l} \\ \varrho(u_1^L) = p-1}} p \cdot \gamma(u_1^L) \cdot \sum_{j=1}^i \beta_{ij}^L \cdot \Phi_j(u_1^L) \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \frac{\partial w_l}{\partial x_l} F(u_1^L)(x_{l,0}, x_{a,0}) \\
& + \sum_{\substack{u_1^A \in \text{LMDAT}_{x_a} \\ \varrho(u_1^A) = p-1}} p \cdot \gamma(u_1^A) \cdot \sum_{j=1}^i m \cdot \left(\frac{1}{m} \right)^{p-1} \delta_{ij}^L \cdot \Phi_j(u_1^A) \left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \frac{\partial w_l}{\partial x_a} F(u_1^A)(x_{l,0}, x_{a,0})
\end{aligned}$$

Nun gilt:

- i) • $-\left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \frac{\partial^l g_l}{\partial x_l^l} \left(F(u_1^L), \dots, F(u_l^L) \right) = F([u_1^L, \dots, u_l^L]_{x_l})$
- $\gamma(u_1^L) \cdot \dots \cdot \gamma(u_l^L) = \gamma([u_1^L, \dots, u_l^L]_{x_l})$
- $\sum_{j, \nu_1, \dots, \nu_l} \omega_{ij}^L \cdot \alpha_{j\nu_1}^L \cdot \dots \cdot \alpha_{j\nu_l}^L \cdot \Phi_{\nu_1}(u_1^L) \cdot \dots \cdot \Phi_{\nu_l}(u_l^L) = \Phi_i([u_1^L, \dots, u_l^L]_{x_l})$

$[u_1^L, \dots, u_l^L]_{x_l}$ entsteht aus den Bäumen $u_1^L, \dots, u_l^L$, indem man deren Wurzeln durch jeweils einen Ast mit einem gemeinsamen leeren runden Knoten verbindet.

Daraus folgt sofort: $\varrho([u_1^L, \dots, u_l^L]_{x_l}) = \varrho(u_1^L) + \dots + \varrho(u_l^L)$

- ii) • $\left(\frac{\partial g_l}{\partial x_l} \right)^{-1} \frac{\partial^{l+n} w_l}{\partial x_l^l \partial x_a^n} \left(F(u_1^L), \dots, F(u_l^L), F(u_1^A), \dots, F(u_n^A) \right)$

$$\begin{aligned}
&= F([u_1^L, \dots, u_l^L, u_1^A, \dots, u_n^A]_{y_l}]_{x_l}) \\
&= F([t_1^L]_{x_l}) \text{ mit } t_1^L = [u_1^L, \dots, u_l^L, u_1^A, \dots, u_n^A]_{y_l} \\
&\bullet \gamma(t_1^L) = \varrho(t_1^L) \cdot \gamma(u_1^L) \cdot \dots \cdot \gamma(u_l^L) \cdot \gamma(u_1^A) \cdot \dots \cdot \gamma(u_n^A) \\
&\bullet \sum_{\substack{\lambda_1, \dots, \lambda_l \\ \xi_1, \dots, \xi_n}} \alpha_{i\lambda_1}^L \cdot \alpha_{i\lambda_l}^L \cdot \dots \cdot \delta_{i\xi_1}^L \cdot \dots \cdot \delta_{i\xi_n}^L \cdot m^n \cdot \left(\frac{1}{m}\right)^{\varrho(u_1^A) + \dots + \varrho(u_n^A)} \\
&\quad \Phi_{\lambda_1}(u_1^L) \cdot \dots \cdot \Phi_{\lambda_l}(u_l^L) \cdot \Phi_{\xi_1}(u_1^A) \cdot \dots \cdot \Phi_{\xi_n}(u_n^A) \\
&= \Phi_i([t_1^L]_{x_l})
\end{aligned}$$

$t_1^L = [u_1^L, \dots, u_l^L, u_1^A, \dots, u_n^A]_{y_l}$ entsteht aus den Bäumen $u_1^L, \dots, u_l^L, u_1^A, \dots, u_n^A$, indem man deren Wurzeln durch jeweils einen Ast mit einem gemeinsamen vollen runden Knoten verbindet.

Daraus folgt: $\varrho(t_1^L) = \varrho(u_1^L) + \dots \varrho(u_n^A) + 1$

$[t_1^L]_{x_l}$ entsteht nun noch, indem man der Wurzel von t_1^L einen leeren runden Knoten voranstellt. Damit $\varrho([t_1^L]_{x_l}) = \varrho(t_1^L)$

$$\begin{aligned}
iii) \quad &\bullet \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} \frac{\partial w_l}{\partial x_l} F(u_1^L) \\
&= F([u_1^L]_{y_l}]_{x_l}) \\
&= F([t_1^L]_{x_l}) \text{ mit } t_1^L = [u_1^L]_{y_l} \\
&\bullet \gamma([t_1^L]_{x_l}) = \gamma(t_1^L) = \varrho(t_1^L) \cdot \gamma(u_1^L) \\
&\bullet \sum_{j=1}^i \beta_{ij}^L \Phi_j(u_1^L) = \Phi_i([t_1^L]_{x_l})
\end{aligned}$$

Wie oben ergibt sich auch hier: $\varrho([u_1^L]_{y_l}]_{x_l}) = \varrho(u_1^L) + 1$

$$\begin{aligned}
iv) \quad &\bullet \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} \frac{\partial w_l}{\partial x_a} F(u_1^A) \\
&= F([u_1^A]_{y_l}]_{x_l}) \\
&= F([t_1^L]_{x_l}) \text{ mit } t_1^L = [u_1^A]_{y_l} \\
&\bullet \gamma([t_1^L]_{x_l}) = \gamma(t_1^L) = \varrho(t_1^L) \cdot \gamma(u_1^A) \\
&\bullet \sum_{j=1}^i m \left(\frac{1}{m}\right)^{p-1} \cdot \beta_{ij}^L \Phi_j(u_1^A) = \Phi_i([t_1^L]_{x_l})
\end{aligned}$$

Wie oben ergibt sich auch hier: $\varrho([u_1^A]_{y_l}]_{x_l}) = \varrho(u_1^A) + 1$

Damit folgt nun:

$$\begin{aligned}
(k_i^L)^{(p)}(0) &= \sum_{\substack{u^L = [u_1^L, \dots, u_l^L]_{x_l} \\ \varrho(u^L) = p \\ l \geq 2}} \gamma(u^L) \cdot \Phi_i(u^L) \cdot F(u^L)(x_{l,0}, x_{a,0}) \\
&+ \sum_{\substack{u^L = [u_1^L, \dots, u_l^L, \\ u_1^A, \dots, u_n^A]_{y_l}]_{x_l} \\ \varrho(u^L) = p \\ l+n \geq 2}} \gamma(u^L) \cdot \Phi_i(u^L) \cdot F(u^L)(x_{l,0}, x_{a,0})
\end{aligned}$$

$$\begin{aligned}
& + \sum_{\substack{u^L = [[u_1^L]_{y_l}]_{x_l} \\ \varrho(u^L) = p}} \gamma(u^L) \cdot \Phi_i(u^L) \cdot F(u^L)(x_{l,0}, x_{a,0}) \\
& + \sum_{\substack{u^L = [[u_1^A]_{y_l}]_{x_l} \\ \varrho(u^L) = p}} \gamma(u^L) \cdot \Phi_i(u^L) \cdot F(u^L)(x_{l,0}, x_{a,0})
\end{aligned}$$

Aufgrund der Konstruktionsvorschriften der Bäume taucht jeder Baum $\in \text{LMDAT}$ genau einmal in genau einer der obigen Summen auf, da sich die Mengen, über die die jeweiligen Summen gebildet werden, nicht überschneiden. Somit folgt, fasst man die Summen zusammen:

$$\begin{aligned}
\left(k_i^L\right)^{(p)}(0) &= \sum_{\substack{u^L \in \text{LMDAT}_{x_l} \\ \varrho(u^L) = p}} \gamma(u^L) \cdot \Phi_i(u^L) \cdot F(u^L)(x_{l,0}, x_{a,0}) \\
&= \sum_{\substack{u^L \in \text{MDAT}_{x_l} \\ \varrho(u^L) = p}} \alpha(u^L) \cdot \gamma(u^L) \cdot \Phi_i(u^L) \cdot F(u^L)(x_{l,0}, x_{a,0})
\end{aligned}$$

□

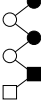
B) Differentiale und Bäume der Ordnungen 1,2,3

Zu den Bedingungsgleichungen der Koeffizientenbestimmung im compound-Schritt hier nun eine Aufstellung aller Differentiale bis zur Ordnung 3:

u_{x_l}	$\alpha(u_{x_l})$	$F(u_{x_l})$
	1	$\left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l$
	1	$-\left(\frac{\partial g_l}{\partial x_l}\right)^{-1} \frac{\partial^2 g_l}{\partial x_l \partial x_l} \left(\left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l, \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l \right)$
	1	$\left(\frac{\partial g_l}{\partial x_l}\right)^{-1} \frac{\partial w_l}{\partial x_l} \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l$
	1	$\left(\frac{\partial g_l}{\partial x_l}\right)^{-1} \frac{\partial w_l}{\partial x_a} \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a$
	1	$\left(\frac{\partial g_l}{\partial x_l}\right)^{-1} \frac{\partial^2 w_l}{\partial x_l \partial x_l} \left(\left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l, \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l \right)$
	2	$\left(\frac{\partial g_l}{\partial x_l}\right)^{-1} \frac{\partial^2 w_l}{\partial x_l \partial x_a} \left(\left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l, \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a \right)$
	1	$\left(\frac{\partial g_l}{\partial x_l}\right)^{-1} \frac{\partial^2 w_l}{\partial x_a \partial x_a} \left(\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a, \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a \right)$
	1	$\left(\frac{\partial g_l}{\partial x_l}\right)^{-1} \frac{\partial w_l}{\partial x_l} \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} \frac{\partial w_l}{\partial x_l} \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l$
	1	$\left(\frac{\partial g_l}{\partial x_l}\right)^{-1} \frac{\partial w_l}{\partial x_l} \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} \frac{\partial w_l}{\partial x_a} \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a$

[illegible]

Tabelle 4.5: Elementare Differentiale, latenter Teil

u_{x_a}	$\alpha(u_{x_a})$	$F(u_{x_a})$
	1	$\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a$
	1	$-\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} \frac{\partial^2 g_a}{\partial x_a \partial x_a} \left(\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a, \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a \right)$
	1	$\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} \frac{\partial w_a}{\partial x_a} \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a$
	1	$\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} \frac{\partial w_a}{\partial x_l} \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l$
	1	$\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} \frac{\partial^2 w_a}{\partial x_a \partial x_a} \left(\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a, \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a \right)$
	2	$\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} \frac{\partial^2 w_a}{\partial x_l \partial x_a} \left(\left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l, \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a \right)$
	1	$\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} \frac{\partial^2 w_a}{\partial x_l \partial x_l} \left(\left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l, \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l \right)$
	1	$\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} \frac{\partial w_a}{\partial x_a} \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} \frac{\partial w_a}{\partial x_a} \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a$
	1	$\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} \frac{\partial w_a}{\partial x_a} \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} \frac{\partial w_a}{\partial x_l} \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l$
	1	$\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} \frac{\partial w_a}{\partial x_l} \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} \frac{\partial w_l}{\partial x_l} \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l$

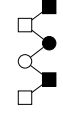
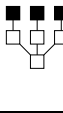
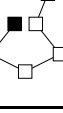
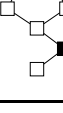
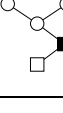
	1	$\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} \frac{\partial w_a}{\partial x_l} \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} \frac{\partial w_l}{\partial x_a} \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a$
	3	$-\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} \frac{\partial^2 g_a}{\partial x_a \partial x_a} \left(\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a, \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} \frac{\partial w_a}{\partial x_a} \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a \right)$
	3	$-\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} \frac{\partial^2 g_a}{\partial x_a \partial x_a} \left(\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a, \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} \frac{\partial w_a}{\partial x_l} \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l \right)$
	1	$-\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} \frac{\partial^3 g_a}{\partial x_a \partial x_a \partial x_a} \left(\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a, \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a, \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a \right)$
	3	$\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} \frac{\partial^2 g_a}{\partial x_a \partial x_a} \left(\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a, \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} \frac{\partial^2 g_a}{\partial x_a \partial x_a} \left(\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a, \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a \right) \right)$
	1	$-\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} \frac{\partial w_a}{\partial x_a} \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} \frac{\partial^2 g_a}{\partial x_a \partial x_a} \left(\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a, \left(\frac{\partial g_a}{\partial x_a}\right)^{-1} w_a \right)$
	1	$-\left(\frac{\partial g_a}{\partial x_a}\right)^{-1} \frac{\partial w_a}{\partial x_l} \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} \frac{\partial^2 g_l}{\partial x_l \partial x_l} \left(\left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l, \left(\frac{\partial g_l}{\partial x_l}\right)^{-1} w_l \right)$

Tabelle 4.6: Elementare Differentiale, aktiver Teil

C) Übertragung auf nicht-autonome Systeme

Die Anpassung von (2.23) an nicht-autonome Systeme (am Beispiel des latenten Parts)

$$\begin{aligned} A_l \dot{z}_l &= f_l(x_l, x_a, t) \\ 0 &= z_l(t) - q(x_l, t) \end{aligned} \tag{4.4}$$

geschieht durch Einfaches Hinzufügen der trivialen Differentialgleichung $\dot{t} = 1$.

Nach Abschnitt 2.2 ist 4.4 im wesentlichen äquivalent zu:

$$\begin{aligned}\dot{y}_l &= w_l(x_l, x_a, t) \\ \dot{x}_l &= \frac{1}{\varepsilon}(y_l - g_l(x_l, t)) \quad (\varepsilon \rightarrow 0) \\ \dot{t} &= 1\end{aligned}$$

Ein ROW-Ansatz liefert hier:

$$\begin{pmatrix} y_{l,1} \\ x_{l,1} \\ t_1 \end{pmatrix} = \begin{pmatrix} y_{l,0} \\ x_{l,0} \\ t_0 \end{pmatrix} + \sum_{i=1}^s b_i^L \begin{pmatrix} l_i^L \\ k_i^L \\ \lambda_i^L \end{pmatrix}$$

mit $\begin{pmatrix} l_i^L \\ k_i^L \\ \lambda_i^L \end{pmatrix} = h_l \cdot \begin{pmatrix} w_l(\dots, \dots, t_0 + \sum_{j=1}^{i-1} \alpha_{ij}^L \lambda_j^L) \\ \frac{1}{\varepsilon}(y_{l,0} + \sum_{j=1}^{i-1} \alpha_{ij}^L l_j^L - g_l(\dots, t_0 + \sum_{j=1}^{i-1} \alpha_{ij}^L \lambda_j^L)) \\ 1 \end{pmatrix}$

$$+ h_l \cdot \begin{pmatrix} 0 & \frac{\partial w_l}{\partial x_l} & \frac{\partial w_l}{\partial t} \\ \frac{1}{\varepsilon} \mathcal{I} & -\frac{1}{\varepsilon} \frac{\partial g_l}{\partial x_l} & -\frac{1}{\varepsilon} \frac{\partial g_l}{\partial t} \\ 0 & 0 & 0 \end{pmatrix} \cdot \sum_{j=1}^i \gamma_{ij}^L \begin{pmatrix} l_i^L \\ k_i^L \\ \lambda_i^L \end{pmatrix}$$

hieraus folgt $\lambda_i^L = h_l$ ($i = 1, \dots, s$) und mit

$$\tau_i^L = \sum_{j=1}^i \beta_{ij}^L \gamma_j^L$$

liefert analoges Vorgehen wie in Abschnitt 2.2 (Gauß, Transformationen, ...) schließlich:

$$\begin{aligned}(A_l \frac{\partial q_l}{\partial x_l} - h_l \gamma^L \frac{\partial f_l}{\partial x_l}) &= A_l(q_l(x_0, t_0) - q_l(\tilde{a}_i^L, t_0 + \alpha_i^L h_l)) \\ &+ h_l \sum_{j=1}^i \beta_{ij}^L f_l(\tilde{a}_j^L, \tilde{d}_j^L, t_0 + \alpha_j^L h_l) \\ &+ h_l \sum_{j=1}^{i-1} \beta_{ij}^L \frac{\partial f_l}{\partial x_l} \kappa_j^L + h_l^2 \tau_i^L \frac{\partial f_l}{\partial t} - h_l \gamma_i^L A_l \frac{\partial q_l}{\partial t}\end{aligned}$$

D) Koeffizienten für ein 4-stufiges Verfahren

Prinzipiell ist ein Verfahren der Ordnung 3 mit einem eingebetteten Verfahren der Ordnung 2 mit 4 Stufen möglich. Dies ist in dem Sinne möglich, dass es Koeffizienten gibt, die die Bedingungsgleichungen aus Tabelle 3.3 erfüllen.

Es zeigt sich, dass es nur ein A-stabiles Verfahren

$$\gamma^L = \gamma^A = 0.351937430897529974$$

gibt.

Dies führt jedoch zu einer unverhältnismäßigen Überbetonung der Einkopplung der jeweils anderen Aktivitätsschicht. Deutlich wird dies an einer Gegenüberstellung der transformierten Koeffizienten, die die Einkopplung der jeweils anderen Aktivitätsebene steuern, und denen, die das jeweils eigene Aktivitätsniveau „rückkoppeln“ (hier am Beispiel des latenten Teils):

$$\begin{aligned}
\varrho_{2,1}^L &= 6.51309895282093711 & \sigma_{2,1}^L &= 3.60918244283132180 \\
\varrho_{3,1}^L &= 652069.203733206331 - 265889.967904378941 \cdot m & \sigma_{3,1}^L &= 1.30791589060016350 \\
\varrho_{3,2}^L &= -604103.699903255889 + 246333.345988884038 \cdot m & \sigma_{3,2}^L &= 1.42070708058779884 \\
\varrho_{4,1}^L &= -769933.328123503219 + 313953.698826270808 \cdot m & \sigma_{4,1}^L &= 1.30791589060016350 \\
\varrho_{4,2}^L &= 713306.220557359842 - 290861.914524259854 \cdot m & \sigma_{4,2}^L &= 1.42070708058779884 \\
\varrho_{4,3}^L &= 3.70087463765145700 - 1.52011170471001065 \cdot m & \sigma_{4,3}^L &= 1
\end{aligned}$$

Dieser Unterschied in den Größenordnungen zeigte auch in der Implementierung seine Auswirkungen. Dort konnten mit diesem Koeffizientensatz keine befriedigenden Ergebnisse erzielt werden.

E) MDAE-Hilfsfunktionen

Bei der Implementation des vorgestellten MDAE-Algorithmus sind einige Hilfsfunktionen vonnöten, die nun hier kurz erwähnt sein sollen.

1. `num_jacobi(f,i,varargin)`

bestimmt mittels dividierter Differenzen eine Näherung der Jacobi Matrix $\frac{\partial f}{\partial x_i}$ der Funktion f an der Stelle $(x_1, x_2, \dots, x_n)$. Hierbei darf die Anzahl der Funktionsparameter beliebig sein die Werte $x_1, \dots, x_n$ müssen Spaltenvektoren oder Skalare sein.

2. `horner(a,x)`

berechnet nach dem Hornerschema den Wert des Polynoms $a_1 \cdot x^n + a_2 \cdot x^{n-1} + \dots + a_{n-1} \cdot x + a_n$ für den Spaltenvektor $a = (a_1, \dots, a_n)$ und dem Zeilenvektor $x \in \mathbb{R}^k$

3. `dense_output (x0,d_phi,Kappa,phi)`

berechnet den dense-output gemäss: $x_{ds} = x_0 + \sum d_i(\phi) \cdot \kappa_i$

4. `[h_prop,err]=step_control (L,U,h,x,p,xd,pd,rtol,atol)`

berechnet nach der Standard-Schrittweitensteuerung ([HNW00, II.4]) und dem skalierten Fehlerschätzer $e = (L \cdot U)^{-1}(x - xd)$ den normierten Fehler err und einen Schrittweitevorschlag h_prop .

x, xd sind zwei Näherungen der Ordnung p bzw. pd , h die hierzu benutzte Schrittweite, $rtol$ und $atol$ sind optionale Argumente für relative und absolute Toleranz. Ist $err \leq 1$, so wird die vorliegende Näherung angenommen um im nächsten Schritt mit h_prop weitergerechnet. Andernfalls wird der aktuelle Schritt mit h_prop wiederholt.

Literaturverzeichnis

- [BG01] A. Bartel, M. Günther
A Multirate W-Method for Electrical Networks in State-Space Formulation
Preprint Nr. 01/16, IWRMM Universität Karlsruhe
wird erscheinen im J. Comp. Appl. Math
- [BGK] A. Bartel, M. Günther and A. Kværnø
Multirate Methods in Electrical Circuit Simulation
- [G98] M. Günther
Simulating digital circuits numerically – a charge-oriented ROW approach
Numer. Math. (1998) 79: 203-212
- [GHR00] M. Günther, M. Hoschek and P. Rentrop
Differential-Algebraic Equations in Electric Circuit Simulation
Int. J. Electron. Commun. (AEÜ) 54 (2000) No.2, 101-107
- [GKP99] M. Günther, A. Kværnø and P. Rentrop
Multirate Partitioned Runge-Kutta Methods
BIT-Num. Math. Vol. 41 No. 4 pages 504-514
- [GR93] M. Günther and P. Rentrop
Multirate ROW methods and latency of electric circuits
Appl. Num. Math. 13 (1993),83-102
- [GR00] M. Günther and P. Rentrop
Numerical Simulation of Electrical Circuits
In GAMM-Mitteilungen 23: 1-2, pages 51-77, 2000
- [H81] E. Hairer
Order Conditions for Numerical Methode for Partitioned Ordinary Differential Equations
Numer.Math. 36,431-445(1981)

- [HNW00] E. Hairer, S.P. Nørsett, G. Wanner
Solving Ordinary Differential Equations I: Nonstiff Problems, Second Revised Edition
Springer Series in Computational Mathematics 8, Springer-Verlag, Berlin 2000
- [HW96] E. Hairer and G. Wanner
Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems, Second Revised Edition
Springer Series in Computational Mathematics 14, Springer-Verlag, Berlin 1996
- [KR99] A. Kværnø and P. Rentrop
Low Order Multirate Runge-Kutta Methods in Electric Circuit Simulation
Preprint Nr. 99/1, IWRMM Universität Karlsruhe
- [Kv] A. Kværnø
Stability of Multirate Runge-Kutta Schemes
Department of Mathematical Sciences,
Norwegian University of Sciences, Trondheim
- [Ro88] M. Roche
Rosenbrock Methods for Differential Algebraic Equations
Numer. Math. (1988) 52: 45-63
- [RSW96] P. Rentrop, K. Strehmel, R. Weiner
Ein Überblick über Einschrittverfahren zur numerischen Integration in der technischen Simulation
GAMM-Mitteilungen 1996, Heft 1, 9-43
- [S01] H. Stoll
Ein adaptives Waveletverfahren in der Schaltungssimulation
Diplomarbeit, Universität Karlsruhe 10/2001
- [SFR91] B. Simeon, C. Führer, P. Rentrop
Differential-algebraic equations in vehicle systems dynamics
Surv. Math. Ind. (1991) I:1-37

Danksagung

Bei Herrn PD Dr. Michael Günther bedanke ich mich dafür, dass er mir ein Thema angeboten hat, das von Anfang an sehr interessant war, dafür, dass er mir viel Geduld und Vertrauen entgegengebracht hat und mir stets mit Rat und Tat half, wo es etwas zu helfen gab. Er ermöglichte mir auch, durch ein Praktikum bei der Firma *infineon technologies* erste Einblicke zu erhalten, wo und wie numerische Simulationsmethoden entwickelt und eingesetzt werden.

Stellvertretend für alle, die mir bei *infineon technologies* helfend zur Seite standen, bedanke ich mich bei Herrn Dr. Feldmann und Herrn Dr. Schultz in München.

Danken möchte ich auch Herrn Andreas Bartel vom Institut für Wissenschaftliches Rechnen und Mathematische Modellbildung (IWRMM) in Karlsruhe für die Diskussion in Karlsruhe und die bereitwillige Beschaffung von Informationsmaterialien zum Thema.

Herrn Prof. Dr. Forst aus Ulm sei gedankt für die Bereitschaft, die Aufgabe des Zweitberichters zu übernehmen.

Mein besonderer Dank gebührt meiner Familie, meinen Eltern und meinem Bruder, da sie mir stets den Rücken gestärkt haben. Meinen Eltern möchte ich ganz besonders danken, da sie mir das Studium erst ermöglicht haben.

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst und dabei keine als die angegebenen Hilfsmittel benutzt habe.

Leutkirch, den 28. Mai 2002

Michael Striebel