Universität Leipzig

Fakultät für Mathematik und Informatik

Mathematisches Institut

Automatisches Differenzieren und minimal erweiterte Systeme zur Berechnung singulärer Punkte

Diplomarbeit
(korrigierte Fassung)

vorgelegt von

Stefan Gille
geboren am 09. Juli 1986

Studiengang Mathematik-Diplom

Leipzig, Oktober 2012

Betreuender Hochschullehrer: Prof. Dr. Peter Kunkel

Mathematisches Institut, Fakultät für Mathematik und Informatik,
Universität Leipzig

Inhaltsverzeichnis

Notationen **i**

Liste verwendeter Symbole i

1 Automatisches Differenzieren **1**

1.1 Einleitung 1

1.2 Vorwärtsmethode 7

1.2.1 Aufwand und Fehlerabschätzung 9

1.3 Höhere Ableitungen 12

2 Ableitung iterativer Verfahren **13**

2.1 Einleitung 13

2.2 Voraussetzungen 13

2.3 Konvergenzbegriffe 19

2.4 Konvergenz des Basisverfahrens 21

2.5 Vollständig differenziertes Verfahren 24

2.6 Vereinfacht differenziertes Verfahren 25

2.7 Konvergenz der Ableitungen 26

2.8 Nachträglich differenzierte Verfahren 31

2.9 Numerische Beispiele 31

2.9.1 Polarkoordinaten 32

2.9.2 Ljapunow-Schmidt-Reduktion 33

3 Grundlagen der Singularitäten-Theorie **35**

3.1 Begriff der reduzierten Funktion 35

3.2 Kontaktäquivalenz 39

3.3 Klassifikation 44

3.3.1 Skalare Singularitäten 47

3.3.2 Höher-dimensionale Singularitäten 48

3.3.3 Moduli 48

4 Numerisches Verfahren **50**

4.1 Wahl der Ränderungen 50

4.2 Anheben der Entfaltung 51

4.3 Algorithmus 54

5 Numerische Beispiele **57**

5.1 Skalare Singularitäten 57

5.1.1 Umkehrpunkt 57

5.1.2 Einfache Verzweigung 58

5.1.3 Pitchfork 59

5.1.4 Geflügelter Spitzpunkt 59

5.1.5 Einsiedlerpunkt 60

5.1.6 Tripel Punkt 60

5.2 Brusselator 61

5.2.1 Einfache Verzweigung mit Rangdefekt 2 62

5.2.2 Höhere Verzweigung mit Rangdefekt 2 63

5.2.3 Einfache Verzweigung mit Rangdefekt 3 64

5.3 Weitere Beispiele 65

A Hilfsmittel **67**

A.1 Satz über implizite Funktionen 67

Literatur **68**

Erklärung **71**

Zusammenfassung

Zur Bestimmung singulärer Punkte eines bestimmten Typs muss eine zugehörige reduzierte Funktion und deren Ableitungen bestimmte Bedingungen erfüllen. Dabei ist diese reduzierte Funktion implizit durch ein nichtlineares Gleichungssystem definiert. Man erhält letztendlich ein minimal erweitertes System, das auch Ableitungen der reduzierten Funktion enthält, und den singulären Punkt als reguläre Lösung besitzt.

In der vorliegenden Arbeit wird die Technik des automatischen Differenzierens für die Vorwärtsmethode dargestellt, insbesondere wird die Differentiation iterativer Verfahren untersucht. Es wird ein Überblick über die Theorie von singulären Punkten gegeben und das Erkennungsproblem definiert. Ein zweistufiges Verfahren zur Bestimmung singulärer Punkte wird auf Basis der Vorwärtsmethode und des Newton-Verfahrens beschrieben und wurde an verschiedenen Typen von singulären Punkten getestet.

Notationen

Im Rahmen dieser Arbeit werden Funktionen häufig nur in einer beliebig kleinen Umgebung eines Punktes betrachtet, beispielsweise beim Definitionsbereich impliziter Funktionen oder bei der Kontaktäquivalenz reduzierter Funktionen. In diesen Fällen handelt es sich vielmehr um Funktionenkeime als um Funktionen und beide Begriffe werden synonym verwendet. Insbesondere bei der Angabe von Definitionsbereichen wird die Angabe einer Umgebung des Öfteren zugunsten der Dimensionsangabe der Argumente geopfert.

Liste verwendeter Symbole

Im Folgendem seien $f : \mathbb{R}^n \to \mathbb{R}^m,\ x \mapsto f(x)$, und $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^l,\ (x, y) \mapsto g(x, y)$, genügend reguläre Funktionen und $M \in \mathbb{R}^{n \times m}$ eine Matrix.

Symbol	Beschreibung
$\equiv$	Gleichheit von Funktionen auf dem gesamten Definitionsbereich oder in einer Umgebung
$\mathcal{C}^k$	Menge der k-mal stetig differenzierbaren Funktionen, $k \in \mathbb{N} \cup \{\infty\}$
∂f	$= \left(\frac{\partial f}{\partial x_1} \cdots \frac{\partial f}{\partial x_n}\right) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$, Jacobi-Matrix von f

Symbol	Beschreibung
$g_x = \partial_x g$	$= \left(\frac{\partial g}{\partial x_1} \dots \frac{\partial g}{\partial x_n}\right)$, Jacobi-Matrix von g bezüglich der Variablen x
$\partial^k f$	$= \left(\frac{\partial^{\lvert\mathbf{i}\rvert} f}{\partial x_1^{\mathbf{i}_1} \cdots \partial x_n^{\mathbf{i}_n}}\right)_{\lvert\mathbf{i}\rvert=k}$, Ableitungstensor k-ter Ordnung von f
$\mathbb{B}_\varrho(x)$	$= \{y : \lVert x - y\rVert < \varrho\}$, Ball um x mit Radius ϱ
$\overline{M}$	$= \{y : \exists (x_i)_{i\in\mathbb{N}} \subset M \text{ mit } \lim_{i\to\infty} x_i = y\}$, Abschluss der Menge M
$j \prec i$	Abhängigkeitsbeziehung, beschreibt die Menge aller Variablen mit Index j, die in direkter Weise zur Berechnung der Variable mit Index i benötigt werden
$\mathrm{OPS}(f)$	Anzahl der Fließkommaoperationen, die zur Berechnung von f durchgeführt werden müssen
$\delta_{i,j}$	$= \begin{cases} 1, & i = j \\ 0, & \text{sonst} \end{cases}$, Kroneckersymbol
$\mathbb{M}_{b,l}$	Menge der Maschinenzahlen zur Basis b mit Mantissenlänge l
$\mathrm{fl} : \mathbb{R} \to \mathbb{M}_{b,l}$	Rundung in $\mathbb{M}_{b,l}$
$\mathrm{eps}_{b,l}$	$= \min\{x \in \mathbb{M}_{b,l} : \mathrm{fl}(1 + x) > 1\}$, Maschinenepsilon in $\mathbb{M}_{b,l}$

Symbol	Beschreibung
$\mathrm{Id}, \mathrm{Id}_n$	$(n \times n)$ Einheitsmatrix, Dimension wird in eindeutigen Fällen weggelassen
e_i, e_i^n	i-te Spalte der $(n \times n)$ Einheitsmatrix
$f(x) = \mathcal{O}(g(x))$ für $x \to x_0$	$\iff \limsup_{x \to x_0} \left\lvert \frac{f(x)}{g(x)} \right\rvert < \infty,\ x_0 \in \mathbb{R} \cup \{\pm\infty\}$, Landau'sche Groß-O-Notation
$\ker M$	$= \{x : Mx = 0\}$, Kern oder Nullraum von M
$\mathrm{im}\, M$	$= \{y : y = Mx\}$, Bildraum von M
$\mathrm{rang}\, M$	Anzahl der linear unabhängigen Zeilen oder Spalten von M
$f \sim g$	Kontaktäquivalenz der Funktionen f und g

Kapitel 1

Automatisches Differenzieren

1.1 Einleitung

Um Ableitungswerte einer Funktion $f : \mathbb{R}^n \to \mathbb{R}^m$ numerisch zu bestimmen, kann man Ableitungen eines Interpolationspolynoms p verwenden. In der Praxis führt dies häufig zur Anwendung einfacher Formeln wie

$$\partial_i f(x) \approx \frac{f(x + he_i) - f(x)}{h} \tag{1.1}$$

oder

$$\partial_i f(x) \approx \frac{f(x + he_i) - f(x - he_i)}{2h} \tag{1.2}$$

mit einer Schrittweite $h > 0$. Dies kann durchaus die einzige Möglichkeit zur Berechnung von Approximationen an die Ableitung sein, etwa wenn die Funktionswerte gar nicht durch ein Computerprogramm, sondern durch physikalische Messungen gegeben werden. Allerdings treten dabei große Fehler auf, denn einerseits gibt es starke Auslöschungseffekte, wenn h klein ist und damit die Funktionswerte nah beieinander liegen, andererseits nimmt der Diskretisierungsfehler zu, wenn h groß ist. Als Faustregel gilt: Etwa die Hälfte der Mantisse wird durch Auslöschung zugunsten des Diskretisierungsfehlers geopfert. Bei höheren Ableitungen verstärkt sich dieses Problem, insbesondere können höhere Ableitungen nicht rekursiv mit der gleichen Formel berechnet werden, sondern es sind spezielle Formeln nötig.

Abbildung 1.1 zeigt die Abhängigkeit des Fehlers von der gewählten Schrittweite, aber wie Abbildung 1.2 zeigt, hängt der Fehler auch von der Beschaffenheit der Funktion an der zu differenzierenden Stelle ab.

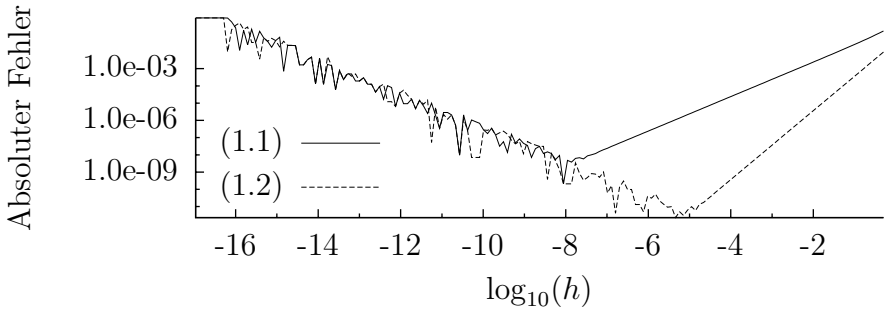

Abbildung 1.1: Absoluter Fehler pro Diskretisierungsschrittweite bei Approximation der Ableitung von $\sin(x)$ an der Stelle $x = 0.5$ mittels Vorwärtsdifferenzenquotient und zentralem Differenzenquotient.

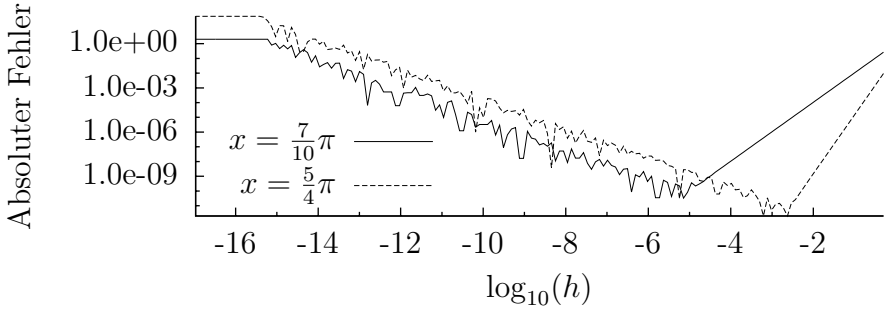

Abbildung 1.2: Das Beispiel $f(x) = \sin(x) * \exp(x)$ zeigt, dass die Wahl der optimalen Schrittweite stark von der Krümmung der Funktion an der entsprechenden Stelle abhängt. Dabei wurde der zentrale Differenzenquotient (1.2) verwendet.

Werden raffiniertere Techniken angewendet, etwa Ridders Methode (siehe z.B. [PTVF92, p. 188f]), nimmt die Anzahl der Funktionsauswertungen deutlich zu und es werden immer noch starke Annahmen über das Wachstumsverhalten der Funktion getroffen und wir erhalten wieder nur erste Ableitungen.

Falls die Funktion durch eine analytische Vorschrift gegeben ist, kann man auch symbolisches Differenzieren verwenden, um explizite Formeln für die Ableitungen zu bestimmen. Dieser Ansatz ist aber sehr aufwändig und relativ fehleranfällig, falls er von Hand durchgeführt wird. Wird er durch ein Computeralgebrasystem erledigt, kann die Komplexität der resultierenden Ausdrücke besonders bei höheren Ableitungen bezüglich einer großen Anzahl an Variablen erhebliche Probleme für die Laufzeiteffizienz nach sich ziehen.

Abhilfe für diese Probleme schafft das Automatische (auch Algorithmische) Differenzieren, im Folgenden kurz *AD* genannt. Dies ist eine Methode, um ein Computerprogramm zur Berechnung einer Funktion durch wiederholtes Anwenden der Kettenregel

$$\frac{df}{dx} = \frac{dg}{dh}\frac{dh}{dx} \quad \text{für} \quad f(x) = g(h(x)) \tag{1.3}$$

in eines zu transformieren, das auch Ableitungswerte jener Funktion liefert. Im Gegensatz zu algebraischen Systemen, die mit symbolischer Differentiation arbeiten, wird beim *AD* die Kettenregel allerdings nicht auf symbolische Ausdrücke, sondern auf numerische Werte angewendet. Da die Ableitungen der Elementaroperationen bekannt sind, müssen diese nicht mit Hilfe einer Näherungsformel berechnet werden und die oben beschriebenen Auslöschungseffekte treten nicht auf. Stattdessen unterliegt eine mittels *AD* berechnete Ableitung im Wesentlichen den gleichen Rundungsfehlern wie die Ausgangsfunktion.

Für das Automatische Differenzieren gibt es zwei Techniken: Die Vorwärtsmethode, bei der die Ableitungen mit einer Art erweiterter Variablen zusammen mit dem Funktionswert berechnet werden, und die Rückwärtsmethode, bei der die Berechnung der Ableitungen entgegen der Berechnungsfolge der Funktionswerte mit Hilfe von sogenannten Adjungierten erfolgt.

Die Implementation der Vorwärtsmethode stellt sich unkompliziert dar, denn die Kettenregel wird hier einfach von rechts nach links durchlaufen, also in (1.3) erst $\frac{dh}{dx}$ und dann $\frac{dg}{dh}$ berechnet.

Bei der Rückwärtsmethode erfolgt die Berechnung dagegen in umgekehrter Reihenfolge. Sie erfordert deutlich weniger Rechenaufwand, wenn die Anzahl der Funktionswerte klein gegenüber der Anzahl der Veränderlichen ist, benötigt dafür allerdings mehr Speicherplatz. Dies ist das sogenannte *cheap gradient principle* [GW08, 3.3], das besagt, dass die Anzahl der benötigten Operationen zur Berechnung des Gradienten einer skalaren Funktion mit der Rückwärtsmethode unabhängig von der Anzahl der Veränderlichen durch das 4-fache der Anzahl der Operationen der Ausgangsfunktion beschränkt ist.

Ist jedoch die Anzahl der Variablen klein, stellt die Vorwärtsmethode die bessere Wahl dar. Beide Methoden ergänzen sich also und es ist vom konkreten Problem abhängig, welche - oder im Fall höherer Ableitungen, welche Kombination - verwendet werden sollte.

Vom technischen Standpunkt aus gesehen gibt es zwei mögliche Implementationswege: Quellcodetransformation und überladene Operatoren. Bei der

Quellcodetransformation wird zum Beispiel durch Anmerkungen an den Programmcode die zu differenzierende Stelle markiert und ein externes Programm erzeugt den Quellcode zur Berechnung der Ableitung. Während hierbei sehr effizienter Quellcode für die Berechnung der Ableitungen erzeugt werden kann, der auch von den durch Compiler angewendeten Optimierungen profitiert, besteht aber auch die Gefahr, dass bei sehr komplexen Ausgangsfunktionen der zusätzlich erzeugte Code den Compiler vor Probleme stellt.

Die dieser Arbeit beiliegende Referenzimplementation `deriv` nutzt die Methode der überladenen Operatoren.

Überladene Operatoren basieren auf einer Technik, die es erlaubt, die Bedeutung von Operatoren und Funktionen einer Programmiersprache auf benutzerdefinierte Typen zu erweitern. Dabei wird entweder der Berechnungsweg aufgezeichnet und später durchlaufen, um die Ableitungswerte zu ermitteln, oder die Ableitungen werden direkt mitgerechnet. Der Benutzer verwendet dann statt des üblichen Fließkommadatentyps die vom *AD* bereitgestellte Klasse. Der eigentliche Code zur Berechnung der Funktion muss nicht oder nur geringfügig verändert werden.

Während die Technik der überladenen Operatoren die Messlatte an die Programmiersprache etwas höher legt und somit nicht auf C oder ältere Versionen von Fortran anwendbar ist, kann man hier mit Quellcodetransformationen Erfolg haben. Andererseits ist zum Einlesen und zur Analyse der Programme vor der Transformation ein erheblicher Aufwand nötig, insbesondere bei verschiedenen Standards (z.B. Fortran 66/77/90/95/2003/2008), bei denen sich mitunter die Struktur des Quellcodes stark unterscheidet. Bedenkt man außerdem, dass eine komplexe numerische Berechnung Teilschritte in Funktionen oder Objekte kapselt, ist es leicht ersichtlich, dass bei der Implementation mittels überladenen Operatoren wesentlich weniger technische Details eine Hürde darstellen.

Im Folgenden werden zunächst die Grundbestandteile des *AD* formalisiert. Anschließend wird die Vorwärtsmethode eingeführt.

Definition 1.1. *Sei* $f : \mathbb{X} \to \mathbb{Y},\ \mathbb{X} \subset \mathbb{R}^n, \mathbb{Y} \subset \mathbb{R}^m$. *Ein* zu f gehöriger Algorithmus *ist eine endliche Zerlegung von* f *entsprechend*

$$f = \phi_l \circ \phi_{l-1} \circ \cdots \circ \phi_1$$

mit $\phi_i : \mathbb{X}_{i-1} \to \mathbb{X}_i,\ \mathbb{X}_i \subset \mathbb{R}^{n_i}$ *für* $i = 0, \ldots, l$, *und* $\mathbb{X}_0 = \mathbb{X}, \mathbb{X}_l = \mathbb{Y}$. *Dabei hat jede Komponente von* $\phi_i(x_i),\ i = 1, \ldots, l$, *mit* $x_i = (x_{i,1}, \ldots, x_{i,n_i})^\intercal \in \mathbb{X}_i$

die Form

$$(\phi_i(x_i))_j = \begin{cases} \pm x_{i,k} & \text{mit } k \in \{1,\dots,n_i\} \\ \pm(x_{i,k} \diamond x_{i,m}) & \text{mit } k,m \in \{1,\dots,n_i\},\ \diamond \in \{+,-,\cdot,/\} \\ \pm f(x_{i,k}) & \text{mit } k \in \{1,\dots,n_i\},\ f \text{ Standardfunktion} \end{cases}$$
$$j = 1,\dots,n_{i+1}.$$

Beim *AD* wird ausgenutzt, dass eine Implementation eines Algorithmus immer aus einer Folge von Zuweisungen eines durch Elementaroperationen erzeugten Zwischenergebnisses besteht. Eine mögliche Formalisierung dieses Sachverhaltes gibt folgende Definition, siehe auch [GW08, p. 19, Table 2.2].

Definition 1.2. *Ein* Computerprogramm *zur Auswertung einer Funktion $f : \mathbb{R}^n \to \mathbb{R}^m$ an einer Stelle $x \in \mathbb{R}^n$ ist eine Folge von Anweisungen der Form*

$$\begin{aligned} v_{i-n} &= x_i, & i &= 1,\dots,n \\ v_i &= \theta_i(v_j)_{j \prec i}, & i &= 1,\dots,l \\ y_{m-i} &= v_{l-i}, & i &= m-1,\dots,0, \end{aligned} \tag{1.4}$$

wobei die θ_i Elementaroperationen seien. Dabei bedeute die Notation $j \prec i$, dass v_i direkt von v_j abhängt. Bei der Berechnung von v_i kann insbesondere eine beliebige Anzahl von Variablen v_j mit $j \leq i$ herangezogen werden, üblicherweise ist diese Anzahl allerdings nicht größer als zwei. Dabei heißt $i \prec i$, dass die Berechnung iterativ erfolgt. Ein derartiger Berechnungsschritt ist zwar in Definition 1.1 nicht vorgesehen, ist aber zur Berechnung der Ableitungen impliziter Funktionen unabdingbar. Dieses Thema wird in Kapitel 2 ausführlich diskutiert.

Bemerkung 1.3. *Entscheidend bei obiger Definition ist die Wahl der Elementaroperationen. Die Initialisierung mit einer Konstante, Addition, Multiplikation und der Vorzeichenwechsel sind in jedem Fall notwendig. Daraus können theoretisch alle anderen Operationen durch Polynomauswertungen und Nachschlageoperationen (*table look-up*) gewonnen werden. In vielen Programmiersprachen stehen auch komplexere Funktionen, z.B. die trigonometrischen Funktionen und ihre Inversen, als Standardfunktionen zur Verfügung. Da deren Ableitungen bekannt sind, ist es sinnvoll diese Funktionen den Elementaroperationen zuzurechnen. Tabelle 1.1 zeigt eine mögliche Auswahl.*

In der dieser Arbeit beiliegenden Referenzimplementation werden folgende Elementaroperationen zur Verfügung gestellt:

$$\pm u,\ u \pm v,\ u \cdot v,\ \tfrac{u}{v},\ \exp(u),\ \log(u),\ u^2,\ \sqrt{u},\ u^k,\ u^v$$
$$\sin(u),\ \cos(u),\ \tan(u),\ \arcsin(u),\ \arccos(u),\ \arctan(u),$$

wobei u, v Variablen oder Konstanten bezeichnen und $k \in \mathbb{Z}$.

	Essentiell	Optional	Vektor
Glatt	$u + v$, $u * v$, $-u$, c, $\frac{1}{u}$, $\exp(u)$, $\log(u)$, $\sin(u)$, $\cos(u)$, $\lvert u\rvert^c,\ c > 1$ ...	$u - v$, $\frac{u}{v}$ $c * u$, $c \pm u$, u^k, $\arcsin(u)$, $\tan(u)$, ...	$\sum_{k=1}^n u_k v_k$, $\sum_{k=1}^n c_k v_k$
Lipschitz	$\lvert u\rvert$, $\lVert u, v\rVert$	$\max(u, v)$, $\min(u, v)$	$\max_k(\lvert u_k\rvert)$, $\sum_{k=1}^n \lvert u_k\rvert$, $\sqrt{\sum_{k=1}^n u_k^2}$
Allgemein	$\text{heav}(u) = \begin{cases} 1, & u \geq 0 \\ 0, & \text{sonst} \end{cases}$	$\text{sign}(u)$, $(u > 0)\ ?\ u : v$	

Tabelle 1.1: Elementare Operatoren und Funktionen nach Griewank [GW08, Table 2.3]. Dabei sind u, v, u_k und v_k Variablen, c und c_k Konstanten, sowie n und k natürliche Zahlen.
Die Spalte *Optional* zählt Operationen auf, die auch durch Hintereinanderausführung der essentiellen Funktionen gewonnen werden könnten, deren Ableitung aber bekannt und als analytischer Ausdruck darstellbar ist. Sie tauchen in einer Vielzahl von Programmiersprachen als Standardfunktionen auf und sind deshalb eigentlich unentbehrlich.
In der Spalte *Vektor* finden sich Operationen, die innerhalb von Bibliotheken zur Linearen Algebra den Status von Elementaroperationen einnehmen würden und daher von einer direkten Unterstützung durch *AD* Software profitieren können.
Die Zeilen *Lipschitz* und *Allgemein* enthalten Operationen mit nicht differenzierbaren Stellen, die dennoch fast überall differenzierbar und für die Implementation komplexerer Algorithmen unabdinglich sind.

In dieser Definition wird die Anwendung von Verzweigungen bei der Berechnung einer Funktion vernachlässigt. Wird beispielsweise der Absolutwert eines Ausdrucks benötigt, erfolgt dies üblicherweise in der Form

```
if(x >= 0.0)
    return x;
else
    return -x;
```

Dabei ist jeder Teil der Verzweigung stetig differenzierbar, nur der Grenzfall $x = 0.0$ ist problematisch. Dieses Beispiel ist exemplarisch für alle nicht differenzierbaren Stellen, die ein Computerprogramm aufweisen kann. [BF94] behandelt das Problem stückweise definierter Funktionen im Zusammenhang mit AD ausführlich und liefert folgendes Resultat für $\mathcal{C}^1$-Funktionen f, die stückweise durch $\mathcal{C}^1$-Funktionen r_i definiert werden :

Ist $f : \mathbb{R} \to \mathbb{R}$ *stückweise definiert auf* $\mathbb{D}_i \subset \mathbb{U}_i \subset \mathbb{R}$ *durch* $r_i : \mathbb{U}_i \to \mathbb{R},\ i \in \mathcal{I}$, *dann gilt* $\partial f(x) = \partial r_k(x)$, *falls* $x \in \mathbb{U}_k$ *und es gibt eine Folge* $x_m \in \mathbb{D}_k$ *mit* $x_m \to x$.

Im Beispiel des Absolutbetrags ist dieses Resultat nicht anwendbar, da die Funktion selbst nicht stetig differenzierbar ist. Das AD liefert selbst in diesem Fall an den kritischen Punkten noch Richtungsableitungen oder kann mit einem Fehlerzustand reagieren.

Zusammenfassend lässt sich sagen, dass abgesehen von Zuweisungen der Form

$$v_i = \theta_i(v_j)_{j \prec i} \text{ mit } i \in \{j : j \prec i\}$$

alle Teile eines Computerprogramms fast überall analytisch sind. Es wird sich zeigen, dass auch diese Operationen unter einigen Vorausetzungen an die Iterationsvorschrift θ_i genügend regulär sind. Dies motiviert folgende Annahme.

Annahme 1.4. *Die Elementaroperationen und -funktionen seien auf einer offenen Teilmenge* $\mathbb{D} \subset \mathbb{R}^n$ *d-mal stetig differenzierbar,* $0 < d \leq \infty$.

1.2 Vorwärtsmethode

Die Vorwärtsmethode liefert in ihrer grundlegenden Form mit jedem Durchlauf eine Richtungsableitung $\partial f \cdot a,\ a \in \mathbb{R}^n$. Abweichend von dieser in der Literatur üblichen Formulierung wird im Folgenden eine spezielle Variante etabliert, die stets die gesamte Jacobi-Matrix liefert.

Definition 1.5. *Ausgehend von Definition 1.2 liefert die* Vorwärtsmethode

ein Computerprogramm zur Auswertung der Jacobi-Matrix ∂f durch

$$\begin{array}{rll} v_{i-n} = x_i, & & \\ \dot{v}_{i-n,k} = \delta_{i,k}, & k = 1, \ldots, n & \Big\} \quad i = 1, \ldots, n \\ \hline v_i = \theta_i(v_j)_{j \prec i}, & & \\ \dot{v}_{i,k} = \sum_{s \prec i} \dot{v}_{s,k}\, \partial_{v_s} \theta_i(v_j)_{j \prec i}, & k = 1, \ldots, n & \Big\} \quad i = 1, \ldots, l \\ \hline y_{m-i} = v_{l-i}, & & \\ \dot{y}_{m-i,k} = \dot{v}_{l-i,k}, & k = 1, \ldots, n & \Big\} \quad i = m-1, \ldots, 0 \end{array} \tag{1.5}$$

Bemerkung 1.6.

(i) Allgemeiner kann mit obigen Algorithmus auch eine Richtungsableitung in Richtung $a \in \mathbb{R}^n$ berechnet werden, indem $\dot{v}_{i-n} = a_i$ für $i = 1, \ldots, n$ gesetzt und der Index k überall weggelassen wird. Tatsächlich ist dies der AD*-Literatur eingeschlagene Weg, die Jacobi-Matrix erhält man dann durch wiederholtes Anwenden auf die Einheitsvektoren. Im Rahmen dieser Arbeit wird aber immer die gesamte Jacobi-Matrix benötigt.*

(ii) In der Beschreibung (1.5) *erkennt man auch die mechanische Natur der Vorwärtsmethode. Die Ableitungen $\partial_{v_j} \theta_i(v_j)_{j \prec i}$ der Elementaroperationen und -funktionen sind a-priori bekannt und müssen nur noch mit der inneren Ableitung $\dot{v}_{j,k}$ multipliziert und gegebenenfalls anschließend aufsummiert werden. Die Umsetzung dieser Methode kann also vollständig durch eine Maschine geschehen.*

(iii) Die bei der Implementation übliche Praxis, Variablen mit neuen Werten zu überschreiben, spiegelt sich nicht in der Beschreibung (1.5) *wieder. Schlimmer noch, enthält das Programm zur Auswertung der Funktion eine Anweisung der Form*

$$v = \theta(v) \text{ oder } v = \theta(v, u),$$

wobei v einen vorher berechneten, aber nicht weiter benötigten Wert enthält, so ist der Wert der durch (1.5) *gegebenen Variable $\dot{v}$ fehlerhaft, da in der Auswertung $\partial_v \theta(v)$ bzw. $\partial_v \theta(v, u)$, $\partial_u \theta(v, u)$ bereits der neue Wert von v verwendet wird. Eine mögliche Lösung ist, die Auswertung der Ableitung der Funktionsauswertung voranzustellen, was in einigen Fällen jedoch zusätzliche Auswertungen des gleichen Ausdrucks erfordert.*

1.2.1 Aufwand und Fehlerabschätzung

Unter der Annahme, dass keine Variable v_i von mehr als zwei anderen direkt abhängt, ist der Aufwand zur Berechnung der Jacobi-Matrix mit Hilfe von (1.5) beschränkt durch

$$\mathrm{OPS}_{(1.5)}(\partial f) \leq (1+3n)\,\mathrm{OPS}(f). \tag{1.6}$$

Der Faktor 3 tritt im Fall der Multiplikation auf, für unäre Funktionen ist der Faktor nicht größer als 2, bei der Addition/Subtraktion sogar nur 1.

Beispielsweise muss bei der Ableitung des Arkussinus zwar ein Quadrat, eine Wurzel, eine Subtraktion, eine Inverse und schließlich die Multiplikation mit der inneren Ableitung berechnet werden, jedoch ist die Berechnung des Arkussinus selbst mit hoher Wahrscheinlichkeit teurer als dies.

Im allgemeinen wird diese Schranke also deutlich unterschritten.

Zum Vergleich, die Kosten zur Auswertung von ∂f mit dem einfachen Differenzenquotienten (1.1) sind gegeben durch

$$\mathrm{OPS}_{(1.1)}(\partial f) = \mathrm{OPS}(f) + n\Big(2m + \mathrm{OPS}(f)\Big).$$

Wird Ridders Methode verwendet, sind die Kosten bereits mindestens so groß wie beim *AD*.

Bei Berechnungen mittels Fließkommazahlen ist die Genauigkeit beim *AD* jedoch überragend. In (1.5) erkennt man, dass Auslöschungseffekte nicht auftreten können, sofern sie nicht bereits in der Berechnung von f selber auftreten. Man sagt deshalb: Was gut für die Funktionswerte ist, ist auch gut für ihre Ableitungen ([GW08, Rule 2, p. 20]). Die Berechnungsvorschrift für die Ableitungen, die durch *AD* gegeben wird, entspricht der exakten Vorschrift bezüglich im Bereich der Maschinengenauigkeit gestörten Elementarfunktionen ([GW08, Rule 3, p. 51]).

Definition 1.7 (Rundung)**.** *Eine* Rundung *ist eine Abbildung* $\mathrm{fl} : \mathbb{R} \to \mathbb{M}_{b,l}$ *mit*

$$\mathrm{fl}(x) = x \qquad \forall\, x \in \mathbb{M}_{b,l}, \qquad \textit{(Projektion)} \tag{1.7}$$

$$\mathrm{fl}(x) \leq \mathrm{fl}(y) \quad \forall\, x, y \in \mathbb{R} \text{ mit } x \leq y. \qquad \textit{(Monotonie)} \tag{1.8}$$

Definition 1.8. *Eine Rundung* $\mathrm{fl} : \mathbb{R} \to \mathbb{M}_{b,l}$ *heißt* optimal*, falls*

$$|\,\mathrm{fl}(x) - x| \leq \frac{1}{2} b^{-l+1} |x|.$$

Die Größe $\mathrm{eps}_{b,l} = \frac{1}{2}b^{-l+1}$ *heißt* Maschinenepsilon *zur Basis* b *und Mantissenlänge* l*.*

Bemerkung 1.9.

(i) Für die nach Standard [IEE08] definierte Menge der doppelt genauen Fließkommazahlen,

$$\mathbb{M} = \left\{ \pm \sum_{k=1}^{53} d_k 2^{r-k} \mid d_i \in 0,1,\ d_1 = 1,\ -1022 \le r \le 1023 \right\} \cup \Big\{0\Big\} \subset \mathbb{M}_{2,53}$$

ist die kaufmännische Rundung eine optimale Rundung, wenn der Definitionsbereich von $\mathbb{R}$ *auf* $[-2^{1023}, 2^{1023}]$ *eingeschränkt wird. Dies bedeutet im Wesentlichen nur, dass bei Überläufen des Exponenten Probleme auftreten und mit einem Fehler zu rechnen ist.*

(ii) Ist eine Rundung optimal, so existiert zu jedem $x \in \mathbb{R}$ *ein* $\varepsilon \in \mathbb{R}$ *mit*

$$\mathrm{fl}(x) = x(1+\varepsilon),\ |\varepsilon| \le \mathrm{eps}\,. \tag{1.9}$$

Satz 1.10. *(Fehlerabschätzung) Sei* $\mathrm{fl} : \mathbb{R} \to \mathbb{M}_{b,l}$ *eine optimale Rundung. Dann liefert die Vorwärtsmethode unter Beachtung der Rundung exakte Ergebnisse bezüglich im Bereich der Maschinengenauigkeit* eps *gestörten Elementarfunktionen.*

Beweis. (vergleiche [GW08, p. 50f], dort nur für univariate Funktionen)
Wegen (1.9) wird angenommen, dass

$$\tilde{v}_i := \mathrm{fl}(v_i) = (1+\varepsilon_i)\theta_i(\tilde{v}_j)_{j \prec i},\ i = 1, \ldots, l$$

und für $c_{is}(\tilde{v}_j)_{j \prec i} := \partial_{v_s}\theta_i(\tilde{v}_j)_{j \prec i}$ gilt

$$\tilde{c}_{is}(\tilde{v}_j)_{j \prec i} := \mathrm{fl}\left(c_{is}(\tilde{v}_j)_{j \prec i}\right) = (1+\varepsilon_{is})\partial_{v_s}\theta_i(\tilde{v}_j)_{j \prec i},\ s \prec i,$$

wobei $|\varepsilon_i| \le \mathrm{eps} \ge |\varepsilon_{is}|$ für alle i und $s \prec i$.
Wird nun (1.5) unter Beachtung der Rundung fl angewendet, ergibt sich für

die gerundeten Werte $\dot{\tilde{v}}_{i,k}$ der Ableitungen $\dot{v}_{i,k}$ im Sinne der Rückwärtsanalyse

$$\begin{aligned}
\dot{\tilde{v}}_{i,k} &= \mathrm{fl}\left(\sum_{s\prec i} \tilde{c}_{is}(\tilde{v}_j)_{j\prec i}\dot{\tilde{v}}_{s,k}\right) \\
&= (1+\varepsilon_{ik,\mathrm{sum}})^{n_i-1}\sum_{s\prec i}\mathrm{fl}\left(\tilde{c}_{is}(\tilde{v}_j)_{j\prec i}\dot{\tilde{v}}_{s,k}\right) \\
&= (1+\varepsilon_{ik,\mathrm{sum}})^{n_i-1}\sum_{s\prec i}(1+\varepsilon_{iks,\mathrm{mul}})(1+\varepsilon_{is})c_{is}(\tilde{v}_j)_{j\prec i}\dot{\tilde{v}}_{s,k} \\
&=: \sum_{s\prec i}(1+\dot{\varepsilon}_{iks})c_{is}(\tilde{v}_j)_{j\prec i}\dot{\tilde{v}}_{s,k}\,, \qquad\qquad k=1,\ldots,n,
\end{aligned}$$

wobei $n_i = \#\{j : j \prec i\}$ die Anzahl der Argumente von θ_i bezeichne. Dabei sind alle $|\varepsilon_{ik,\mathrm{sum}}|, |\varepsilon_{iks,\mathrm{mul}}| \le \mathrm{eps}$ und für die $\dot{\varepsilon}_{iks}$ gilt

$$(1-\mathrm{eps})^{1+n_i} \le 1+\dot{\varepsilon}_{iks} \le (1+\mathrm{eps})^{1+n_i}$$

und wegen $\mathrm{eps} < 1$ folgt daraus

$$\dot{\varepsilon}_{iks} \in \mathcal{O}((1+n_i)\,\mathrm{eps}).$$

Definiert man für $k = 1,\ldots,n$

$$\tilde{\theta}_i^k(v_j)_{j\prec i} := (1+\varepsilon_i)\,\theta_i\left(\frac{(1+\dot{\varepsilon}_{ikj})v_j + (\varepsilon_i - \dot{\varepsilon}_{ikj})\tilde{v}_j}{1+\varepsilon_i}\right)_{j\prec i}$$

ist also

$$\begin{aligned}
\tilde{\theta}_i^k(\tilde{v}_j)_{j\prec i} &= (1+\varepsilon_i)\theta_i\left(\frac{1+\dot{\varepsilon}_{ikj}+\varepsilon_i-\dot{\varepsilon}_{ikj}}{1+\varepsilon_i}\tilde{v}_j\right)_{j\prec i} \\
&= (1+\varepsilon_i)\theta_i(\tilde{v}_j)_{j\prec i} \\
&= \tilde{v}_i
\end{aligned}$$

und

$$\begin{aligned}
\frac{d}{dx_k}\tilde{\theta}_i^k(\tilde{v}_j)_{j\prec i} &= \sum_{s\prec i}(1+\varepsilon_i)\frac{\partial}{\partial v_s}\theta_i(\tilde{v}_j)_{j\prec i}\left(\frac{1+\dot{\varepsilon}_{iks}}{1+\varepsilon_i}\right)\dot{\tilde{v}}_{s,k} \\
&= \sum_{s\prec i}(1+\dot{\varepsilon}_{iks})c_{is}(\tilde{v}_j)_{j\prec i}\dot{\tilde{v}}_{s,k} \\
&= \dot{\tilde{v}}_{i,k}
\end{aligned}$$

für alle i und $k = 1,\ldots,n$. □

1.3 Höhere Ableitungen

Bei der Verallgemeinerung der in (1.5) beschriebenen Technik auf höhere Ableitungen gibt es eine Reihe von Schwierigkeiten. Der naive Ansatz, die Gleichungen für die ersten Ableitungen der gleichen Methode zu unterwerfen, ist zwar prinzipiell durchführbar, geht jedoch mit einem hohen Preis in Hinblick auf die günstigen Kosten des AD einher. Nach dem Satz von Schwarz über die Vertauschbarkeit der zweiten Ableitungen ist es überflüssig, die Ableitung $\partial_{x_i x_j} f$ zu berechnen, wenn $\partial_{x_j x_i} f$ bereits bekannt ist. Dieser Sachverhalt gilt umso mehr für höhere Ableitungen. Tatsächlich enthält der mn^k Einträge umfassende k-te Ableitungstensor nur

$$m\binom{n+k-1}{k} = m\frac{n\cdot(n+1)\cdots(n+k-1)}{k!} \approx m\frac{n^k}{k!}$$

verschiedene Einträge (Anzahl der Kombinationen von k Elementen einer n-elementigen Menge, wobei jedes Element beliebig oft vorkommen kann). Grob gesagt kann der Aufwand durch Beachtung der Symmetrie fast um den Faktor $k!$ reduziert werden.

Das Ausnutzen der Symmetrie stellt jedoch hohe technische Anforderungen an die Implementation. ADOL-C ([GJU96]) stellt zu diesem Zweck spezielle Module zur Berechnung univariater Taylorentwicklungen bereit, mit deren Hilfe die Symmetrie bei der Berechnung höherer Ableitungstensoren ausgenutzt werden kann, siehe [GUW00].

Kapitel 2

Ableitung iterativer Verfahren

2.1 Einleitung

Nachdem im letzten Kapitel die automatische Differentiation geschlossener Berechnungsvorschriften diskutiert wurde, wenden wir uns jetzt der Differentiation iterativer Verfahren zu. Insbesondere zur Berechnung implizit definierter Funktionen müssen iterative Verfahren verwendet werden. Um die Bestimmungsgleichungen einer Singularität aufzustellen, werden darüber hinaus aber auch Ableitungen einer implizit definierten Funktion benötigt. Dieses Kapitel beschreibt für eine Klasse von Fixpunktverfahren Techniken und Konvergenzbedingungen, mit denen nicht nur die Funktionswerte, sondern auch ihre Ableitungen numerisch bestimmt werden können.

2.2 Voraussetzungen

Sei

$$f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m,\ (x, y) \mapsto f(x, y)$$

eine hinreichend reguläre Funktion mit

$$f(x_0, y_0) = 0 \text{ und } \partial_y f(x_0, y_0) \text{ nichtsingulär} \tag{2.1}$$

für ein $(x_0, y_0) \in \mathbb{R}^n \times \mathbb{R}^m$. Nach Satz A.1 existiert eine (lokal) eindeutig bestimmte Funktion $g : \mathbb{B}_\varepsilon(x_0) \subset \mathbb{R}^n \to \mathbb{R}^m$ mit $f(x, g(x)) \equiv 0$ für alle $x \in \mathbb{B}_\varepsilon(x_0)$, die genauso regulär ist wie f.

Bezeichnung 2.1. *Für den Rest dieses Kapitels sei $x \in \mathbb{B}_\varepsilon(x_0)$ beliebig aber fest und*

$$y_* := g(x) \tag{2.2}$$
$$y'_* := \partial_x g(x) = -f_y(x, y_*)^{-1} f_x(x, y_*). \tag{2.3}$$

Zur Berechnung von y_* werde ein iteratives Verfahren der Form

$$y_{k+1} = \Phi_k(x, y_k) := y_k - P_k f(x, y_k) \tag{2.4}$$

mit $P_k \in \mathbb{R}^{m \times m}$ und einer Ausgangsschätzung $y_0 \in \mathbb{R}^m$ verwendet.

Bemerkung 2.2.

(i) Hier und auch im Rest dieses Kapitels wird die Abhängigkeit der Ierierten y_k, der Ableitungen y'_k und des Vorkonditionierers P_k von den unabhängigen Variablen x nicht explizit aufgeführt. Der Vorkonditionierer P_k kann außerdem von den Iterierten y_i, $i \leq k$, abhängen.

(ii) In [Gil92] können in der Iterationsvorschrift Φ_k noch zusätzliche Parameter a_k auftreten. Dies beeinhaltet beispielsweise temporäre Variablen, in denen die Norm der aktuellen Korrektur abgespeichert wird, oder Hilfsvariablen, die beispielsweise bei globalen Newton-Verfahren wie in [Deu06] Verwendung finden. Wird ein solches Verfahren tatsächlich vollständig als "Black Box" betrachtet und mit einem AD-*Tool differenziert, würden auch von solchen Variablen Ableitungen berechnet werden. Gilbert fordert dann, dass die Ausgangsfunktion nicht von diesen zusätzlichen Parametern abhängt und zeigt die Konvergenz für die tatsächlich abhängigen Variablen y_k.*

Da die Ableitungen dieser Größen nicht relevant sind, wird hier auf diese Unterscheidung verzichtet. Variablen, die nicht zur Funktionsdefinition von f gehören, werden als Teil der Iterationsvorschrift Φ_k betrachtet und nicht differenziert.

Dabei ergibt etwa die Wahl $P_k = \mathrm{Id}$ die *Picard-Iteration* oder

$$P_k = (\partial_y f(x, y_k))^{-1} \tag{2.5}$$

das Newton-Verfahren. Für das Newton-Verfahren gilt folgender Konvergenzsatz. Der Einfachheit halber werden hier die unabhängigen Variablen x gar nicht aufgeführt, es sei also

$$F(y) := f(x, y).$$

Der Beweis erfolgt in Abschnitt 2.4 als Spezialfall des allgemeinen Konvergenzsatzes für Verfahren der Form (2.4).

Satz 2.3 (Konvergenz und Fehlerabschätzung)**.**
Sei $F : \mathbb{D} \to \mathbb{R}^m$ stetig differenzierbar mit $\mathbb{D} \subset \mathbb{R}^m$ offen und $y_* \in \mathbb{D}$ mit $F(y_*) = 0$. Für alle $y \in \mathbb{D}$ sei die Jacobi-Matrix $\partial F(y)$ invertierbar und es gelte für ein $\omega > 0$:

$$\left\|\partial F(y)^{-1}\Big(\partial F(y) - \partial F(\tilde{y})\Big)\right\| \leq \omega \|y - \tilde{y}\| \quad \textit{für alle } y, \tilde{y} \in \mathbb{D} \tag{2.6}$$

in einer Vektornorm mit zugehöriger Matrixnorm. Die Startschätzung y_0 genüge $\varrho := \|y_* - y_0\| \leq \frac{2}{\omega}$. Dabei sei $\overline{\mathbb{B}_\varrho(y_*)} \subset \mathbb{D}$. Dann definiert (2.4) zusammen mit (2.5) eine Folge $\{y_k\}$ in $\overline{\mathbb{B}_\varrho(y_*)}$ mit $y_k \to y_*$ und es gelten:

$$\|y_* - y_{k+1}\| \leq \frac{\omega}{2} \|y_* - y_k\|^2 \tag{2.7a}$$

$$\|y_* - y_k\| \leq \frac{2}{\omega}\gamma^{2^k}, \ \gamma \in (0, 1). \tag{2.7b}$$

Ferner ist y_* die einzige Nullstelle von F in $\mathbb{B}_{\frac{2}{\omega}}(y_*) \cap \mathbb{D}$.

Bemerkung 2.4.

(i) Unter veränderten Voraussetzungen kann auch die Existenz einer Lösung gefolgert werden, vergleiche zum Beispiel das Newton-Kantorovich Theorem oder das Newton-Mysovskikh Theorem in [Deu06, 2.1, 2.2]. Im Rahmen dieser Arbeit ist die Existenz aber immer a-priori gegeben oder aber durch die Existenzaussage in Satz A.1 gesichert.

(ii) Das Newton-Verfahren ist recht unempfindlich gegenüber Störungen in der Jacobi-Matrix. Dabei geht aber im Allgemeinen die quadratische Konvergenz verloren, siehe z.B. den Abschnitt zu vereinfachten Newton-Verfahren *in [Deu06, 2.1.2].*

Um auch eine Approximation an die Ableitung y'_* zu berechnen, werden in der Literatur ([Gil92], [GBC+93], [BB98]) eine Reihe von Verfahren vorgestellt.

(i) Beim *vollständig differenzierten Verfahren* wird die gesamte Iteration als „Black Box" betrachtet und vollständig dem *AD* unterworfen.

(ii) Das *vereinfacht differenzierte Verfahren* unterdrückt die Abhängigkeit des Vorkonditionierers P_k von den unabhängigen Variablen, d.h. es wird $\frac{d}{dx}P_k = 0$ gesetzt.

(iii) Beim *nachträglich differenzierten Verfahren* wird die Iteration zunächst ohne Differentiation bis zur Konvergenz durchgeführt und erst anschließend der Wert der Ableitung bestimmt.

Wegen $\partial_y f(x_0, y_0)$ nichtsingulär ist die Jacobi-Matrix $\partial_y f$ auch in einer ganzen Umgebung von (x_0, y_0) nichtsingulär. Schränkt man $f(x, \cdot)$ auf einen kompakten Ball um $g(x)$ ein, so sind die Funktion und ihre Ableitungen beschränkt und Lipschitz-stetig. Für jedes feste x aus $\mathbb{B}_\varepsilon(x_0)$ erfüllt also $f(x, \cdot)$ in einer Umgebung von $g(x)$ die Voraussetzungen von Satz 2.3. Im Fall des Newton-Verfahrens genügt dies in Verbindung mit der Regularität von f und $\|y_0 - y_*\| \leq \frac{2}{\omega}$ bereits, um auch die Konvergenz der differenzierten Verfahren zu sichern.

Für den allgemeinen Fall geben Griewank et.al. folgende Annahmen als Voraussetzung für die Konvergenz der Ableitungen an.

Annahme 2.5 (Regularität)**.** *(vergleiche [GBC+93, Assumption 1])*
Die Funktion $f(x, \cdot)$ sei in $\mathbb{B}_\varrho(y_*)$ Lipschitz-stetig differenzierbar mit nichtsingulärer Jacobi-Matrix $f_y(x, \cdot)$ bezüglich y, so dass mit Konstanten $M > 0$ und $L > 0$ für alle $y, \tilde{y} \in \mathbb{B}_\varrho(y_*)$ gilt

$$\left\|f_y(x, y)^{-1}\right\| + \left\|\partial f(x, y)\right\| \leq M \tag{2.8}$$

und

$$\|\partial f(x, y) - \partial f(x, \tilde{y})\| \leq L \|y - \tilde{y}\| \tag{2.9}$$

in einer Vektornorm und zugehöriger Operatornorm.

Bemerkung 2.6.

(i) ϱ wird ausschlaggebend für die benötigte Genauigkeit der Startschätzung y_0 sein.

(ii) Die erste Ungleichung bedeutet insbesondere, dass M eine obere Schranke für die Jacobi-Matrix ∂f und damit eine Lipschitz-Konstante von $f(x, \cdot)$ in $\mathbb{B}_\varrho(y_*)$ ist. Ferner ist $f(x, \cdot)$ wegen $f(x, y_*) = 0$ linear beschränkt in $\mathbb{B}_\varrho(y_*)$, d.h.

$$\|f(x, y)\| = \|f(x, y) - f(x, y_*)\| \leq M \|y - y_*\| . \tag{2.10}$$

Zudem erhalten wir aus (2.8) *mit Hilfe von* (2.3) *eine Abschätzung für die gesuchte Ableitung y'_*:*

$$\left\|y'_*\right\| \leq \left\|f_y(x, y_*)^{-1}\right\| \left\|f_x(x, y_*)\right\| \leq M^2. \tag{2.11}$$

(iii) Im Gegensatz zu [GBC+93] wird hier die Lipschitz-Stetigkeit von ∂f mit der Lipschitz-Konstanten L gefordert. In [GBC+93, Assumption 1] wird dagegen die Bedingung

$$\|\partial f(x,y) - \partial f(x,y_*)\| \leq \tilde{L}\,\|y - y_*\| \tag{2.12}$$

für alle $y \in \mathbb{B}_\varrho(y_*)$ gestellt. Wir werden (2.9) aber nur im Fall $\tilde{y} = y_*$ oder unter Integralen in der Form

$$I(y) = \int_0^1 \left\|\partial_y f(x, y_* + t(y - y_*)) - \partial_y f(x,y)\right\| dt$$

benötigen. In diesem Fall haben wir mit (2.9)

$$\begin{aligned} I(y) &\leq L \int_0^1 \left\|y_* - y + t(y - y_*)\right\| dt \\ &= L\,\|y - y_*\| \int_0^1 (1-t)\,dt \\ &= \frac{1}{2} L\,\|y - y_*\| \end{aligned}$$

und andererseits mit (2.12)

$$\begin{aligned} I(y) &\leq \int_0^1 \Big(\left\|\partial_y f\,(x, y_* + t(y - y_*)) - \partial_y f(x,y_*)\right\| + \\ &\qquad + \left\|\partial_y f\,(x,y_*) - \partial_y f(x,y)\right\| \Big)\, dt \\ &\leq \tilde{L} \int_0^1 \Big(\left\|t(y - y_*)\right\| + \left\|y_* - y\right\| \Big)\, dt \\ &\leq \tilde{L}\,\|y - y_*\| \int_0^1 (1+t)\,dt \\ &= \frac{3}{2} \tilde{L}\,\|y - y_*\|\,. \end{aligned}$$

Für unsere Zwecke ist der zusätzliche Faktor 3 nicht relevant, so dass an den entsprechenden Stellen nur L mit $3\tilde{L}$ ersetzt werden muss, wenn wir (2.12) statt (2.9) voraussetzen.

Annahme 2.7 (Kontraktivität)**.** *(vergleiche [GBC+93, Assumption 2]) Sei*

$$D_k := \mathrm{Id} - P_k f_y(x, y_k) \tag{2.13}$$

und für ein $\delta > 0$ gelte

$$\delta_k := \|D_k\| \leq \delta < 1 \text{ für alle } k \in \mathbb{N}. \tag{2.14}$$

Ferner bezeichne

$$\delta_* := \limsup_{k\to\infty} \delta_k \leq \delta. \tag{2.15}$$

Schließlich sei $P_k = P(x, y_k)$ nichtsingulär und stetig differenzierbar für alle $k \in \mathbb{N}$.

Bemerkung 2.8.

(i) Annahme 2.7 ist für das Newton-Verfahren wegen $D_k = 0$ offensichtlich erfüllt.

(ii) Die Forderung $P_k = P(x, y_k)$ bedeutet dabei, dass P_k nicht explizit von den vorhergehenden Iterierten abhängt. Ausgeschlossen sind insbesondere Verfahren, bei denen P_k durch Aufdatierung aus P_i, $i < k$, gewonnen wird, beispielsweise mittels Broyden-Update [Bro65] oder der Davidon-Fletcher-Powell-Formel [FP63]. Für das vereinfacht differenzierte Verfahren ist diese Voraussetzung nicht nötig und ist daher auch nicht Teil von [GBC+93, Assumption 1], sondern wird erst im Konvergenzsatz [GBC+93, Proposition 1] gestellt. Die Autoren zeigen aber dann die Konvergenz beim vollständig differenzierten Verfahren für eine Klasse von Update-Verfahren unter einer zusätzlichen Annahme an die Aufdatierung, siehe [GBC+93, Assumption 3, Lemma 2 und Proposition 2].

(iii) Es ist zu beachten, dass in Annahme 2.7 nicht notwendigerweise die Operatornorm von $\partial_y \Phi$ abgeschätzt wird, wenn $\partial_y P$ von Null verschieden ist, falls es überhaupt existiert.

Die Forderung

$$\varrho(\partial_y \Phi) < 1$$

wurde in [Gil92] gestellt. In praktischen Tests [BCG+92] nährte sich aber die Vermutung, dass diese zu streng ist. In Annahme 2.7 werden daher Terme vermieden, die im Fall des vereinfacht differenzierten Verfahrens nicht zum Tragen kommen.

Die geforderte Abschätzung ist genau auf die Anforderungen der Konvergenz der Ableitungen y'_k zugeschnitten, garantiert aber auch zusammen mit [GBC+93, Assumption 1] nicht die Konvergenz der Iterierten, die dort vorausgesetzt wird. Dies ist besonders handlich, wenn man nur die Konvergenz der Ableitungen untersuchen will.

Wir werden unter einer vergleichbaren Bedingung an den Startwert wie in Satz 2.3 die Konvergenz der Iterierten y_k nachweisen.

(iv) Aus beiden Annahmen erhalten wir eine einfache Abschätzung des Vorkonditionierers:

$$\begin{aligned}\|P_k\| &= \left\|(\mathrm{Id} - D_k) f_y(x, y_k)^{-1}\right\| \\ &\leq \left\|f_y(x, y_k)^{-1}\right\| \|\mathrm{Id} - D_k\| \\ &\overset{(2.8)}{\leq} M(1+\delta) \\ &\overset{(2.14)}{\leq} 2M. \end{aligned} \tag{2.16}$$

Im Folgenden werden zunächst die verwendeten Konvergenzbegriffe definiert, anschließend die Resultate zur Konvergenz der differenzierten Iteration dargestellt.

2.3 Konvergenzbegriffe

Folgende Definitionen gehen auf [OR00] zurück und lassen sich in einem beliebigen normiertem Raum $(X, \|\cdot\|)$.

Definition 2.9. *(vergleiche [OR00, Def. 9.1.1. und S. 285f])*
Sei $\{x_k\}$ eine Folge in X mit $x_k \to x_*$. Dann heißt für $p \in \mathbb{N}$

$$Q_p(x_k) = \begin{cases} 0, & x_k = x_* \text{ für fast alle } k \\ \limsup_{k\to\infty} \frac{\|x_{k+1} - x_*\|}{\|x_k - x_*\|^p}, & x_k \neq x_* \text{ für fast alle } k \\ \infty, & \text{sonst} \end{cases}$$

der Q-Faktor der Ordnung p *der Folge $\{x_k\}$. Die Folge heißt* Q-linear *konvergent, falls $0 < Q_1(x_k) < 1$ und* Q-superlinear *konvergent, falls sogar $Q_1(x_k) = 0$. Gilt darüber hinaus sogar $Q_2(x_k) < \infty$, so heißt $\{x_k\}$* Q-quadratisch *konvergent.*

Definition 2.10. *(vergleiche [OR00, Def. 9.2.1. und p. 290f])*
Sei $\{x_k\}$ eine Folge in X mit $x_k \to x_*$. Dann heißt für $p \in \mathbb{N}$

$$R_p(x_k) = \begin{cases} \limsup_{k\to\infty} \|x_k - x_*\|^{1/k}, & p = 1 \\ \limsup_{k\to\infty} \|x_k - x_*\|^{1/p^k}, & p > 1 \end{cases}$$

der R-Faktor der Ordnung p *der Folge. Die Folge heißt* R-linear *konvergent, falls $0 < R_1(x_k) < 1$ und* R-quadratisch *konvergent, falls $0 < R_2(x_k) < 1$.*

Bemerkung 2.11.

(i) Eine stärkere Formulierung der Q-linearen Konvergenz stellt die Bedingung

$$\|x_{k+1} - x_*\| \leq \alpha \|x_k - x_*\|, \quad 0 < \alpha < 1$$

dar. Im Gegensatz zur Q-linearen Konvergenz hat man in diesem Fall eine gleichmässige Abnahme des Fehlers, während bei der Q-linearen Konvergenz einzelne Iterierte einen größeren Fehler als ihre Vorgänger aufweisen können. Analoges gilt für die Q-quadratische Konvergenz und die stärkere Bedingung $\|x_{k+1} - x_*\| \leq \alpha \|x_k - x_*\|^2$.

(ii) Ist eine Folge Q-linear konvergent, so konvergiert sie auch R-linear, denn: Bezeichne $\nu_k := \|x_k - x_*\|$. *Wegen*

$$\limsup_{k\to\infty} \frac{\nu_{k+1}}{\nu_k} \leq \alpha < 1,$$

existiert zu jedem $\varepsilon > 0$ *ein* $k_0 \in \mathbb{N}$ *mit*

$$\frac{\nu_{k+1}}{\nu_k} \leq \alpha + \varepsilon \quad \forall k \geq k_0.$$

Sei ε *so klein, dass* $\alpha + \varepsilon < 1$. *Dann gilt für alle* $k \geq k_0$

$$\nu_k \leq \nu_{k-1}(\alpha + \varepsilon) \leq \cdots \leq \nu_{k_0}(\alpha + \varepsilon)^{k-k_0},$$

also auch

$$\sqrt[k]{\nu_k} \leq (\alpha + \varepsilon) \sqrt[k]{\frac{\nu_{k_0}}{(\alpha + \varepsilon)^{k_0}}}.$$

Daraus folgt nach Anwenden des Limes superior

$$\limsup_{k\to\infty} \sqrt[k]{\nu_k} \leq (\alpha + \varepsilon) \limsup_{k\to\infty} \sqrt[k]{\frac{\nu_{k_0}}{(\alpha + \varepsilon)^{k_0}}} = \alpha + \varepsilon < 1,$$

also die R-lineare Konvergenz von ν_k. *Da* ε *beliebig war, gilt insbesondere*

$$\limsup_{k\to\infty} \sqrt[k]{\nu_k} \leq \alpha.$$

Für eine Veralllgemeinerung dieser Tatsache siehe [OR00, prop. 9.3.2.].

2.4 Konvergenz des Basisverfahrens

Es ist offensichtlich, dass die Folge $\{y_k'\}$ der Ableitungen nur konvergieren kann, wenn auch die Folge $\{y_k\}$ der Iterierten konvergiert. Während in [GBC+93] die Konvergenz der Folge $\{y_k\}$ vorausgesetzt wurde, soll hier unter einer zusätzlichen Lokalitätsbedingung die Konvergenz gezeigt werden.

Annahme 2.12 (Lokalität)**.** *Es gelte* $\gamma := \delta + LM\varrho < 1$.

Dies ist wegen $\delta < 1$ durch Verkleinern der betrachteten Umgebung von y_* immer möglich.

Bezeichnung 2.13. *Bezeichne*

$$\varrho_k := \|y_k - y_*\|$$

den absoluten Fehler der Iterierten nach der k-ten Iteration.

Satz 2.14. *Unter den Annahmen 2.5, 2.7 und 2.12 sowie der Bedingung*

$$\|y_0 - y_*\| = \varrho_0 < \varrho \tag{2.17}$$

wird durch (2.4) *eine Folge* $\{y_k\}$ *in* $\mathbb{B}_\varrho(y_*)$ *definiert mit* $y_k \to y_*$*. Es gilt die Fehlerabschätzung*

$$\varrho_{k+1} \leq \gamma\varrho_k, \tag{2.18}$$

$$\varrho_k \leq \gamma^k\varrho. \tag{2.19}$$

Ferner ist y_* *die einzige Nullstelle von* $f(x,\cdot)$ *in* $\mathbb{B}_\varrho(y_*)$*.*

Gilt zusätzlich $\delta_k \leq d\varrho_k$ *für ein* $d > 0$*, so konvergiert die Folge* $\{y_k\}$ *sogar Q-quadratisch.*

Beweis. Seien $y_1, \ldots, y_k \in \mathbb{B}_\varrho(y_*)$. Subtrahiert man y_* von (2.4) ergibt sich

$$\begin{aligned} y_{k+1} - y_* &= y_k - P_k f(x, y_k) - y_* \\ &= y_k - y_* - P_k f_y(x, y_k)(y_k - y_*) + P_k f_y(x, y_k)(y_k - y_*) \\ &\quad - P_k f(x, y_k) \\ &= \Big(\mathrm{Id} - P_k f_y(x, y_k)\Big)(y_k - y_*) - P_k\Big(f(x, y_k) - f_y(x, y_k)(y_k - y_*)\Big) \\ &\overset{(2.13)}{=} D_k(y_k - y_*) + r_k \end{aligned}$$

wobei

$$r_k := -P_k\Big(f(x,y_k) - f_y(x,y_k)(y_k - y_*)\Big).$$

Mit dem Mittelwertsatz in Integralform (siehe z.B. [AE98, Kapitel VII, Theorem 3.10]) lässt sich r_k schreiben als

$$\begin{aligned}
r_k &= -P_k\Big(f(x,y_k) - f_y(x,y_k)(y_k - y_*)\Big)\\
&= -P_k\Big(f(x,y_k) - \underbrace{f(x,y_*)}_{=0} - f_y(x,y_k)(y_k - y_*)\Big)\\
&= -P_k\left(\Big[f\Big(x, y_* + t(y_k - y_*)\Big)\Big]_{t=0}^{1} - f_y(x,y_k)(y_k - y_*)\right)\\
&= -P_k\int_0^1 \Big(f_y\Big(x, y_* + t(y_k - y_*)\Big) - f_y(x,y_k)\Big)(y_k - y_*)\,dt
\end{aligned}$$

und damit gilt nach Anwenden der Norm unter Beachtung der Monotonie des Integrals

$$\begin{aligned}
\|r_k\| &\le \|P_k\| \int_0^1 \Big\|f_y\Big(x, y_* + t(y_k - y_*)\Big) - f_y(x,y_k)\Big\|\,\|y_k - y_*\|\,dt\\
&\overset{(2.16)}{\le} 2M\varrho_k \int_0^1 \Big\|f_y\Big(x, y_* + t(y_k - y_*)\Big) - f_y(x,y_k)\Big\|\,dt\\
&\overset{(2.9)}{\le} 2M\varrho_k \int_0^1 L\Big\|y_* - y_k + t(y_k - y_*)\Big\|\,dt\\
&= 2LM\varrho_k^2 \int_0^1 (1-t)\,dt\\
&= LM\varrho_k^2.
\end{aligned}$$

Zusammen mit (2.14) erhalten wir daraus

$$\begin{aligned}
\varrho_{k+1} &= \|y_{k+1} - y_*\|\\
&\le \|D_k\|\,\|y_k - y_*\| + \|r_k\|\\
&\le \delta_k\varrho_k + LM\varrho_k^2. \qquad (2.20)
\end{aligned}$$

Dies impliziert unter Beachtung von Annahme 2.12 und der Induktionsvoroussetzung $y_k \in \mathbb{B}_\varrho(y_*)$

$$\varrho_{k+1} \le (\delta + LM\varrho)\varrho_k = \gamma\varrho_k \le \gamma\varrho < \varrho,$$

also $y_{k+1} \in \mathbb{B}_\varrho(y_*)$ und (2.18).
Wegen $\gamma < 1$ gilt außerdem $\varrho_k \le \gamma^k\varrho \to 0$, also $y_k \to y_*$. Unter der zusätzlichen Bedingung $\delta_k \le d\varrho_k$ erhalten wir die Abschätzung

$$\varrho_{k+1} \le (d + LM)\,\varrho_k^2$$

und damit die Q-quadratische Konvergenz:

$$\limsup_{k\to\infty} \frac{\|y_{k+1} - y_*\|}{\|y_k - y_*\|^2} \le d + LM.$$

Ist $\tilde{y}_*$ eine zweite Lösung von $f(x,y) = 0$ in $\mathbb{B}_\varrho(y_*)$, so gilt

$$\begin{aligned}
\|y_* - \tilde{y}_*\| &= \Big\| f_y(x,y_*)^{-1}\Big(f_y(x,y_*)(\tilde{y}_* - y_*)\Big)\Big\| \\
&= \Big\| f_y(x,y_*)^{-1}\Big(f(x,\tilde{y}_*) - f(x,y_*) - f_y(x,y_*)(\tilde{y}_* - y_*)\Big)\Big\| \\
&\le M\Big\| f(x,\tilde{y}_*) - f(x,y_*) - f_y(x,y_*)(\tilde{y}_* - y_*)\Big\| \\
&= M\left\| \Big[f\Big(x, y_* + s(\tilde{y}_* - y_*)\Big)\Big]_{s=0}^{1} - f_y(x,y_*)(\tilde{y}_* - y_*)\right\| \\
&= M\left\| \int_0^1 \Big(f_y\Big(x, \tilde{y}_* + t(y_* - \tilde{y}_*)\Big) - f_y(x,y_*)\Big)(\tilde{y}_* - y_*)\, dt\right\| \\
&\le LM \|y_* - \tilde{y}_*\|^2 \\
&\le LM\varrho \|y_* - \tilde{y}_*\|.
\end{aligned}$$

Dies ist aber wegen $LM\varrho \le \gamma < 1$ nur möglich, wenn $y_* = \tilde{y}_*$. □

Bemerkung 2.15. *Ersetzt man* (2.9) *durch* (2.12)*, so kann das Restglied* r_k *wie in Bemerkung 2.6(iii) beschrieben abgeschätzt werden durch*

$$\begin{aligned}
\|r_k\| &\le \|P_k\| \int_0^1 \Big\| f_y\Big(x, y_* + t(y_k - y_*)\Big) - f_y(x,y_k)\Big\| \, \|y_k - y_*\| \, dt \\
&\le 3\tilde{L}M\varrho_k^2.
\end{aligned}$$

Zum Nachweis der Konvergenz muss dann statt Annahme 2.12

$$\tilde{\gamma} := \delta + 3\tilde{L}M\varrho < 1$$

gefordert werden. Die Eindeutigkeit ergibt sich dann gleichermaßen wegen $3\tilde{L}M\varrho < 1$.

Beweis von Satz 2.3

Beweis. Im Fall des Newton-Verfahrens $P_k = f_y(x,y_k)^{-1}$ ist $\delta = 0$ und es gilt (2.7a) mit $\omega = 2LM$. (2.7b) ergibt sich mit $\gamma = LM\varrho$ wegen

$$LM\varrho_{k+1} \le (LM\varrho_k)^2 \le \gamma^{2^{k+1}}$$

für $\gamma < 1$.

Man beachte dabei, dass die Voraussetzung (2.8) in diesem Fall abgeschwächt werden kann. Die Teilaussage $\|f_y(x,y)^{-1}\|$ beschränkt wurde im Beweis von Satz 2.14 nicht benötigt, ebenso nicht die Beschränktheit der Jacobi-Matrix f_x. Bei der Abschätzung des Fehlerterms $\|r_k\|$ genügt wegen $P_k = f_y(x,y_k)^{-1}$ auch die etwas schwächere Bedingung (2.6), so dass hier Annahme 2.5 komplett durch (2.6) ersetzt werden kann.

Annahme 2.7 ist trivialerweise erfüllt und Annahme 2.12 entspricht zusammen mit (2.17) gerade der Voraussetzung $\|y_0 - y_*\| < \frac{2}{\omega}$. □

Bemerkung 2.16. *Unter Beachtung der Bemerkungen 2.6(iii) und 2.15 kann Annahme 2.5 sogar noch abgeschwächt werden zu*

$$\Big\| f_y(x,y)^{-1}\Big(f_y(x,y) - f_y(x,y_*)\Big)\Big\| < \tilde{\omega}\,\|y - y_*\|$$

für alle $y \in \mathbb{B}_\varrho(y_*)$.

Nachdem sichergestellt ist, dass unter einigen Annahmen durch (2.4) konvergente Verfahren beschrieben werden, werden nun das vollständig und das vereinfacht differenzierte Verfahren im Sinne von [GBC+93] vorgestellt.

2.5 Vollständig differenziertes Verfahren

Aus (2.4) erhält man durch vollständige Differentiation nach den unabhängigen Variablen x die Iterationsvorschrift

$$y'_{k+1} = y'_k - P_k\Big(f_y(x,y_k)y'_k + f_x(x,y_k)\Big) - P'_k f(x,y_k) \tag{2.21}$$

mit $P'_k = \frac{d}{dx}P_k$ der totalen Ableitung von P_k nach x. Subtraktion der tatsächlichen Ableitung

$$y'_* = -f_y(x,y_*)^{-1} f_x(x,y_*)$$

liefert die Fixpunktform

$$y'_{k+1} - y'_* = D_k(y'_k - y'_*) - r'_k - P'_k f(x,y_k) \tag{2.22}$$

mit dem Restglied

$$r'_k := P_k\Big(f_y(x,y_k)y'_* + f_x(x,y_k)\Big). \tag{2.23}$$

Da selbstverständlich die Konvergenz des Basisverfahrens Vorausetzung für die Konvergenz der Ableitungen ist, also insbesondere $f(x,y_k) \to 0$, verschwindet der letzte Term in (2.21) und (2.22), sofern P'_k beschränkt ist.

Benötigt wird aber nur die Konvergenz des Produktes gegen 0. Die durch (2.21) gegebene Folge $\{y'_k\}$ kann in diesem Fall nur den Grenzwert y'_* haben, falls sie überhaupt konvergiert. Unter der Voraussetzung $y_k \to y_*$ konvergiert das Restglied gegen null und (2.22) erweckt den Eindruck einer Kontraktion.

Bemerkung 2.17 (Aufwand)**.** *Diese Variante ist zwar am einfachsten zu implementieren, stellt aber auch im Hinblick auf die Effizienz den schlechtesten Fall dar. Da die gesamte Iterationsvorschrift* (2.4) *abgeleitet wird, müssen im Fall des Newton-Verfahrens die Ableitungswerte bei der Zerlegung von $f_y(x, y_k)$ einbezogen werden, das heißt jeder Eintrag von f_y ist eine zusammengesetzte Variable, die alle Ableitungstensoren bis zur gewünschten Ordnung enthält. Wird dann beispielsweise die LR-Zerlegung mit $\mathcal{O}(m^3)$ Additionen und Multiplikationen angewendet, muss jede dieser Operationen auf alle Tensoren angewendet werden, was den Aufwand enorm erhöht.*

2.6 Vereinfacht differenziertes Verfahren

Nach dem Satz über implizite Funktionen ist die Ableitung y'_* bereits durch die ersten Ableitungen von f bestimmt. Wird aber beispielsweise für das Newton-Verfahren mit $P_k = f_y(x, y_k)^{-1}$ das vollständig differenzierte Verfahren verwendet, kommen auch die zweiten Ableitungen von f ins Spiel, denn zum Beispiel im skalaren Fall $m = n = 1$ ist

$$\begin{aligned}\frac{d}{dx}\Phi_k(x, y_k) &= \frac{d}{dx}\Big(y_k - f_y(x, y_k)^{-1} f(x, y_k)\Big) \\ &= y'_k - f_y(x, y_k)^{-1}\Big(f_x(x, y_k) + f_y(x, y_k)y'_k\Big) \\ &\qquad + f_y(x, y_k)^{-1}\Big(f_{yx}(x, y_k) + f_{yy}(x, y_k)y'_k\Big) f_y(x, y_k)^{-1} f(x, y_k).\end{aligned}$$

Dies betrifft natürlich alle Verfahren, deren Vorkonditionierer in irgendeiner Weise von den Ableitungen von f abhängt.

Nun kann es sein, dass eine gegebene Funktion gar keine zweiten Ableitungen besitzt, nach dem Satz über implizite Funktionen sollten wir dennoch in der Lage sein, die ersten impliziten Ableitungen zu berechnen. Deswegen betrachten wir im Folgenden eine Variante des vollständig differenzierten Verfahrens, bei dem nur die ersten Ableitungen von f zum Tragen kommen. Wie beim vollständig differenzierten Verfahren wird (2.4) differenziert, dabei allerdings angenommen, dass P_k konstant bezüglich x ist und damit $\frac{d}{dx}P_k = 0$ gilt. Es ergibt sich

$$\tilde{y}'_{k+1} = \tilde{y}'_k - P_k\Big(f_y(x, y_k)\tilde{y}'_k + f_x(x, y_k)\Big)$$

mit der Fixpunktform

$$\tilde{y}'_{k+1} - y'_* = D_k(\tilde{y}'_k - y'_*) + r'_k.$$

Wir können dieses Vorgehen rechtfertigen, indem wir (2.4) als einen Schritt einer Picard-Iteration zum nichtlinearen System

$$f_k(x, y) := P_k f(x, y) = 0$$

betrachten. Da P_k nichtsingulär ist, hat jedes f_k genau die gleiche Nullstellenmenge wie die Ausgangsfunktion. Insbesondere sind die implizite Funktion $g(x)$ und ihre Ableitungen nicht von der Folge der P_k abhängig.

Da dieses Verfahren durch die Vorgabe $\frac{d}{dx}P_k := 0$ direkt aus dem vollständig differenzierten Verfahren hervorgeht, können die folgenden Resultate ungeändert übernommen werden.

2.7 Konvergenz der Ableitungen

Bezeichnung 2.18. *Bezeichnen*

$$\begin{aligned}\mu_k &:= \left\|y'_k - y'_*\right\|, \\ \eta_k &:= \left(Lc_1 + \|P'_k\|\right)\varrho_k, \text{ wobei } c_1 := 2\left(M^2 + 1\right)\end{aligned}$$

und

$$f'(x, y_k, y'_k) := f_y(x, y_k)y'_k + f_x(x, y_k). \tag{2.24}$$

Satz 2.19 (Fehlerabschätzung)**.** *(vergleiche [GBC+93, Lemma 1]) Unter den Annahmen 2.5 und 2.7 folgt*

$$\mu_k \leq \frac{1}{1-\delta}\left\|P_k f'(x, y_k, y'_k)\right\| + \frac{1}{2}LMc_1\varrho_k, \tag{2.25}$$

$$\mu_{k+1} \leq \delta_k\mu_k + M\eta_k \tag{2.26}$$

und

$$\|r'_k\| \leq LMc_1\varrho_k. \tag{2.27}$$

Beweis. Mit (2.24) können wir schreiben

$$y'_k - y'_* = f_y(x, y_k)^{-1}f'(x, y_k, y'_k) - \left(f_y(x, y_k)^{-1}f_x(x, y_k) + y'_*\right)$$

und erhalten nach Anwenden der Norm und der Dreiecksungleichung

$$\begin{aligned}
\mu_k &= \|y'_k - y'_*\| \\
&= \left\| f_y(x,y_k)^{-1} f'(x,y_k,y'_k) - \left(f_y(x,y_k)^{-1} f_x(x,y_k) + y'_* \right) \right\| \\
&\leq \left\| f_y(x,y_k)^{-1} f'(x,y_k,y'_k) \right\| + \left\| f_y(x,y_k)^{-1} f_x(x,y_k) + y'_* \right\| \\
&= \left\| f_y(x,y_k)^{-1} P_k^{-1} P_k f'(x,y_k,y'_k) \right\| + \left\| f_y(x,y_k)^{-1} f_x(x,y_k) + y'_* \right\| \\
&\leq \left\| f_y(x,y_k)^{-1} P_k^{-1} \right\| \left\| P_k f'(x,y_k,y'_k) \right\| + \left\| f_y(x,y_k)^{-1} f_x(x,y_k) + y'_* \right\| .
\end{aligned} \tag{2.28}$$

Den ersten Faktor des ersten Terms können wir mit dem Störungslemma [OR00, 2.3.2] abschätzen und erhalten

$$\left\| f_y(x,y_k)^{-1} P_k^{-1} \right\| = \left\| (\mathrm{Id} - D_k)^{-1} \right\| \leq \frac{1}{1 - \|D_k\|} \leq \frac{1}{1-\delta}$$

unter Beachtung von (2.14).

Der zweite Term von (2.28) ist gerade der Fehler bei der Approximation von y'_* durch Einsetzen von y_k in die Gleichung (2.3). Diesen können wir allgemeiner für alle y mit $\|y - y_*\| < \varrho$ mittels Annahme 2.5 abschätzen durch

$$\begin{aligned}
&\left\| f_y(x,y)^{-1} f_x(x,y) + y'_* \right\| \\
&\quad \overset{(2.3)}{=} \left\| f_y(x,y)^{-1} f_x(x,y) - f_y(x,y_*)^{-1} f_x(x,y_*) \right\| \\
&\quad = \Big\| f_y(x,y)^{-1} f_x(x,y) - f_y(x,y)^{-1} f_x(x,y_*) + \\
&\qquad + f_y(x,y)^{-1} f_x(x,y_*) - f_y(x,y_*)^{-1} f_x(x,y_*) \Big\| \\
&\quad \leq \left\| f_y(x,y)^{-1} \Big(f_x(x,y) - f_x(x,y_*) \Big) \right\| + \\
&\qquad + \left\| \Big(f_y(x,y)^{-1} - f_y(x,y_*)^{-1} \Big) f_x(x,y_*) \right\| \\
&\quad = \left\| f_y(x,y)^{-1} \Big(f_x(x,y) - f_x(x,y_*) \Big) \right\| + \\
&\qquad + \left\| f_y(x,y)^{-1} \Big(f_y(x,y_*) - f_y(x,y) \Big) f_y(x,y_*)^{-1} f_x(x,y_*) \right\|
\end{aligned}$$

$$
\begin{aligned}
&\leq \underbrace{\left\|f_y(x,y)^{-1}\right\|}_{\leq M} \underbrace{\left\|f_x(x,y)-f_x(x,y_*)\right\|}_{\leq \|\partial f(x,y)-\partial f(x,y_*)\| \leq L\|y-y_*\|} + \\
&\quad + \underbrace{\left\|f_y(x,y)^{-1}\right\|}_{\leq M} \underbrace{\left\|f_y(x,y_*)-f_y(x,y)\right\|}_{\leq L\|y-y_*\|} \underbrace{\left\|f_y(x,y_*)^{-1}\right\|}_{\leq M} \underbrace{\left\|f_x(x,y_*)\right\|}_{\leq M} \\
&\leq LM\,\|y-y_*\| + LM^3\,\|y-y_*\| \\
&\leq LM(M^2+1)\,\|y-y_*\| \\
&= \frac{1}{2}LMc_1\,\|y-y_*\|\,.
\end{aligned}
$$

Zusammen ist also

$$
\begin{aligned}
\mu_k &\leq \left\|f_y(x,y_k)^{-1}P_k^{-1}\right\| \left\|P_k f'(x,y_k,y_k')\right\| + \\
&\qquad + \left\|f_y(x,y_k)^{-1}f_x(x,y_k) - f_y(x,y_*)^{-1}f_x(x,y_*)\right\| \\
&\leq \frac{1}{1-\delta}\,\|P_k f'(x,y_k,y_k')\| + \frac{1}{2}LMc_1\,\|y_k-y_*\|
\end{aligned}
$$

und damit ist die erste Ungleichung gezeigt.

Die dritte Ungleichung erhält man aus (2.23) durch

$$
\begin{aligned}
\|r_k'\| &= \left\|P_k\left(f_y(x,y_k)y_*' + f_x(x,y_k)\right)\right\| \\
&\leq \|P_k\|\left\|f_y(x,y_k)y_*' + f_x(x,y_*) + f_x(x,y_k) - f_x(x,y_*)\right\| \\
&\overset{(2.3)}{=} \|P_k\|\left\|\Big(f_y(x,y_k) - f_y(x,y_*)\Big)y_*' + f_x(x,y_k) - f_x(x,y_*)\right\| \\
&\leq \|P_k\|\left(\left\|f_y(x,y_k) - f_y(x,y_*)\right\|\,\|y_*'\| + \left\|f_x(x,y_k) - f_x(x,y_*)\right\|\right) \\
&\overset{(2.11)}{\leq} \|P_k\|\left(M^2\left\|f_y(x,y_k) - f_y(x,y_*)\right\| + \left\|f_x(x,y_k) - f_x(x,y_*)\right\|\right) \\
&\leq \|P_k\|\,(M^2+1)\left\|\partial f(x,y_k) - \partial f(x,y_*)\right\| \\
&\overset{(2.9)}{\leq} \|P_k\|\,L(M^2+1)\,\|y_k-y_*\| \\
&\overset{(2.16)}{\leq} 2ML(M^2+1)\,\|y_k-y_*\| \\
&= LMc_1\varrho_k.
\end{aligned}
$$

Schließlich hat man ausgehend von (2.22) wegen (2.14) und (2.10) mit der

eben gezeigten Abschätzung auch

$$\begin{aligned}\mu_{k+1} &= \left\| D_k(y'_k - y'_*) - r'_k - P'_k f(x, y_k) \right\| \\ &\leq \|D_k\|\, \mu_k + \|r'_k\| + \|P'_k\|\, \|f(x, y_k)\| \\ &\leq \delta_k \mu_k + LMc_1 \varrho_k + \|P'_k\|\, M \varrho_k \\ &= \delta_k \mu_k + M\Big(Lc_1 + \|P'_k\|\Big) \varrho_k \\ &= \delta_k \mu_k + M\eta_k.\end{aligned}$$

□

Bemerkung 2.20. *Die Abschätzung* (2.26) *legt nahe, dass der Fehler der Iterierten* y'_k *aus* (2.21) *dem der Iterierten* y_k *aus* (2.4) *um eine Iteration hinterherhinkt.*

Bis jetzt genügte es, die Nichtsingularität und Differenzierbarkeit der Vorkonditionierer P_k zu fordern. Inbesondere gilt die eben gezeigte Fehlerabschätzung für alle iterativen Verfahren, die (2.14) erfüllen. Um aus dieser Fehlerabschätzung einen Konvergenzbeweis zu gewinnen, wird allerdings eine starke Abschätzung von $\|P'_k\|$ benötigt. Wie schon in Abschnitt 2.5 bemerkt, können wir mit Konvergenz rechnen, wenn P'_k nicht zu stark *wächst*. Die im folgenden Satz gestellte Bedingung an die Norm von P'_k fordert allerdings ein zügiges Abfallen dieser Norm auf ein gewisses Niveau:

$$\|P'_k\| = \|\partial_y P_k y'_k + \partial_x P_k\| \leq c_2(1 + \mu_k) \tag{2.29}$$

Beim vereinfacht differenzierten Verfahren ist sie allerdings trivialerweise mit $c_2 = 0$ erfüllt.

Satz 2.21 (Konvergenz)**.** *(vergleiche [GBC+93, Proposition 1])*
Es gelten die Annahmen 2.5, 2.7 und 2.12, sowie

$$\|y_0 - y_*\| \leq \varrho.$$

Dann folgt aus (2.29)*, dass die durch* (2.21) *definierte Folge* $\{y'_k\}$ *R-linear gegen* y'_* *konvergiert. Ebenso konvergiert für jedes hinreichend kleine* $\alpha > 0$ *die Sobolev-Norm*

$$\|y_k - y_*\| + \alpha \|y'_k - y'_*\|$$

Q-linear gegen null.

Ist außerdem

$$\delta_k \leq d\varrho_k = d\, \|y_k - y_*\|$$

für ein $d > 0$, so konvergiert die Folge $\{y'_k\}$ sogar R-quadratisch:

$$\limsup_{k\to\infty} \|y'_k - y'_*\|^{1/2^k} < 1.$$

Beweis. Einsetzen der zusätzlichen Annahme (2.29) in die Definition von η_k ergibt

$$\begin{aligned}\eta_k &= (Lc_1 + \|P'_k\|)\varrho_k \\ &\le \Big(Lc_1 + c_2(1+\mu_k)\Big)\varrho_k \\ &= (Lc_1 + c_2)\varrho_k + c_2\mu_k\varrho_k,\end{aligned}$$

und damit folgt aus (2.26)

$$\begin{aligned}\mu_{k+1} &\le \delta_k\mu_k + M\eta_k \\ &\le \delta_k\mu_k + M(Lc_1 + c_2)\varrho_k + Mc_2\mu_k\varrho_k \\ &= (\delta_k + Mc_2\varrho_k)\mu_k + c_3\varrho_k \text{ mit } c_3 = M(Lc_1 + c_2).\end{aligned}$$

Zusammen mit $\varrho_{k+1} \le \delta_k\varrho_k + LM\varrho_k^2$ aus (2.20) ergibt sich für jedes $\alpha > 0$

$$\begin{aligned}\frac{\varrho_{k+1} + \alpha\mu_{k+1}}{\varrho_k + \alpha\mu_k} &\le \frac{(\delta_k + \alpha c_3 + LM\varrho_k)\varrho_k + \alpha(\delta_k + Mc_2\varrho_k)\mu_k}{\varrho_k + \alpha\mu_k} \\ &\le \delta_k + \alpha c_3 + M(L + c_2)\varrho_k.\end{aligned}$$

Wegen $\varrho_k \to 0$ und (2.15) hat die rechte Seite $\delta_* + \alpha c_3$ als größten Häufungspunkt. Damit konvergiert die Sobolev-Norm $\|y_k - y_*\| + \alpha\|y'_k + y'_*\|$ Q-linear gegen 0, falls $0 < \alpha < (1 - \delta_*)/c_3$. Insbesondere ist nach Bemerkung 2.11(ii) die Sobolev-Norm auch R-linear konvergent mit dem R-Faktor $\delta_* + \alpha c_3$. Wegen

$$\limsup_{k\to\infty} \sqrt[k]{\mu_k} = \limsup_{k\to\infty} \sqrt[k]{\alpha\mu_k} \le \limsup_{k\to\infty} \sqrt[k]{\varrho_k + \alpha\mu_k} \le \delta_* + \alpha c_3$$

gilt dies auch für die Folge $\{\mu_k\}$. Also ist der R-Faktor der Folge $\{\mu_k\}$ für jedes hinreichend kleine α beschränkt durch $\delta_* + c_3\alpha$, also auch nicht größer als δ_*.

Schließlich folgt mit der zusätzlichen Bedingung an δ_k, dass

$$\begin{aligned}\mu_{k+1} &\le (\delta_k + Mc_2\varrho_k)\mu_k + c_3\varrho_k \\ &\le (d\varrho_k + Mc_2\varrho_k)\mu_k + c_3\varrho_k \\ &\le (d + Mc_2\mu_k + c_3)\varrho_k \\ &\le c_4\varrho_k \text{ für ein } c_4 > 0,\end{aligned}$$

das heißt, die konvergente Folge $\{\mu_k\}$ ist beschränkt durch ein Vielfaches der quadratisch konvergenten Folge $\{\varrho_{k-1}\}$. □

Bemerkung 2.22.

(i) In der letzten Abschätzung $\mu_{k+1} \leq c_4 \varrho_k$ zeigt sich wieder ganz deutlich, dass man mit einem Nachhinken der Ableitugen rechnen muss. Dies wird schließlich auch durch die Beispiele in Abschnitt 2.9 bestätigt.

(ii) Wegen $\delta_k = 0$ ist die zusätzliche Bedingung für das Newton-Verfahren erfüllt und man erhält die quadratische Konvergenz für die Ableitungen.

(iii) Es sei noch explizit darauf hingewiesen, dass keinerlei Voraussetzungen an den Startwert y'_0 gemacht wurden.

2.8 Nachträglich differenzierte Verfahren

Das nachträglich differenzierte Verfahren beruht auf der Beobachtung, dass die Korrekturen $\Delta y'_k = y'_{k+1} - y'_k$ der Ableitungswerte erst dann sinnvoll werden, wenn die Iterierten y_k selber nah genug an der Lösung y_* sind. Da die Jacobimatrix $\partial_y f(x, y_k)$ nach der Konvergenz $y_k \to y_*$ keine Veränderungen mehr erfährt, muss sie auch nicht erneut zerlegt werden. Die Referenzimplementation nutzt diesen Sachverhalt aus und berechnet bei der Verwendung des nachträglich differenzierten Verfahrens die LR-Zerlegung der Jacobimatrix in der Post-Iteration nur beim ersten Schritt.

2.9 Numerische Beispiele

Für Ableitungstensoren der Ordnung $d \in \mathbb{N}$ verwenden wir die Frobenius-Norm:

$$\|y\|_{(d)} := \sqrt{\sum_{|\mathbf{i}|=d} \frac{\partial^{|\mathbf{i}|} y}{\partial x_1^{\mathbf{i}_1} \dots \partial x_n^{\mathbf{i}_n}}} \tag{2.30}$$

Dabei ist $\mathbf{i} \in \mathbb{N}^d$ ein Multiindex. Da sich alles in endlich-dimensionalen Räumen abspielt, unterscheidet sich die Frobenius-Norm von den Operatornormen (wie sie in den Beweisen verwendet werden) nur um einen Faktor.

Als Abbruchkriterium bietet sich dann eine Sobolev-Norm der Form

$$\|y\|_{\mathcal{S}_o} := \sum_{i=0}^{o} \|y\|_{(i)} \tag{2.31}$$

an, wobei o die maximale Differentiationsordnung bezeichnet.

2.9.1 Polarkoordinaten

Betrachtet werde das Gleichungssystem

$$x - r\sin(\phi) = 0 \tag{2.32a}$$
$$y - r\cos(\phi) = 0, \tag{2.32b}$$

das die Beziehung zwischen kartesischen Koordinaten $(x, y) \in \mathbb{R} \times \mathbb{R}$ und Polarkoordinaten $(r, \phi) \in [0, \infty) \times [0, 2\pi)$ in der Ebene beschreibt. Die Abbildungen 2.1 und 2.2 zeigen den relativen Fehler der Iterierten y_k und Ableitungen bis zur Ordnung 4 bei der Berechnung von $(r(x, y), \phi(x, y))$ mit dem vereinfacht bzw. vollständig differenzierten Verfahren. Als Abbruchkriterium wurde

$$\left\| \begin{pmatrix} \Delta r \\ \Delta\phi \end{pmatrix} \right\|_{\mathcal{S}_4} < 10^{-12}$$

verwendet.

Dabei ist $\|y_k\|_{\mathcal{R},i} := \|y_k - y_*\|_{(i)} / \|y_*\|_{(i)}$, $i = 0, \ldots, 4$, der relative Fehler der Iterierten.

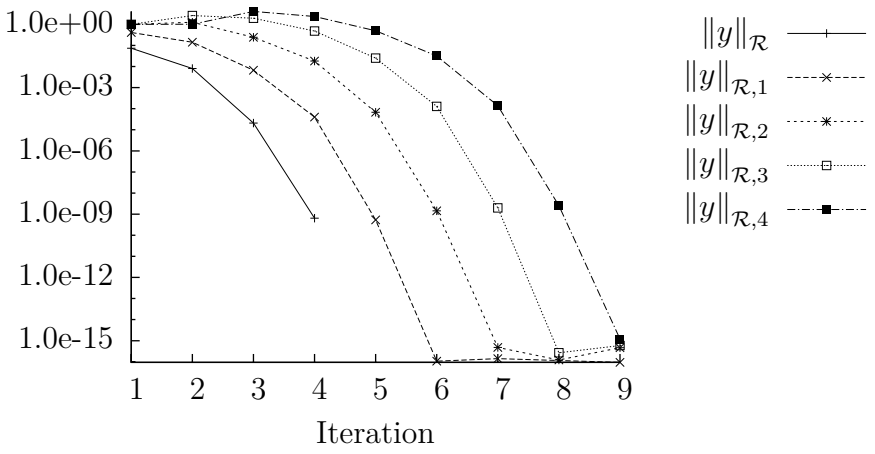

Abbildung 2.1: Relativer Fehler pro Iteration bei der Berechnung von $(r(x, y), \phi(x, y))$ und partiellen Ableitungen bis zur Ordnung 4 aus (2.32) an der Stelle $x = \frac{3}{2}$, $y = \frac{1}{2}$ mit dem vereinfacht differenzierten Verfahren.

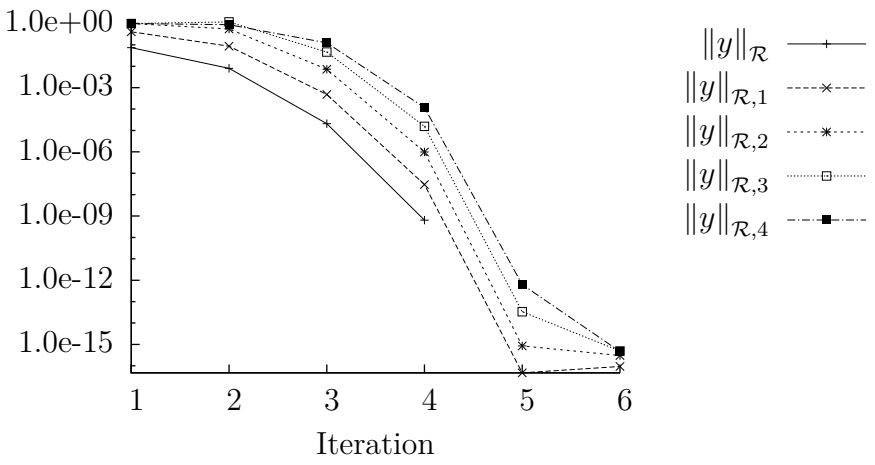

Abbildung 2.2: Analog zu Abbildung 2.1 unter Verwendung des vollständig differenzierten Verfahren.

2.9.2 Ljapunow-Schmidt-Reduktion

Bei der Ljapunow-Schmidt-Reduktion, siehe 3.1, wird eine implizite Funktion berechnet. In Abschnitt 5.2 wird dieser Schritt (mehrfach) als Teilproblem für eine Funktion $\mathbb{R}^{10} \to \mathbb{R}^{10}$ durchgeführt. Abbildungen 2.3 und 2.4 zeigen den Iterationsverlauf bei der ersten Berechnung dieser implizit definierten Funktion im Detail. Das Abbruchkriterium ist hier $\|\Delta y\|_{\mathcal{S}_3} < 10^{-8}$.

Wir sehen auch hier den Effekt des vollständig differenzierten Verfahrens, dass die Ableitungen jeder Ordnung nur eine Iteration nachhinken, während beim vereinfacht differenzierten Verfahren die Iterierten zur Ableitung der Ordnung d um d Iterationen hinterherhinken.

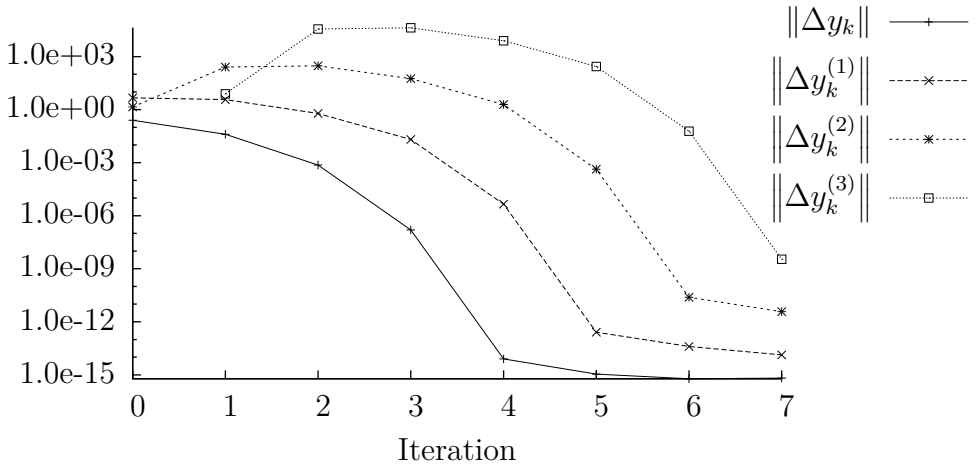

Abbildung 2.3: Norm der Korrektur pro Iteration beim vereinfacht differenzierten Verfahren.

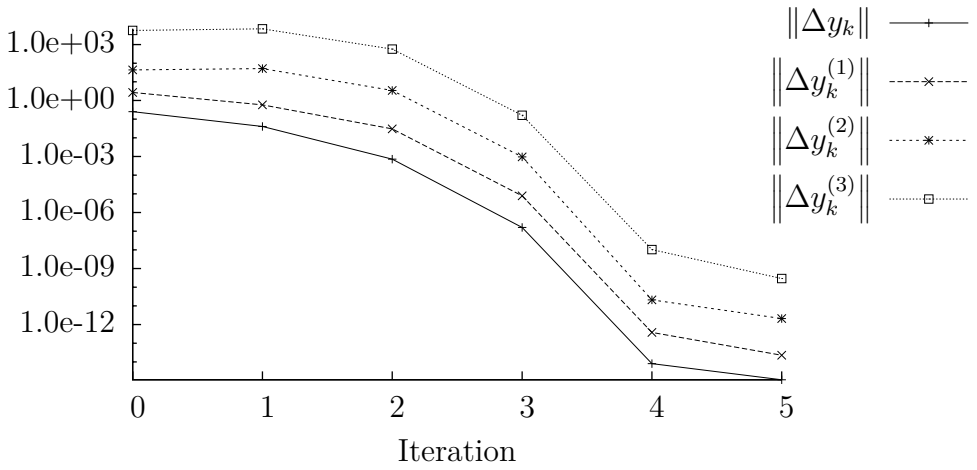

Abbildung 2.4: Norm der Korrektur pro Iteration beim vollständig differenzierten Verfahren.

Kapitel 3

Grundlagen der Singularitäten-Theorie

Nachdem nun alle technischen Voraussetzungen geklärt sind, werden die grundlegenden Begriffe und Sachverhalte zum Thema der singulären Punkte vorgestellt. Dies umfasst die Definition der reduzierten Funktion, der Kontaktäquivalenz und minimal erweiterter Systeme. Schließlich geben wir eine Übersicht über die Klassifikation der Singularitäten bis zur Kodimension 3.

3.1 Begriff der reduzierten Funktion

Sei

$$F : \mathbb{R}^{n+l} \times \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}^n, \quad (x, \tau, \alpha) \mapsto F(x, \tau, \alpha), \tag{3.1}$$

in einer Umgebung von $(x^*, \tau^*, \alpha^*) \in F^{-1}(\{0\})$ unendlich oft stetig differenzierbar und (x^*, τ^*, α^*) ein *singulärer Punkt* von F, es gelte also

$$\operatorname{rang} L = n - k, \quad L := F_x(x^*, \tau^*, \alpha^*) \in \mathbb{R}^{n\times(n+l)}, \quad k \in \mathbb{N}, \tag{3.2}$$

das heißt die Jacobi-Matrix von F bezüglich der Zustandsvariablen x hat in (x^*, τ^*, α^*) den Rangdefekt k. Die Parameter τ heißen *Bifurkationsparameter* oder *Verzweigungsparameter*, die α *Entfaltungsparameter*.

Dann existieren $T \in \mathbb{R}^{(n+l)\times(k+l)}$ und $Z \in \mathbb{R}^{n\times k}$ mit jeweils linear unabhängigen Spalten, so dass

$$LT = 0, \quad Z^\intercal L = 0. \tag{3.3}$$

Die Spalten von T bzw. Z bilden also eine Basis von $\ker L$ bzw. von $(\operatorname{im} L)^\perp$.

Um den regulären Anteil der Zustandsvariablen aus dem System zu eliminieren, definiert man zu $\tilde{T} \in \mathbb{R}^{(n+l)\times(k+l)}$ und $\tilde{Z} \in \mathbb{R}^{n\times k}$ eine Funktion

$$H : \mathbb{R}^{n+l} \times \mathbb{R}^p \times \mathbb{R}^q \times \mathbb{R}^k \times \mathbb{R}^{k+l} \to \mathbb{R}^n \times \mathbb{R}^{k+l}$$

durch

$$H(x, h, \xi, \tau, \alpha) = \begin{pmatrix} F(x, \tau, \alpha) - \tilde{Z}h \\ \tilde{T}^\intercal x - \xi \end{pmatrix}. \tag{3.4}$$

Offenbar gilt für $h^* = 0$ und $\xi^* = \tilde{T}^\intercal x^*$

$$H(x^*, h^*, \xi^*, \tau^*, \alpha^*) = 0$$

und

$$M := H_{x,h}(x^*, h^*, \xi^*, \tau^*, \alpha^*) = \begin{pmatrix} L & -\tilde{Z} \\ \tilde{T}^\intercal & 0 \end{pmatrix}. \tag{3.5}$$

Satz 3.1. *Es sind äquivalent:*

$$M \text{ nichtsingulär} \tag{3.6a}$$

$$\tilde{T}^\intercal T \text{ und } \tilde{Z}^\intercal Z \text{ nichtsingulär} \tag{3.6b}$$

Beweis. Wähle $\hat{T} \in \mathbb{R}^{(n+l)\times(n-k)}$, $\hat{Z} \in \mathbb{R}^{n\times(n-k)}$ so, dass $\begin{pmatrix} \hat{T} & T \end{pmatrix}$ und $\begin{pmatrix} \hat{Z} & Z \end{pmatrix}$ nichtsingulär sind. Unter Beachtung von (3.3) ergibt sich dann:

$$\begin{aligned} \operatorname{rang} M &= \operatorname{rang} \underbrace{\begin{pmatrix} \hat{Z}^\intercal & 0 \\ Z^\intercal & 0 \\ 0 & \mathrm{Id} \end{pmatrix}}_{\text{nichtsingulär}} \begin{pmatrix} L & -\tilde{Z} \\ \tilde{T}^\intercal & 0 \end{pmatrix} \underbrace{\begin{pmatrix} \hat{T} & T & 0 \\ 0 & 0 & \mathrm{Id} \end{pmatrix}}_{\text{nichtsingulär}} \\ &= \operatorname{rang} \begin{pmatrix} \tilde{Z}^\intercal L\hat{T} & 0 & -\hat{Z}^\intercal\tilde{Z} \\ 0 & 0 & -Z^\intercal\tilde{Z} \\ \tilde{T}^\intercal\hat{T} & \tilde{T}^\intercal T & 0 \end{pmatrix} \\ &= \operatorname{rang} \begin{pmatrix} \tilde{Z}^\intercal L\hat{T} & 0 & 0 \\ 0 & 0 & Z^\intercal\tilde{Z} \\ 0 & \tilde{T}^\intercal T & 0 \end{pmatrix} \end{aligned}$$

Da $\hat{Z}^\intercal L\hat{T}$ nach Konstruktion nichtsingulär ist, ergibt sich die Behauptung. □

Bemerkung 3.2. *Unter der Voraussetzung* (3.6b) *gelten*

$$\operatorname{rang} \tilde{T} = k + l, \quad \operatorname{rang} \tilde{Z} = k.$$

Unter der Voraussetzung (3.6a) liefert der Satz über implizite Funktionen die Existenz von glatten Funktionen

$$x : \mathbb{R}^{k+l} \times \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}^{n+l}, \quad h : \mathbb{R}^{k+l} \times \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}^k$$

mit

$$\begin{aligned} x(\xi^*, \tau^*, \alpha^*) &= x^* \\ h(\xi^*, \tau^*, \alpha^*) &= 0 \end{aligned}$$

und

$$H(x(\xi, \tau, \alpha), h(\xi, \tau, \alpha), \xi, \tau, \alpha) \equiv 0 \tag{3.7}$$

oder

$$F(x(\xi, \tau, \alpha), \tau, \alpha) - \tilde{Z} h(\xi, \tau, \alpha) \equiv 0 \tag{3.8a}$$

$$\tilde{T}^\intercal x(\xi, \tau, \alpha) - \xi \equiv 0 \tag{3.8b}$$

jeweils in einer Umgebung von $(\xi^*, \tau^*, \alpha^*)$.

Multiplikation von (3.8a) mit $Z^\intercal$ von links liefert

$$Z^\intercal F(x(\xi, \tau, \alpha), \tau, \alpha) - Z^\intercal \tilde{Z} h(\xi, \tau, \alpha) \equiv 0$$

und man kann nach h auflösen.

Definition 3.3. *Die Funktion*

$$h(\xi, \tau, \alpha) = \left(Z^\intercal \tilde{Z} \right)^{-1} Z^\intercal F(x(\xi, \tau, \alpha), \tau, \alpha) \tag{3.9}$$

heißt eine zum singulären Punkt (x^*, τ^*, α^*) gehörige reduzierte Funktion *von F.*

Folgendes Resultat liefert eine ein-eindeutige Beziehung zwischen den Nullstellen von F und denen einer reduzierten Funktion in einer Umgebung des singulären Punktes.

Satz 3.4. *Für $(x^0, \tau^0, \alpha^0, \xi^0) \in \mathbb{R}^{n+l} \times \mathbb{R}^p \times \mathbb{R}^q \times \mathbb{R}^{k+l}$ aus einer Umgebung von $(x^*, \tau^*, \alpha^*, \xi^*)$ gilt:*

$$\begin{aligned} F(x^0, \tau^0, \alpha^0) &= 0 \\ \xi^0 &= \tilde{T}^\intercal x^0 \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} h(\xi^0, \tau^0, \alpha^0) &= 0 \\ x(\xi^0, \tau^0, \alpha^0) &= x^0 \end{aligned}$$

Beweis. ($\Rightarrow$) Sei zunächst $F(x^0, \tau^0, \alpha^0) = 0,\ \xi^0 = \tilde{T}^\intercal x^0$. Dann ist

$$H(x^0, h, \xi^0, \tau^0, \alpha^0) = 0$$

und wegen der Eindeutigkeit im Satz über implizite Funktionen folgt

$$x(\xi^0, \tau^0, \alpha^0) = x^0,\ h(\xi^0, \tau^0, \alpha^0) = 0.$$

($\Leftarrow$) Sei nun umgekehrt $h(\xi^0, \tau^0, \alpha^0) = 0,\ x^0 = x(\xi^0, \tau^0, \alpha^0)$. Setzt man dies in (3.8) ein, so ergibt sich

$$0 = F(x(\xi^0, \tau^0, \alpha^0), \tau^0, \alpha^0) - \tilde{Z}h(\xi^0, \tau^0, \alpha^0) = F(x^0, \tau^0, \alpha^0)$$

und

$$\tilde{T}^\intercal x(\xi^0, \tau^0, \alpha^0) - \xi^0 = 0 \quad \Rightarrow \quad \xi^0 = \tilde{T}^\intercal x^0.$$

□

Durch Differentiation von (3.8) bezüglich ξ erhält man außerdem

$$\begin{aligned} F_x(x(\xi, \tau, \alpha), \tau, \alpha) x_\xi(\xi, \tau, \alpha) - \tilde{Z} h_\xi(\xi, \tau, \alpha) &\equiv 0, \\ \tilde{T}^\intercal x_\xi(\xi, \tau, \alpha) - \mathrm{Id} &\equiv 0. \end{aligned}$$

Im singulären Punkt gilt also

$$Lx_\xi(\xi^*, \tau^*, \alpha^*) - \tilde{Z} h_\xi(\xi^*, \tau^*, \alpha^*) = 0, \tag{3.10a}$$

$$\tilde{T}^\intercal x_\xi(\xi^*, \tau^*, \alpha^*) - \mathrm{Id} = 0. \tag{3.10b}$$

Aus der ersten Gleichung (3.10a) folgt nach Multiplikation mit $Z^\intercal$ von links wegen (3.3) und (3.6b)

$$h_\xi(\xi^*, \tau^*, \alpha^*) = 0.$$

Deswegen muss also auch $Lx_\xi(\xi^*, \tau^*, \alpha^*) = 0$ gelten und es ergibt sich

$$x_\xi(\xi^*, \tau^*, \alpha^*) = TV \text{ für ein } V \in \mathbb{R}^{(k+l)\times(k+l)}.$$

Mit (3.10b) folgt dann

$$\tilde{T}^\intercal TV - \mathrm{Id} = 0 \Rightarrow V = \left(\tilde{T}^\intercal T\right)^{-1}.$$

Zusammen hat man also

$$x_\xi(\xi^*, \tau^*, \alpha^*) = T\left(\tilde{T}^\intercal T\right)^{-1}. \tag{3.11}$$

Wie beabsichtigt ist die Jacobi-Matrix $h_\xi(\xi^*, \tau^*, \alpha^*)$ die Nullmatrix, die Singularität liegt nun in ihrer reinen Form vor.

3.2 Kontaktäquivalenz

Bei der Konstruktion der reduzierten Funktion wurden die Matrizen $\tilde{T}, \tilde{Z}$ abgesehen von der Forderung (3.6b) beliebig gewählt. Damit ist die reduzierte Funktion nicht eindeutig bestimmt und es stellt sich die Frage, in welcher Weise verschiedene reduzierte Funktionen zusammenhängen. Zum einen müssen glatte Koordinatentransformationen im Bildbereich einer reduzierten Funktion zulässig sein, die in einer Umgebung des singulären Punktes nichtsingulär sind und damit in dieser Umgebung keine neuen Nullstellen erzeugen. Zum anderen ändern auch genügend reguläre parametrisierte Transformationen der Variablen nichts am qualitativen Bild des Nullstellengebildes in einer Umgebung des singulären Punktes. Je allgemeiner die zulässigen Transformationen sind, desto umfangreicher werden die Äquivalenzklassen konkreter Singularitäten ausfallen. Zu beachten ist dabei, dass der reduzierte singuläre Punkt einer reduzierten Funktion nicht mit dem einer Anderen zusammenfallen muss. Ferner beachte man, dass die Glattheit der Ausgangsfunktion F die Glattheit der reduzierten Funktion nach sich zieht.

Im Folgenden wird die Kontaktäquivalenz als lokale Äquivalenzrelation eingeführt.

Definition 3.5. *Zwei $\mathcal{C}^\infty$-Funktionen $h_1, h_2 : \mathbb{R}^{k+l} \times \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}^k$ heißen* kontakt-äquivalent*, falls es $\mathcal{C}^\infty$-Funktionen*

- $S : \mathbb{R}^{k+l} \times \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}^{n\times n}$ *mit* $S(\xi_1^*, \tau^*, \alpha^*)$ *nichtsingulär,*
- $X : \mathbb{R}^{k+l} \times \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}^{k+l}$ *mit* $X(\xi_1^*, \tau^*, \alpha^*) = \xi_2^*$, $X_\xi(\xi_1^*, \tau^*, \alpha^*)$ *nichtsingulär,*
- $\Lambda : \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}^p$ *mit* $\Lambda(\tau^*, \alpha^*) = \tau^*$, $\Lambda_\tau(\tau^*, \alpha^*)$ *nichtsingulär*

gibt, so dass

$$h_1(\xi, \tau, \alpha) \equiv S(\xi, \tau, \alpha) h_2\Big(X(\xi, \tau, \alpha), \Lambda(\tau, \alpha), \alpha\Big)$$

in einer Umgebung von $(\xi_1^*, \tau^*, \alpha^*)$.

Bemerkung 3.6.

(i) Im folgenden wird kurz $h \sim \tilde{h}$ geschrieben, falls h und $\tilde{h}$ kontaktäquivalent sind.

(ii) Diese Definition der Kontaktäquivalenz ist eine leichte Verallgemeinerung der Äquivalenz aus [GS85, Chapter IX, Definition 1.2.]. Zunächst wird dort auf die Parametrisierung mit den Entfaltungsparametern α verzichtet und die Verzweigungsparameter sind dort nur eindimensional.

Außerdem verzichtet Definition 3.5 auf den Erhalt der Orientierung bei der Transformation von ξ und λ. Während diese aus algebraischer Sicht, insbesondere bei der Betrachtung von Eigenwerten, entscheidend ist, spielt sie bei der Konstruktion numerischer Verfahren keine Rolle.

Satz 3.7. *Die Kontaktäquivalenz ist eine Äquivalenzrelation.*

Beweis. Zu zeigen ist die Reflexivität, Transitivität und Symmetrie der Kontaktäquivalenz.

(i) Mit der Wahl $S(\xi,\tau,\alpha) \equiv \mathrm{Id}$, $X(\xi,\tau,\alpha) \equiv \xi$, $\Lambda(\tau,\alpha) \equiv \tau$ ergibt sich

$$h(\xi,\tau,\alpha) \equiv S(\xi,\tau,\alpha)h\Big(X(\xi,\tau,\alpha),\Lambda(\tau,\alpha),\alpha\Big).$$

Alle Bedingungen an S, X, Λ sind offenbar erfüllt, also ist die Kontaktäquivalenz reflexiv.

(ii) Ist $h_1 \sim h_2$ und $h_2 \sim h_3$ mittels S_1, X_1, Λ_1 respektive S_2, X_2, Λ_2, so folgt

$$\begin{aligned} h_1(\xi,\tau,\alpha) &\equiv S_1(\xi,\tau,\alpha)h_2\Big(X_1(\xi,\tau,\alpha),\Lambda_1(\tau,\alpha),\alpha\Big) \\ &\equiv S_1(\xi,\tau,\alpha)S_2(X_1(\xi,\tau,\alpha),\Lambda_1(\tau,\alpha),\alpha)\cdot \\ &\qquad \cdot h_3\Big(X_2(X_1(\xi,\tau,\alpha),\Lambda_1(\tau,\alpha),\alpha),\Lambda_2(\Lambda_1(\tau,\alpha),\alpha),\alpha\Big), \end{aligned}$$

also ist $h_1 \sim h_3$ vermöge

$$\begin{aligned} S(\xi,\tau,\alpha) &\equiv S_1(\xi,\tau,\alpha)S_2(X_1(\xi,\tau,\alpha),\Lambda_1(\tau,\alpha),\alpha), \\ X(\xi,\tau,\alpha) &\equiv X_2(X_1(\xi,\tau,\alpha),\Lambda_1(\tau,\alpha),\alpha), \\ \Lambda(\tau,\alpha) &\equiv \Lambda_2(\Lambda_1(\tau,\alpha),\alpha), \end{aligned}$$

denn

$$\begin{aligned}
S(\xi_1^*,\tau^*,\alpha^*) &= S_1(\xi_1^*,\tau^*,\alpha^*)S_2(X_1(\xi_1^*,\tau^*,\alpha^*),\Lambda_1(\tau^*,\alpha^*),\alpha^*)\\
&= S_1(\xi_1^*,\tau^*,\alpha^*)S_2(\xi_2^*,\tau^*,\alpha^*) \text{ nichtsingulär},\\
\Lambda(\tau^*,\alpha^*) &= \Lambda_2(\Lambda_1(\tau^*,\alpha^*),\alpha^*) = \Lambda_2(\tau^*,\alpha^*) = \tau^*,\\
\Lambda_\tau(\tau^*,\alpha^*) &= \partial_\tau\Lambda_2(\Lambda_1(\tau^*,\alpha^*),\alpha^*)\partial_\tau\Lambda_1(\tau^*,\alpha^*)\\
&= \partial_\tau\Lambda_2(\tau^*,\alpha^*)\partial_\tau\Lambda_1(\tau^*,\alpha^*) \text{ nichtsingulär},\\
X(\xi_1^*,\tau^*,\alpha^*) &= X_2(X_1(\xi_1^*,\tau^*,\alpha^*),\Lambda_1(\tau^*,\alpha^*),\alpha^*)\\
&= X_2(\xi_2^*,\tau^*,\alpha^*)\\
&= \xi_3^*,\\
X_\xi(\xi_1^*,\tau^*,\alpha^*) &= \partial_\xi X_2(X_1(\xi_1^*,\tau^*,\alpha^*),\Lambda_1(\tau^*,\alpha^*),\alpha^*)\partial_\xi X_1(\xi_1^*,\tau^*,\alpha^*)\\
&= \partial_\xi X_2(\xi_2^*,\tau^*,\alpha^*)\partial_\xi X_1(\xi_1^*,\tau^*,\alpha^*) \text{ nichtsingulär},
\end{aligned}$$

und die Kontaktäquivalenz ist transitiv.

(iii) Sei $h_1 \sim h_2$ mittels S, X, Λ. Wegen $S(\xi_1^*,\tau^*,\alpha^*)$ nichtsingulär und $S \in \mathcal{C}^\infty$ ist S in einer Umgebung von $(\xi_1^*,\tau^*,\alpha^*)$ invertierbar mit $S^{-1} \in \mathcal{C}^\infty$.

Da $X(\xi_1^*,\tau^*,\alpha^*) = \xi_2^*$ und $X_\xi(\xi_1^*,\tau^*,\alpha^*)$ nichtsingulär ist existiert nach dem Satz über implizite Funktionen eine $\mathcal{C}^\infty$-Funktion $\sigma = \sigma\Big(\xi_2,\tau,\alpha\Big)$ mit

$$X(\sigma\Big(\xi_2,\tau,\alpha\Big),\tau,\alpha) \equiv \xi_2$$

in einer Umgebung von $(\xi_2^*,\tau^*,\alpha^*)$. Ebenso gibt es eine $\mathcal{C}^\infty$-Funktion $K = K(\tau,\alpha)$ mit

$$\Lambda(K(\tau,\alpha),\alpha) \equiv \tau$$

in einer Umgebung von (τ^*,α^*). Damit ergibt sich

$$\begin{aligned}
& h_2\Big(X(\xi,\tau,\alpha),\Lambda(\tau,\alpha),\alpha\Big) \equiv S^{-1}(\xi,\tau,\alpha)h_1(\xi,\tau,\alpha)\\
\iff\quad & h_2\Big(X(\xi,\tau,\alpha),\tau,\alpha\Big) \equiv S^{-1}(\xi,K(\tau,\alpha),\alpha)h_1(\xi,K(\tau,\alpha),\alpha)\\
\iff\quad & h_2(\xi,\tau,\alpha) \equiv S^{-1}(\sigma\Big(\xi,K(\tau,\alpha),\alpha\Big),K(\tau,\alpha),\alpha)\cdot\\
& \qquad\qquad \cdot h_1(\sigma\Big(\xi,K(\tau,\alpha),\alpha\Big),K(\tau,\alpha),\alpha)
\end{aligned}$$

in einer Umgebung von $(\xi_2^*, \tau^*, \alpha^*)$. Dabei gelten:

$$\begin{aligned} K(\tau^*, \alpha^*) &= \tau^*, \\ K_\tau(\tau^*, \alpha^*) &= (\Lambda_\tau(\tau^*, \alpha^*))^{-1} \text{ nichtsingulär}, \\ \sigma\Big(\xi_2^*, K(\tau^*, \alpha^*), \alpha^*\Big) &= \sigma(\xi_2^*, \tau^*, \alpha^*) = \xi_1^*, \\ \sigma_\xi(\xi_2^*, K(\tau^*, \alpha^*), \alpha^*) &= \sigma_\xi(\xi_2^*, \tau^*, \alpha^*) = \\ &= (X_\xi(\xi_1^*, \tau^*, \alpha^*))^{-1} \text{ nichtsingulär}, \end{aligned}$$

und

$$\begin{aligned} S^{-1}(\sigma\Big(\xi_2^*, K(\tau^*, \alpha^*), \alpha^*\Big), K(\tau^*, \alpha^*), \alpha^*) &= S^{-1}(\sigma(\xi_2^*, \tau^*, \alpha^*), \tau^*, \alpha^*) \\ &= S^{-1}(\xi_1^*, \tau^*, \alpha^*) \text{ nichtsingulär}. \end{aligned}$$

Also ist die Kontaktäquivalenz symmetrisch. □

Satz 3.8. *Seien h_1, h_2 reduzierte Funktionen von F. Dann sind h_1 und h_2 kontakt-äquivalent.*

Beweis. Seien $\tilde{T}_1, \tilde{T}_2 \in \mathbb{R}^{(n+l)\times(k+l)}$ und $\tilde{Z}_1, \tilde{Z}_2 \in \mathbb{R}^{n\times k}$ die Matrizen, die zur Definition von h_1, h_2 führten. Zu bestimmen sind S, X und Λ so, dass

$$h_1(\xi, \tau, \alpha) \equiv S(\xi, \tau, \alpha) h_2(X(\xi, \tau, \alpha), \Lambda(\tau, \alpha), \alpha).$$

Deshalb muss in einer Umgebung von $(\xi_1^*, \tau^*, \alpha^*)$ gelten

$$\begin{aligned} &\left(Z^\intercal \tilde{Z}_1\right)^{-1} Z^\intercal F(x_1(\xi, \tau, \alpha), \tau, \alpha) \equiv \\ &\qquad \equiv S(\xi, \tau, \alpha) \left(Z^\intercal \tilde{Z}_2\right)^{-1} Z^\intercal F(x_2(X(\xi, \tau, \alpha), \Lambda(\tau, \alpha), \alpha), \Lambda(\tau, \alpha), \alpha). \end{aligned}$$

Wählt man zunächst

$$\begin{aligned} S(\xi, \tau, \alpha) &\equiv \left(Z^\intercal \tilde{Z}_1\right)^{-1} (Z^\intercal \tilde{Z}_2), \\ \Lambda(\tau, \alpha) &\equiv \tau, \end{aligned}$$

und fordert

$$x_1(\xi, \tau, \alpha) \equiv x_2(X(\xi, \tau, \alpha), \tau, \alpha),$$

so ergibt sich nach Multiplikation mit $\tilde{T}_2^\intercal$ von links wegen (3.8b)

$$X(\xi, \tau, \alpha) \equiv \tilde{T}_2^\intercal x_1(\xi, \tau, \alpha).$$

Im singulären Punkt gelten dann

$$\begin{aligned}
S(\xi_1^*, \tau^*, \alpha^*) &= \left(Z^\intercal \tilde{Z}_1\right)^{-1} (Z^\intercal \tilde{Z}_2) \text{ nichtsingulär,} \\
\Lambda(\tau^*, \alpha^*) &= \tau^*, \\
\Lambda_\tau(\tau^*, \alpha^*) &= \mathrm{Id} \text{ nichtsingulär,} \\
X(\xi_1^*, \tau^*, \alpha^*) &= \tilde{T}_2^\intercal x_1(\xi_1^*, \tau^*, \alpha^*) = \tilde{T}_2^\intercal x^* = \xi_2^*, \\
X_\xi(\xi_1^*, \tau^*, \alpha^*) &= \tilde{T}_2^\intercal \partial_\xi x_1(\xi_1^*, \tau^*, \alpha^*) \\
&\overset{(3.11)}{=} \tilde{T}_2^\intercal T \left(\tilde{T}_1^\intercal T\right)^{-1} \text{ nichtsingulär.}
\end{aligned}$$

Damit ist alles bewiesen. □

3.3 Klassifikation

Im nächsten Schritt werden die singulären Punkte in ihre Äquivalenzklassen bezüglich der Kontaktäquivalenz, genannt Singularitäten, eingeteilt und sogenannte Bestimmungsgleichungen angegeben, die die jeweilige Singularität als reguläre Lösung besitzen. Dazu werden als Normalform bekannte Funktionen als ausgewählte Repräsentanten ihrer jeweiligen Klasse identifiziert. Die Einteilung erfolgt anhand der Kodimension q der Klasse, welche ein Maß für die Degeneration der Klasse darstellt. Die Bestimmungsgleichungen werden das Problem darstellen, das es numerisch zu lösen gilt, um singuläre Punkte zu bestimmen.

Es hat sich gezeigt, dass ein singulärer Punkt zumindest die Gleichungen

$$\begin{aligned} h &= 0 \\ h_\xi &= 0 \end{aligned}$$

erfüllen muss. Da dieses Gleichungssystem durch Reduktion der Ausgangsgleichung $F = 0$ entstanden ist, heißt es *reduziertes System* zum singulären Punkt. Für Singularitäten mit höherem Rangdefekt k ist dieses System nicht notwendigerweise regulär oder überhaupt quadratisch.

Definition 3.9. *Ein* erweitertes System *des reduzierten Systems zum singulären Punkt* (x^*, τ^*, α^*) *ist gegeben durch*

$$\begin{aligned} h(\xi, \tau, \alpha) &= 0, \\ h_\xi(\xi, \tau, \alpha) &= 0, \\ b(\xi, \tau, \alpha) &= 0, \end{aligned} \tag{3.12}$$

wobei $b : \mathbb{R}^{k+l} \times \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}^m$ *so gewählt sei, dass die Jacobi-Matrix*

$$\begin{pmatrix} h_\xi & h_\tau \\ h_{\xi\xi} & h_{\xi\tau} \\ b_\xi & b_\tau \end{pmatrix} \tag{3.13}$$

im singulären Punkt vollen Spaltenrang hat.

Ist dabei m *minimal und das System quadratisch, also* $m = l+p+q-k(k+l)$*, mit nichtsingulärer Jacobi-Matrix im singulären Punkt, so heißt* (3.12) minimal erweitertes System *und die dabei benötigte Anzahl* q *der Entfaltungsparameter* α *ist die* Kodimension *des singulären Punktes. Die Gleichungen* (3.12) *heißen* Bestimmungsgleichungen.

Bemerkung 3.10.

(i) Dies ist nicht die formale algebraische Definition der Kodimension, vergleiche dafür [GS85, Chapter III, §§1,2].

(ii) Eine Entfaltung $H(\xi, \tau, \alpha)$ *einer Funktion* $h(\xi, \tau)$ *ist eine mehrparametrige Störung von* h *mit*

$$H(\xi, \tau, 0) \equiv h(\xi, \tau).$$

Es zeichnen sich sogenannte universelle Entfaltungen *aus, für die jede genügend kleine Störung von* h *kontakt-äquivalent ist zu* $H(\cdot, \cdot, \alpha)$ *für ein* α *nahe bei* 0*, und dabei mit einer minimalen Anzahl von Entfaltungsparametern auskommen, siehe [GS85, Def. 1.1, Def. 1.3., Chapter III, §1] für Details.*

Ein handlicher Weg, eine Entfaltung zu bestimmen, ist das spaltenreguläre System (3.13) *mit Einheitsvektoren aufzufüllen, so dass die Jacobi-Matrix der Bestimmungsgleichungen im singulären Punkt invertierbar ist. Üblicherweise sind die zusätzlichen Gleichungen* b *von der Form*

$$\frac{\partial^{|\mathbf{i}|} h}{\partial \xi_1^{\mathbf{i}_1} \dots \partial \xi_{k+l}^{\mathbf{i}_{k+l}} \partial \tau_1^{\mathbf{i}_{k+l+1}} \dots \partial \tau_p^{\mathbf{i}_{p+k+l}}} = 0, \ \mathbf{i} \in \mathbb{N}_0^{p+k+l} \text{ Multiindex},$$

so dass sich eine 1 *in der zu dieser Gleichung gehörenden Zeile in einem Summanden der Form*

$$\alpha_i \xi_1^{\mathbf{i}_1} \dots \xi_{k+l}^{\mathbf{i}_{k+l}} \tau_1^{\mathbf{i}_{k+l+1}} \dots \tau_p^{\mathbf{i}_{p+k+l}},$$

in der reduzierten Funktion h *niederschlägt.*

Diese Technik liefert nicht notwendigerweise eine universelle Entfaltung*. Bei der numerischen Bestimmung der singulären Punktes genügt allerdings eine nichtsinguläre Jacobimatrix. In [Kun90] werden beispielsweise nicht-entfaltete Singularitäten untersucht, bei deren Berechnung allerdings zufällige Richtungen gewählt werden müssen, um die benötigten nicht-singulären Bestimmungssysteme zu erhalten. Der hier vorgestellte Ansatz liefert einen generischen Weg, einer Singularität eine brauchbare Entfaltung zuzuordnen.*

Beispiel 3.11. *Die Mistgabelverzweigung (engl.* pitchfork*) ist eine Verzweigung, bei der die Anzahl der Lösungen von einer auf drei springt, wenn der Verzweigungsparameter* τ *einen gewissen Wert* τ_0 *überschreitet. Sie wird gegeben durch die Modellgleichung*

$$h = \xi^3 + \tau\xi = 0$$

mit den Bestimmungsgleichungen

$$h = h_\xi = h_{\xi\xi} = h_\tau = 0.$$

Abbildung 3.1 zeigt das Nullstellengebilde von h. Wir haben $n = 1$, $l = 0$, $p =$

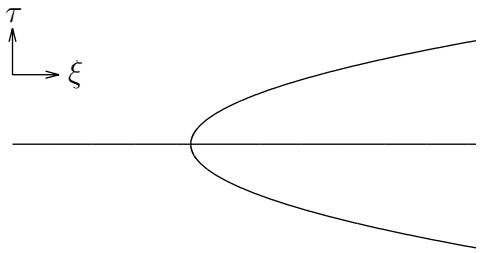

Abbildung 3.1: Nullstellengebilde der Funktion $h = \xi^3 + \tau\xi$

1 und $k = 1$, also zwei Variablen und vier skalare Gleichungen. Damit ist die Kodimension $q = 2$ und wir bestimmen die Entfaltung durch Betrachten der Jacobi-Matrix

$$\begin{pmatrix} h_\xi & h_\tau & * & * \\ h_{\xi\xi} & h_{\xi\tau} & * & * \\ h_{\xi\xi\xi} & h_{\xi\xi\tau} & * & * \\ h_{\tau\xi} & h_{\tau\tau} & * & * \end{pmatrix} = \begin{pmatrix} 3\xi^2 + \tau & \xi & * & * \\ 6\xi & 1 & * & * \\ 6 & 0 & * & * \\ 1 & 0 & * & * \end{pmatrix}$$

im singulären Punkt $\xi^* = 0, \tau^* = 0$. Damit das System regulär wird, müssen die beiden freien Spalten von Null verschiedene Einträge in der ersten und in der dritten oder vierten Zeile haben, beispielsweise

$$\begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 6 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

Dies entspricht gerade der Entfaltung

$$\tilde{h}_1(\xi, \tau, \alpha) = \xi^3 + \tau\xi + \alpha_1 + \frac{1}{2}\alpha_2\xi^2.$$

Die andere Wahl ergibt

$$\tilde{h}_2(\xi, \tau, \alpha) = \xi^3 + \tau\xi + \alpha_1 + \alpha_2\tau$$

und beide sind, abgesehen von dem konstanten Faktor $\frac{1}{2}$, gerade die in [GS85, Chapter III, §3 (a)] angegebenen möglichen universellen Entfaltungen der Mistgabelverzweigung.

Den Grund für die Verwendung von entfalteten Singularitäten liefert [JS85, Theorem 3.9]: Universell entfaltete Singularitäten sind stabil in dem Sinne, dass für jede glatte Störung p und hinreichend kleines $\varepsilon > 0$ die gestörte Funktion

$$\hat{h}(\xi, \tau, \alpha) = h(\xi, \tau, \alpha) + \varepsilon p(\xi, \tau, \alpha)$$

in $(\xi^*, \tau^*, \alpha^*)$ die gleiche Singularität aufweist.

3.3.1 Skalare Singularitäten

Es folgen die Normalformen und Bestimmungsgleichungen der skalaren Singularitäten bis zur Kodimension drei, das heißt für (3.1) mit $k = 1,\ l = 0,\ p = 1$ und $q \leq 3$. Die Dimension n der Zustandsvariablen ist an dieser Stelle durch die Verwendung der Ljapunow-Schmidt-Reduktion nicht entscheidend.

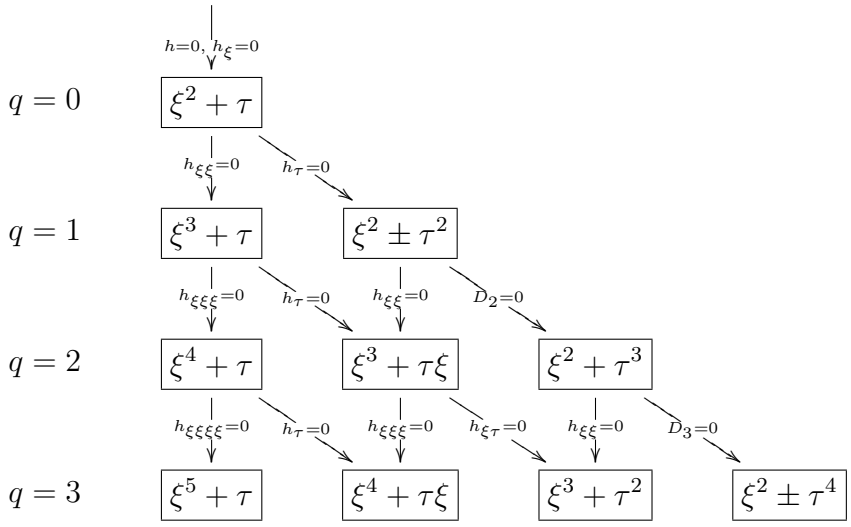

Abbildung 3.2: Hierachie der Singularitäten für $k = 1,\ l = 0,\ p = 1$ bis zur Kodimension $q = 3$.
Dabei gilt: $D_2 = h_{\tau\xi}^2 - h_{\xi\xi}h_{\tau\tau},\ D_{k+1} = h_{\xi\xi}\partial_\tau(D_k) - h_{\xi\tau}\partial_\xi(D_k)$.

Abbildung 3.2 zeigt die Äquivalenzklassen bis zur Kodimension $q = 3$ und ihre Bestimmungsgleichungen, siehe auch [JS85]. Die bestimmenden Gleichungen erhält man, in dem man ausgehend vom Ursprung der Hierarchie alle Gleichungen an den durchlaufenden Pfeilen einsammelt.

Die erste Spalte läßt sich zur Familie der verallgemeinerten Rückkehrpunkte $\xi^k + \tau,\ k \geq 2$, erweitern. Die zweite Spalte bildet analog ab Kodimension 2 zur Familie der verallgemeinerten Mistgabelverzweigungen $\xi^k + \xi\tau,\ k \geq 3$. Ebenso bildet die Diagonale eine Familie, $\xi^2 + \varepsilon\tau^k,\ k \geq 2$, mit $\varepsilon = 1$ für k ungerade, $\varepsilon = \pm 1$ für k gerade.

Die Singularität mit der Normalform $\xi^3 + \tau^2$, der sogenannte geflügelte Spitzpunkt, nimmt eine Sonderstellung ein. Nach [Gov97] bildet er den Ausgangspunkt für alle noch nicht klassifizierten, skalaren Singularitäten.

3.3.2 Höher-dimensionale Singularitäten

Im Bereich der höher-dimensionalen Singularitäten ist die Theorie deutlich überschaubarer. Dies ist vor allem der Tatsache geschuldet, dass die Kodimension nach unten durch $n^2 - 1$ beschränkt ist, siehe [GS85, Chapter IX, Prop. 1.3], die Singularitäten bei steigender Anzahl der Zustandsvariablen also deutlich an Komplexität zunehmen. Insbesondere haben Singularitäten mit drei oder mehr Zustandsvariablen mindestens die Kodimension 8.

Mit den Normalformen

$$\begin{pmatrix} \xi_1^2 - \xi_2^2 + \tau \\ 2\xi_1\xi_2 \end{pmatrix} \tag{3.14a}$$

$$\begin{pmatrix} \xi_1^2 + \tau \\ \xi_2^2 + \tau \end{pmatrix} \tag{3.14b}$$

ist die Klassifikation bis zur Kodimension 3 vollständig, siehe [GS85, Chapter IX, Theorem 2.1].

Beide Singularitäten haben die einfachste Form der Bestimmungsgleichungen

$$h = 0,\ h_\xi = 0$$

und sind nur durch Nicht-Degeneriertheitsbedingungen zu unterscheiden.

In Kapitel 5 werden wir noch drei andere Singularitäten mit Rangdefekt $k = 2$ und $k = 3$ besprechen.

3.3.3 Moduli

Die Obergrenze $q = 3$ für die Klassifikation ist zwar willkürlich, hat aber zwei Hintergründe. Zunächst wird die Behandlung von Singularitäten mit höherer Kodimension durch das Auftauchen von sogenannten *Moduli-Termen*

erschwert, siehe [GS79, p. 57] oder [GS85, Chapter V]. Dabei handelt es sich um Entfaltungsparameter, bei denen auch für von 0 verschiedene Werte von α Singularitäten der gleichen Kodimension auftreten. Im Wesentlichen erhält man dadurch eine ganze Familie von nicht-äquivalenten Singularitäten, siehe auch Beispiel 3.12. Dadurch geht auch die binäre Struktur der Hierarchie verloren. Ferner ist das Auftreten von Singularitäten mit niedriger Kodimension bei der Behandlung praxisrelevanter mathematischer Modelle wahrscheinlicher, vergleiche [GS85, Chapter IV, §0,§1].

Beispiel 3.12 (Moduli)**.** *(vergleiche [GS85, Chapter V, §1])*
Betrachtet werde die Gleichung

$$h_m(\xi, \tau) = \xi(\xi + \tau)(\xi - m\tau)$$

mit $k = 1,\ l = 0$ *und* $p = 1$ *und* $m \in \mathbb{R}$, $m > 0$. *Das Nullstellengebilde von* h *besteht aus drei Geraden, die sich im Ursprung schneiden:*

$$\Big\{(\xi, \tau) : h(\xi, \tau) = 0\Big\} = \Big\{\xi = 0\Big\} \cup \Big\{\xi = -\tau\Big\} \cup \Big\{\xi = m\tau\Big\}.$$

Die Abhängigkeit der Nullstellenmenge von m *ist von einem qualitativen Standpunkt recht gering, die Transformation*

$$X(\xi, \tau) = \begin{cases} \frac{\xi}{m} & \textit{falls } \xi\tau \geq 0 \\ \xi & \textit{falls } \xi\tau \leq 0 \end{cases}, \quad \Lambda(\tau) = \tau, \quad S \equiv \mathrm{Id}$$

bildet das Nullstellengebilde für beliebiges $m > 0$ *auf das für* $m = 1$ *ab. Diese Transformation erfüllt jedoch nicht die Forderungen an die Kontaktäquivalenz und zwei Funktionen* h_m *und* h_n *mit* $m, n > 0$ *genau dann kontaktäquivalent, wenn* $n = m$, *vergleiche [GS85, Chapter V, Lemma 1.2].*

Für unsere Zwecke spielt dieser Sachverhalt kaum eine Rolle. Die Bestimmungsgleichungen gelten für die ganze Familie, so dass wir nur den Moduli-Parameter festhalten müssen um einen singulären Punkt einer solchen Singulärität zu bestimmen. In Kapitel 5 werden wir den *Tripel Punkt* als ein Beispiel aus einer Moduli-Familie betrachten.

Kapitel 4

Numerisches Verfahren

Dieses Kapitel behandelt besondere Aspekte, die bei der Anwendung der bisher beschriebenen Techniken zur Lösung des folgenden Problems auftreten.

Problem 4.1. *Sei F wie in* (3.1) *gegeben und es gelte* (3.2). *Gegeben sei eine Startschätzung (x^0, τ^0, α^0) und der Typ der Singularität, also $k, q \in \mathbb{N}$, die Bestimmungsgleichungen und eine Entfaltung.*
Finde den singulären Punkt (x^*, τ^*, α^*).

Wir werden dieses Problem mit einem zweistufigen Newton-Verfahren lösen. Dabei berechnet die innere Iteration die Ljapunow-Schmidt-Reduzierte, im äußeren Verfahren werden die Bestimmungsgleichungen der Singularität gelöst.

4.1 Wahl der Ränderungen

Die erste Aufgabe eines numerischen Verfahrens zur Lösung des Problems 4.1 muss die Wahl der Ränderungen $\tilde{T}$ und $\tilde{Z}$ zur Definition des geränderten Systems (3.4) sein, so dass (3.6a) erfüllt ist. Dabei gibt es zwei grundlegende Möglichkeiten:

Wegen Satz 3.1 genügt es (3.6b) zu fordern. Man kann also versuchen, mit zufällig gewählten Ränderungen $\tilde{T}$ und $\tilde{Z}$ zu arbeiten. Die Produktmatrizen $T^\intercal\tilde{T}$ und $Z^\intercal\tilde{Z}$ sind dann fast sicher nichtsingulär, da es unwahrscheinlich ist, dass zufällig gewählte Vektoren in den Kern einer Matrix fallen, wenn dieser nicht der ganze Raum ist (echte Unterräume von $\mathbb{R}^n$ sind Lebesgue-Nullmengen).

Bei numerischen Tests zeigte sich dieser Ansatz jedoch nicht so zuverlässig. Eine andere Möglichkeit ist $\tilde{T}$ und $\tilde{Z}$ als tatsächliche Approximation an die orthogonalen Komplemente von Kern bzw. Kobild wählen. Dazu sei $L_0 = \partial_x F(x^0, \tau^0, \alpha^0)$ die Jacobi-Matrix von F an der Ausgangsschätzung. Mit Hilfe einer QR-Zerlegung mit Spaltentausch, bei der außerdem $\operatorname{rang} L_0 = n - k$ gesetzt wird, bestimmt man Kern T_0 und Kobild Z_0 von L_0 und wählt $\tilde{T}$ und $\tilde{Z}$ gemäß

$$\begin{aligned} T_0^\intercal \tilde{T} &= \mathrm{Id}, \\ Z_0^\intercal \tilde{Z} &= \mathrm{Id} \end{aligned} \tag{4.1}$$

als orthogonale Komplemente zu T_0 und Z_0.

Numerische Tests zeigen, dass die Berechnung von $\tilde{T}$ und $\tilde{Z}$ die Anzahl der benötigten Iterationen oft reduziert, wie in den Abbildungen 4.1 und 4.2 zu sehen ist. Dabei wurde bei der zufälligen Wahl der Mittelwert aus zehn erfolgreichen Versuchen verwendet. Konnte keine Konvergenz erzielt werden, wurden die Startwerte wiederhergestellt. Nach zwanzig fehlgeschlagenen Versuchen wurde das Beispiel abgebrochen. Besonders bei den nicht-trivialen Beispielen zeigt sich die Instabilität der zufälligen Wahl der Ränderungen..

Die Approximation kann auch im Laufe der Iteration verbessert werden, indem der Prozess für neue Iterierte (x^k, τ^k, α^k) wiederholt wird. Dies ist allerdings riskant, da dabei die vom äusseren Verfahren verwendete Funktion h verändert wird. Es besteht kein Grund zur Annahme, dass so ein stabiles Verfahren entsteht. Numerische Tests bestätigten diesen /Sachverhalt. Während die Anzahl der Iterationen zur Ljapunow-Schmidt-Reduktion bei einfachen Beispielen abnahm, konnte bei den komplizierteren Problemen überhaupt keine Konvergenz im äußeren Verfahren erzielt werden, wenn die Ränderungen am Anfang jeder Ljapunow-Schmidt-Reduktion neu berechnet wurden.

Für eine ausführlichere Diskussion dieses Themas, insbesondere zur Wahl konditionsoptimierender Ränderungen, siehe [Sch00, Kapitel 3].

4.2 Anheben der Entfaltung

Es ist nicht unüblich, dass im Ausgangsproblem keine freien Parameter auftauchen, die als Entfaltungsparameter ausgezeichnet werden können. Selbst wenn freie Parameter vorhanden sind, kann die Auswahl geeigneter Entfaltungsparameter ein sehr aufwändiger Vorgang sein. Beispielsweise beim

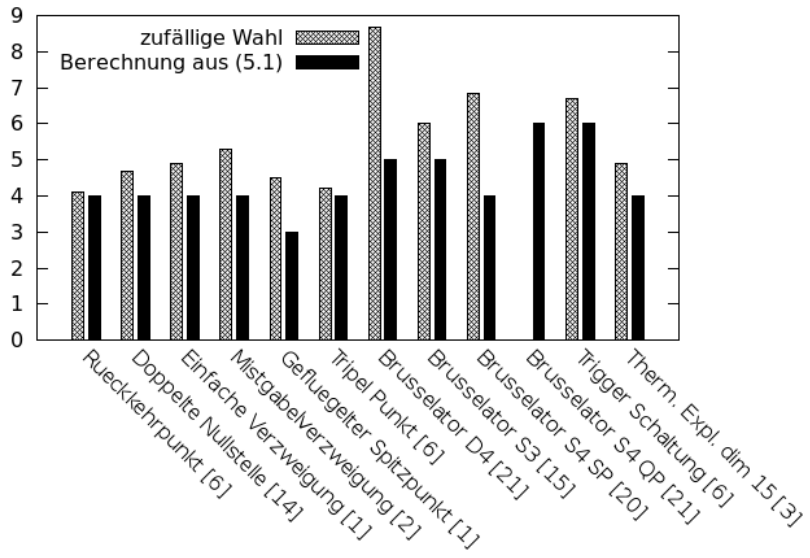

Abbildung 4.1: Anzahl der Iterationen im äusseren Newton-Verfahren für die unterschiedlichen Ansätze zur Wahl von $\tilde{T}$ und $\tilde{Z}$. Die Werte in den Klammern geben die Anzahl der fehlgeschlagenen Versuche bei der zufälligen Wahl an.

Brusselator, der in Abschnit 5.2 vorgestellt wird, werden je nach betrachteter Symmetrie bis zu elf Entfaltungsparameter benötigt. Da Entfaltungen außdem auf der reduzierten Funktion definiert werden, besteht kein zwingender Grund, die benötigten Parameter tatsächlich in der Ausgangsfunktion auszuzeichnen.

Es gibt nun drei mögliche Vorgehensweisen, die Entfaltungsparameter ins Spiel zu bringen:

(i) Die Entfaltung ist bereits in der Ausgangsfunktion gegeben.

(ii) Die Entfaltung wird erst nach der Ljapunow-Schmidt-Reduktion direkt auf die reduzierte Funktion angewendet.

(iii) Die auf der reduzierten Funktion definierte Entfaltung wird auf das Ausgangsproblem angehoben.

Die ersten beiden Varianten sind selbsterklärend, die Dritte wird nun näher beschrieben.

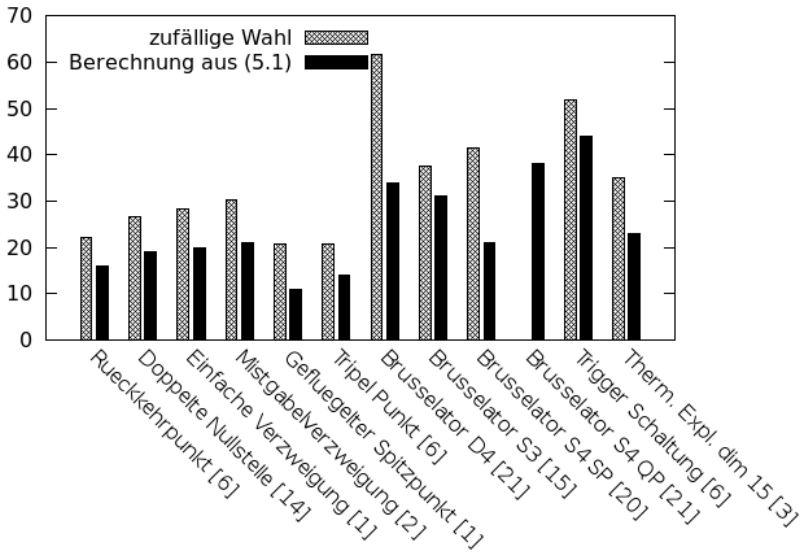

Abbildung 4.2: Gesamtzahl der Iterationen zur Ljapunow-Schmidt-Reduktion in Abbildung 4.1.

Es sei an die Definition der reduzierten Funktion erinnert, wonach h durch die Gleichung

$$h(\xi, \tau, \alpha) = \left(Z^\intercal \tilde{Z}\right)^{-1} Z^\intercal F(x(\xi, \tau, \alpha), \tau, \alpha)$$

gegeben wird. Außerdem ist nach (3.8b)

$$\tilde{T}^\intercal x(\xi, \tau, \alpha) = \xi.$$

Ist dann

$$\alpha_i \prod_{j=1}^{J_1} \xi_{r_j} \prod_{j=1}^{J_2} \tau_{s_j} \tag{4.2}$$

mit $1 \leq r_j \leq k + l$ und $1 \leq s_j \leq p$ für alle j ein Summand der Entfaltung in der t-ten Komponente der reduzierten Funktion, können wir diesen mit

$$\tilde{F}(x, \tau, \alpha) = F(x, \tau, \alpha) + \tilde{Z} e_t \alpha_i \prod_{j=1}^{J_1} \left(e_{r_j} \tilde{T}^\intercal x\right) \prod_{j=1}^{J_2} \tau_{s_j}$$

auf die Ausgangsfunktion liften. Denn setzt man dies in die Definition der

reduzierten Funktion ein, ergibt sich unter Beachtung der Beziehung (3.8b)

$$\begin{aligned}
\tilde{h}(\xi,\tau,\alpha) &= \left(Z^\intercal \tilde{Z}\right)^{-1} Z^\intercal \tilde{F}(x(\xi,\tau,\alpha),\tau,\alpha) \\
&= \left(Z^\intercal \tilde{Z}\right)^{-1} Z^\intercal F(x(\xi,\tau,\alpha),\tau,\alpha) + \\
&\quad + \left(Z^\intercal \tilde{Z}\right)^{-1} Z^\intercal \tilde{Z} e_t \alpha_i \prod_{j=1}^{J_1} \left(e_{r_j} \tilde{T}^\intercal x(\xi,\tau,\alpha)\right) \prod_{j=1}^{J_2} \tau_{s_j} \\
&= h(\xi,\tau,\alpha) + e_t \alpha_i \prod_{j=1}^{J_1} \left(e_{r_j} \tilde{T}^\intercal x(\xi,\tau,\alpha)\right) \prod_{j=1}^{J_2} \tau_{s_j} \\
&= h(\xi,\tau,\alpha) + e_t \alpha_i \prod_{j=1}^{J_1} \xi_{r_j} \prod_{j=1}^{J_2} \tau_{s_j},
\end{aligned}$$

also gerade der betrachtete Term (4.2).

4.3 Algorithmus

Folgender Algorithmus berechnet zu gegebenem (ξ,τ,α) aus einer hinreichend kleinen Umgebung von (ξ^*,τ^*,α^*), hinreichend guten Startschätzungen h_0, x_0 und einer vorgegebenen Genauigkeit `eps` die Werte der Funktionen h und x mit (3.7) und ihrer Ableitungen bis zur Ordnung o. Dabei sind $\begin{pmatrix} x_i \\ h_i \end{pmatrix}$ und $\begin{pmatrix} \Delta x_i \\ \Delta h_i \end{pmatrix}$ zusammengesetzte Variablen, die auch Ableitungen nach ξ,τ,α enthalten.

Algorithmus 4.2 (Ljapunow-Schmidt-Reduktion)**.**

1. *Wähle $\tilde{T}$ und $\tilde{Z}$ nach einer der Methoden aus Abschnitt 4.1 und setze $k = 0$.*
2. *Berechne $F(x_k,\tau,\alpha)$ und $\partial_x F(x_k,\tau,\alpha)$ mit* AD*. Berechne dabei auch $\partial^i_{\xi,\tau,\alpha} F(x_k,\tau,\alpha)$, $i = 1,\ldots,o$, mit* AD*.*
3. *Berechne $H(x_k,h_k,\xi,\tau,\alpha)$.*
4. *Setze $\partial_{x,h} H = \begin{pmatrix} \partial_x F(x_k,\tau,\alpha) & -\tilde{Z} \\ \tilde{T}^\intercal & 0 \end{pmatrix}$*
5. *Löse $\partial_{x,h} H(x_k,h_k,\xi,\tau,\alpha) \begin{pmatrix} \Delta x \\ \Delta h \end{pmatrix} = H(x_k,h_k,\xi,\tau,\alpha)$*

6. *Setze* $x_{k+1} = x_k - \Delta x_k$, $h_{k+1} = h_k - \Delta h_k$.

7. *Falls* $\left\| \begin{pmatrix} \Delta x_k \\ \Delta h_k \end{pmatrix} \right\| < \texttt{eps}$, *akzeptiere* $\begin{pmatrix} x_{k+1} \\ h_{k+1} \end{pmatrix}$ *als Approximation an* $\begin{pmatrix} x(\xi, \tau, \alpha) \\ h(\xi, \tau, \alpha) \end{pmatrix}$. *Andernfalls setze* $k = k + 1$ *und gehe zu 2.*

Bei der Berechnung von H in Schritt 3 werden durch die Verwendung der zusammengesetzen Variablen $\begin{pmatrix} x_i \\ h_i \end{pmatrix}$ auch Ableitungen nach ξ, τ, α berechnet.

Bemerkung 4.3. *Die im Abbruchkriterium gewählte Norm muss die Ableitungen berücksichtigen. Im nachträglich differenzierten Verfahren verzichtet man zunächst auf das Mitrechnen der Ableitungen bis zur Konvergenz und wechselt dann auf eine entsprechende Norm. Tatsächlich wird in beiliegender Referenzimplementation beim nachträglich differenzierten Verfahren im Wesentlichen der gleiche Code für die Pre- und Post-Iteration bei der Ljapunow-Schmidt Reduktion verwendet und es muss nur der Template Parameter für den verwendeten Datentyp angepasst werden. Bei der Post-Iteration wird zusätzlich auf die wiederholte Berechnung der Jakobimatrix verzichtet.*

Schließlich geben wir einen Algorithmus an, um das Problem 4.1 zu lösen. Dabei handelt es sich im Wesentlichen nur noch um ein gewöhnliches Newton-Verfahren, mit dem die Bestimmungsgleichungen für die vorliegende Singularität gelöst werden. Dabei bezeichne $\varrho_k = (\xi_k, \tau_k, \alpha_k)$.

Algorithmus 4.4.

1. *Setze* $\xi_0 = \tilde{T}^\intercal x_0$, $k = 0$.

2. *Berechne* $h(\varrho_k), x(\varrho_k)$ *und ihre Ableitungen bis zur benötigten Ordnung mit Algorithmus 4.2.*

3. *Berechne die Newton-Korrektur zum aktuellen Schritt:*
$$\begin{pmatrix} 0 & h_\tau(\varrho_k) & h_\alpha(\varrho_k) \\ h_{\xi\xi}(\varrho_k) & h_{\xi\tau}(\varrho_k) & h_{\xi\alpha}(\varrho_k) \\ b_\xi(\varrho_k) & h_\tau(\varrho_k) & h_\alpha(\varrho_k) \end{pmatrix} \begin{pmatrix} \Delta\xi_k \\ \Delta\tau_k \\ \Delta\alpha_k \end{pmatrix} = \begin{pmatrix} h(\varrho_k) \\ h_\xi(\varrho_k) \\ b(\varrho_k) \end{pmatrix}$$

4. *Setze* $\varrho_{k+1} = \varrho_k - \Delta\varrho_k$.

5. *Falls* $\|\Delta\varrho_k\| < \texttt{eps}$, *akzeptiere* $\begin{pmatrix} x(\varrho_k) \\ \tau_k \\ \alpha_k \end{pmatrix}$ *als Approximation an* $\begin{pmatrix} x_* \\ \tau_* \\ \alpha_* \end{pmatrix}$. *Andernfalls setze* $k = k + 1$ *und gehe zu 2.*

Tabelle 4.1 zeigt Laufzeiten und die Anzahl der inneren Iterationen beim Bearbeiten *aller* Probleme. Der Programmcode wurde mit dem C++-Compiler der GNU Compiler Collection g++ Version 4.6.3 mit Optimierungslevel 2 (-O2) und deaktivierten Debugmakros (-DNDEBUG) übersetzt.

	Σ Laufzeit	Σ innere Iterationen
Vereinfacht differenziertes Verfahren	`6.2 sec`	`297`
Vereinfacht und nachträglich	`3.5 sec`	`353`
Vollständig	`7.6 sec`	`193`
Vollständig und nachträglich	`4.6 sec`	`335`

Tabelle 4.1: Laufzeiten und Iterationsanzahl bei den verschiedenen Möglichkeiten zur Differentiation eines Newton-Verfahrens. Der Test wurde auf einem Athlon64 3800+ (2.4 GHz) durchgeführt.

Beim nachträglich differenzierten Verfahren werden zwar jeweils mehr Iterationen zur Ljapunow-Schmidt Reduktion benötigt, ein Großteil davon sind allerdings vergleichsweise günstig, so dass insgesamt weniger Rechenleistung nötig ist. Eine leistungsstärkere *AD*-Implementation kann diesen Effekt verringern.

Kapitel 5

Numerische Beispiele

Unter Verwendung des vereinfacht differenzierten Verfahrens, Ränderungen nach (4.1) und gelifteten Entfaltungen sollen nun eine Reihe von Beispielen vorgestellt werden, die mit der Referenzimplementation gelöst wurden. Die Startwerte wurden dabei mit dem Pseudozufallsgenerator der C++-Standardbibliothek mit einer maximalen Abweichung vom singulären Punkt in der Größenordnung $5 \cdot 10^{-1}$ bis 10^{-2} in jeder Komponente zufällig gewählt.

Die Toleranz `eps` wurde sowohl für Algorithmus 4.4, als auch Algorithmus 4.2, auf 10^{-8} gesetzt.

5.1 Skalare Singularitäten

Für jede skalare Singularität mit Normalform $h(\xi, \tau, \alpha)$ betrachten wir hier das Beispiel

$$\begin{pmatrix} h(x_1, \tau, \alpha) \\ x_1 - x_2 \end{pmatrix}.$$

5.1.1 Umkehrpunkt

Mit $k = 1,\ l = 0,\ p = 1,\ q = 0$ ist die Modellfunktion für den Umkehrpunkt gegeben durch

$$h(\xi, \tau) = \xi^2 + \tau \tag{5.1}$$

mit den Bestimmungsgleichungen

$$h = h_\xi = 0.$$

Das System ist bereits quadratisch, die Kodimension $q = 0$ und es werden keine Entfaltungsparameter benötigt. Tabelle 5.1 zeigt den Iterationsverlauf. Dabei ist $u = \begin{pmatrix} x \\ \tau \\ \alpha \end{pmatrix}$, $\mu = \begin{pmatrix} \xi \\ \tau \\ \alpha \end{pmatrix}$ und die Spalte RED gibt die Anzahl der Iterationen für die Ljapunow-Schmidt-Reduktion an.

i	$\|u_i - u_*\|$	$\|\Delta\mu_i\|$	$\|F(u_i)\|$	RED
0	3.312e-02	1.439e-01	2.055e-01	5
1	2.623e-03	1.269e-03	1.143e-02	5
2	6.151e-08	2.554e-08	1.054e-04	4
3	1.726e-17	6.407e-18	2.133e-09	2
4	9.230e-37	-	5.379e-19	-

Tabelle 5.1: Iterationsprotokoll für den Umkehrpunkt

5.1.2 Einfache Verzweigung

Die einfache Verzweigung hat die Normalform

$$h(\xi, \tau) = \xi^2 - \tau^2 + \alpha_1, \tag{5.2}$$

also $k = 1$, $l = 0$ und $p = 1$. Die Bestimmungsgleichungen sind gegeben durch

$$h = h_\xi = h_\tau = 0$$

und wir haben die Kodimension $q = 1$ mit der universellen Entfaltung

$$\tilde{h}(\xi, \tau, \alpha) = \xi^2 - \tau^2 + \alpha_1.$$

Tabelle 5.2 zeigt die Ergebnisse.

i	$\|u_i - u_*\|$	$\|\Delta\mu_i\|$	$\|F(u_i)\|$	RED
0	2.376e-02	6.681e-02	1.963e-01	6
1	1.188e-04	4.548e-03	9.219e-04	5
2	4.228e-07	2.480e-07	2.157e-07	5
3	9.330e-15	5.347e-15	4.766e-15	4
4	4.635e-30	-	2.392e-30	-

Tabelle 5.2: Iterationsprotokoll für die einfache Verzweigung

5.1.3 Pitchfork

Im Anschluss an Beispiel 3.11 betrachten wir das Beispiel

$$F(x,\tau,\alpha) = \begin{pmatrix} x_1^3 + x_1\tau^2 \\ x_1 - x_2 \end{pmatrix}$$

mit der Entfaltung $\tilde{h}(\xi,\tau,\alpha) = \xi^3 + \xi\tau^2 + \alpha_1 + \alpha_2\tau$. Tabelle 5.3 zeigt die quadratische Konvergenz.

i	$\|u_i - u_*\|$	$\|\Delta\mu_i\|$	$\|F(u_i)\|$	RED
0	3.205e-03	3.784e-02	6.059e-02	5
1	9.059e-04	1.362e-03	3.972e-05	5
2	1.841e-10	2.060e-08	2.788e-10	4
3	1.555e-19	5.232e-18	1.225e-19	2
4	3.005e-38	-	1.841e-57	-

Tabelle 5.3: Iterationsprotokoll für die Mistgabelverzweigung

5.1.4 Geflügelter Spitzpunkt

Der Geflügelte Spitzpunkt hat die Normalform

$$h(\xi,\tau) = \xi^3 + \tau^2 \tag{5.3}$$

mit $k = 1,\ l = 0,\ p = 1$. Die Bestimmungsgleichungen sind

$$h = h_\xi = h_\tau = h_{\xi\xi} = h_{\xi\tau} = 0$$

und haben die Jacobimatrix

$$\begin{pmatrix} 3\xi^2 & 2\tau \\ 6\xi & 0 \\ 0 & 2 \\ 6 & 0 \\ 0 & 0 \end{pmatrix} \overset{\xi=\tau=0}{=} \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 2 \\ 6 & 0 \\ 0 & 0 \end{pmatrix}$$

und eine Entfaltung ist nach Bemerkung 3.10(ii) gegeben durch

$$\tilde{h}(\xi,\tau,\alpha) = \xi^3 + \tau^2 + \alpha_1 + \alpha_2\xi + \alpha_3\xi\tau$$

Die Ergebnisse der Iteration sind in Tabelle 5.4 aufgelistet.

i	$\|u_i - u_*\|$	$\|\Delta\mu_i\|$	$\|F(u_i)\|$	RED
0	3.568e-02	1.436e-01	1.756e-01	4
1	1.768e-02	1.840e-02	8.076e-04	4
2	1.352e-12	1.790e-12	5.560e-19	3
3	1.794e-43	-	9.888e-29	-

Tabelle 5.4: Iterationsprotokoll für den geflügelten Spitzpunkt

5.1.5 Einsiedlerpunkt

Diese Singularität hat genau wie die einfache Verzweigung $k = 1,\ l = 0,\ p = 1,\ q = 1,$ die Normalformen unterscheiden sich nur durch ein Vorzeichen:

$$h(\xi, \tau) = \xi^2 + \tau^2 + \alpha_1. \tag{5.4}$$

Auch die Entfaltung stimmt überein und Tabelle 5.5 zeigt die quadratische Konvergenz.

i	$\|u_i - u_*\|$	$\|\Delta\mu_i\|$	$\|F(u_i)\|$	RED
0	5.600e-02	1.232e-01	2.886e-01	6
1	6.452e-04	1.530e-02	4.743e-03	5
2	1.314e-05	8.269e-06	6.947e-06	5
3	2.387e-11	1.518e-11	1.259e-11	4
4	7.769e-23	-	4.092e-23	-

Tabelle 5.5: Iterationsprotokoll für die doppelte Nullstelle

5.1.6 Tripel Punkt

Dieses Beispiel mit $k = 1,\ l = 0,\ p = 1,\ q = 5$ wird auch in [GS85, Chapter V] vorgestellt und hat die Normalform

$$h(\xi, \tau) = \xi^3 - \xi\tau^2 \tag{5.5}$$

mit den Bestimmungsgleichungen

$$h = h_\xi = h_\tau = h_{\xi\xi} = h_{\xi\tau} = h_{\tau\tau} = 0$$

Nach Bemerkung 3.10(ii) ist eine Entfaltung durch

$$\tilde{h}(\xi, \tau, \alpha) = \xi^3 - \xi\tau^2 + \alpha_1 + \alpha_2\xi + \alpha_3\tau + \alpha_4\xi^2$$

gegeben, denn die Jacobimatrix der Bestimmungsgleichungen ist

$$\begin{pmatrix} 3\xi^2-\tau^2 & -2\xi\tau \\ 6\xi & -2\xi \\ -2\tau & -2\xi \\ 6 & 0 \\ 0 & -2 \\ -2 & 0 \end{pmatrix} \overset{\xi=\tau=0}{=} \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 6 & 0 \\ 0 & -2 \\ -2 & 0 \end{pmatrix}$$

und die angebene Entfaltung entspricht gerade den zusätzlichen Spalten e_1, e_2, e_3, e_4.

i	$\|u_i - u_*\|$	$\|\Delta\mu_i\|$	$\|F(u_i)\|$	RED
0	3.822e-02	1.959e-01	3.266e-02	5
1	1.061e-02	2.534e-02	3.530e-04	4
2	7.616e-08	3.331e-07	7.433e-12	3
3	6.200e-25	6.882e-22	5.028e-25	2
4	9.281e-50	-	5.510e-40	-

Tabelle 5.6: Iterationsprotokoll für die dreifache Nullstelle

5.2 Brusselator

Ein Brusselator ist eine chemische Modellreaktion, bei der zwei Substanzen in diffusiv gekoppelten Zellen miteinander reagieren, siehe [PL68]. Durch die Wahl der Kopplung kann jede gewünschte Symmetrie erreicht werden. Dabei zeigen sich eine Reihe von höher-dimensionalen Singularitäten.

Die Gleichgewichtslösungen der Reaktionsgleichungen ergeben sich aus

$$R_1(x_{2i}, x_{2i+1}) + d_1 \sum_{j=1}^{\tilde{n}} \theta_{ij}(x_{2i} - x_{2j}) = 0 \tag{5.6a}$$

$$R_2(x_{2i}, x_{2i+1}) + d_2 \sum_{j=1}^{\tilde{n}} \theta_{ij}(x_{2i+1} - x_{2j+1}) = 0 \tag{5.6b}$$

für $i = 1, \ldots, \tilde{n}$, mit den Diffusionskonstanten $d_1 = \tau = x_{2\tilde{n}+2},\ d_2 = 10\tau = 10x_{2\tilde{n}+1}$ und den Reaktionstermen

$$R_1(x, y) = 2 - 7x + x^2 y, \tag{5.7a}$$

$$R_2(x, y) = 6x - x^2 y. \tag{5.7b}$$

Die Matrix $(\theta_{ij})_{i,j=1,\ldots,\tilde{n}}$ beschreibt die Kopplung der Zellen, also $\theta_{ij} = 1$ falls die i-te und j-te Zelle verbunden sind, $\theta_{ij} = 0$ sonst.

Dabei ist x_{2j} die Menge der ersten Substanz in der j-ten Zelle und x_{2j+1} die Menge der zweiten Substanz in der j-ten Zelle. Wir haben also $n = 2\tilde{n},\ l = 1,\ p = 0$, während der Rangdefekt k und die Kodimension q von der gewählten Kopplung abhängen.

Für $x_{2j} = 2,\ x_{2j+1} = 3,\ j = 1, \ldots, \tilde{n};\ \tau = x_{n+l}$ beliebig, erhält man eine triviale Lösung unabhängig von der gewählten Kopplung θ.

Die Vergleichsdaten für die Beispiele in diesem Abschnitt stammen aus dem in [Kun90] beschriebenem Programm `ALCOND`.

5.2.1 Einfache Verzweigung mit Rangdefekt 2

Die einfache Verzweigung mit Rangdefekt 2 hat die Normalform

$$h(\xi, \tau) = \begin{pmatrix} \xi_1^2 - \xi_2^2 - \xi_1\tau \\ -2\xi_1\xi_2 - \xi_2\tau \end{pmatrix}, \tag{5.8}$$

also $k = 2,\ l = 0,\ p = 1$, und die Bestimmungsgleichungen

$$h = 0,\ h_\xi = 0,\ h_\tau = 0,$$

also benötigen wir fünf Entfaltungsparameter, um ein quadratisches System zu erhalten. Die Entfaltung bestimmen wir aus

$$\begin{pmatrix} h \\ h_{\xi_1} \\ h_{\xi_2} \\ h_\tau \end{pmatrix} = \begin{pmatrix} 2\xi_1 - \tau & -2\xi_2 & -\xi_1 \\ -2\xi_2 & -2\xi_1 - \tau & -\xi_2 \\ 2 & 0 & -1 \\ 0 & -2 & 0 \\ -1 & 0 & 0 \\ 0 & -2 & 0 \\ -2 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

für $\xi_1 = \xi_2 = \tau = 0$ als

$$\tilde{h}(\xi, \tau, \alpha) = \begin{pmatrix} \xi_1^2 - \xi_2^2 - \xi_1\tau + \alpha_1 + \alpha_3\xi_1 + \alpha_5\xi_2 \\ -2\xi_1\xi_2 - \xi_2\tau + \alpha_2 + \alpha_4\xi_1 \end{pmatrix}.$$

Eine derartige Singularität finden wir bei Betrachtung des Brusselators mit $\mathcal{S}_3$-Symmetrie auf der trivialen Lösung und $\tau_* \approx$ `2.96e−02`. Tabelle 5.7 zeigt die quadratische Konvergenz.

i	$\|u_i - u_*\|$	$\|\Delta\mu_i\|$	$\|F(u_i)\|$	RED
0	3.188e-01	4.176e-01	2.590e+00	7
1	1.507e-01	1.032e-01	1.017e-01	7
2	4.218e-03	1.261e-03	7.951e-04	6
3	9.785e-07	5.747e-07	2.818e-07	5
4	3.557e-13	1.818e-13	6.801e-14	4
5	1.668e-16	-	2.631e-15	-

Tabelle 5.7: Iterationsprotokoll für den Brusselator mit $\mathcal{S}_3$-Symmetrie

5.2.2 Höhere Verzweigung mit Rangdefekt 2

Beim Brusselator mit $\mathcal{D}_4$-Symmetrie tritt (unter anderen) eine Singularität mit der Normalform

$$h(\xi,\tau) = \begin{pmatrix} \xi_1^3 - 3\xi_1\xi_2^2 - \xi_1\tau \\ \xi_2^3 - 3\xi_1^2\xi_2 - \xi_2\tau \end{pmatrix}, \tag{5.9}$$

auf, eine höhere Verzweigung mit $k = 3,\ l = 0,\ p = 1$, und den Bestimmungsgleichungen

$$h = 0,\ h_\xi = 0,\ h_\tau = 0,\ \det(e_1^\intercal \partial^2 h) = 0,\ \det(e_2^\intercal \partial^2 h) = 0$$

also benötigen wir sieben Entfaltungsparameter, um ein quadratisches System zu erhalten. Die Entfaltung bestimmen wir aus

$$\begin{pmatrix} h \\ h_{\xi_1} \\ h_{\xi_2} \\ h_\tau \\ \det(e_1^\intercal \partial^2 h) \\ \det(e_2^\intercal \partial^2 h) \end{pmatrix} = \begin{pmatrix} 3\xi_1^2 - 3\xi_2^2 - \tau & -6\xi_1\xi_2 & -\xi_1 \\ -6\xi_1\xi_2 & 3\xi_2^2 - 3\xi_1^2 - \tau & -\xi_2 \\ 6\xi_1 & -6\xi_2 & -1 \\ -6\xi_2 & -6\xi_1 & 0 \\ -6\xi_2 & -6\xi_1 & 0 \\ -6\xi_1 & 6\xi_2 & -1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ -6 & 0 & 0 \\ 0 & -6 & 0 \end{pmatrix}$$

für $\xi_1 = \xi_2 = \tau = 0$ als

$$\tilde{h}(\xi,\tau,\alpha) = \begin{pmatrix} \xi_1^3 - 3\xi_1\xi_2^2 - \xi_1\tau + \alpha_1 + \alpha_3\xi_1 + \alpha_6\tau \\ \xi_2^3 - 3\xi_1^2\xi_2 - \xi_2\tau + \alpha_2 + \alpha_4\xi_1 + \alpha_5\xi_2 + \alpha_7\tau \end{pmatrix}.$$

Tabelle 5.8 zeigt die quadratische Konvergenz.

i	$\|u_i - u_*\|$	$\|\Delta\mu_i\|$	$\|F(u_i)\|$	RED
0	1.281e-01	1.125e+01	2.021e+00	8
1	1.255e+02	1.091e+01	2.395e-02	9
2	6.161e+00	2.107e+00	3.366e-01	9
3	2.399e+00	1.462e+00	2.267e-02	7
4	2.202e-02	8.684e-02	2.406e-03	7
5	1.035e-04	1.791e-03	3.147e-05	7
6	3.886e-08	4.527e-07	8.855e-09	6
7	2.711e-15	3.562e-14	4.757e-15	5
8	4.044e-26	-	1.722e-15	-

Tabelle 5.8: Iterationsprotokoll für den Brusselator mit $\mathcal{D}_4$-Symmetrie

5.2.3 Einfache Verzweigung mit Rangdefekt 3

Schließlich betrachten wir noch den Brusselator mit $\mathcal{S}_4$-Symmetrie. Dabei tritt ähnlich zur $\mathcal{S}_3$-Symmetrie auf der trivialen Lösung eine einfache Verzweigung bei $\tau_* \approx$ 1.128e+00 auf, in diesem Fall aber mit Rangdefekt $k = 3$. Die Bestimmungsgleichungen sind ebenfalls

$$h = 0, h_\xi = 0, h_\tau = 0,$$

hier benötigen wir allerdings elf Entfaltungsparameter. Diese wurden generisch gewählt, also $\tilde{h} = h + \begin{pmatrix}\alpha_1\\ \alpha_2\\ \alpha_3\end{pmatrix} + \begin{pmatrix}\alpha_4\\ \alpha_5\\ \alpha_6\end{pmatrix}\xi_1 + \begin{pmatrix}\alpha_7\\ \alpha_8\\ \alpha_9\end{pmatrix}\xi_2 + \begin{pmatrix}\alpha_{10}\\ \alpha_{11}\\ 0\end{pmatrix}\tau$, Tabelle 5.9 zeigt die Resultate der Iteration.

i	$\|u_i - u_*\|$	$\|\Delta\mu_i\|$	$\|F(u_i)\|$	RED
0	4.244e-03	3.580e-02	8.322e-01	6
1	1.273e-03	1.900e-03	1.407e-03	6
2	7.566e-08	4.446e-07	6.459e-07	5
3	1.304e-11	3.494e-14	6.819e-14	4
4	1.303e-11	-	6.463e-15	-

Tabelle 5.9: Iterationsprotokoll für den Brusselator mit $\mathcal{S}_4$-Symmetrie

Bricht man die Symmetrie, zum Beispiel in der ersten Zelle, findet man unter

Verwendung der Startwerte

$$x_1 = 6.0,\ x_2 = 1.1$$
$$x_3 = x_5 = x_7 = 0.67$$
$$x_4 = x_6 = x_8 = 5.5$$
$$x_9 = \tau = 0.033$$

ebenso wie bei der $\mathcal{S}_3$-Symmetrie eine einfache Verzweigung mit Rangdefekt 2. Auch hier erzielt Verfahren 4.4 quadratische Konvergenz, wie Tabelle 5.10 zeigt.

i	$\|u_i - u_*\|$	$\|\Delta\mu_i\|$	$\|F(u_i)\|$	RED
0	2.753e-03	6.801e-03	2.006e+00	7
1	1.428e-02	6.407e-03	1.183e-01	7
2	2.736e-05	3.621e-05	4.165e-03	6
3	9.749e-11	3.351e-10	4.521e-06	5
4	2.065e-11	-	2.483e-10	-

Tabelle 5.10: Iterationsprotokoll für den Brusselator mit gebrochener $\mathcal{S}_4$-Symmetrie

5.3 Weitere Beispiele

Darüber hinaus wurden für die Trigger-Schaltung [Sch00, Beispiel 4.6] und die diskretisierte Explosionsgleichung [Sch00, Beispiel 4.7] die Ergebnisse bestätigt.

i	$\|u_i - u_*\|$	$\|\Delta\mu_i\|$	$\|F(u_i)\|$	RED
0	3.622e-02	1.225e-01	2.253e-01	9
1	1.616e-02	1.539e-02	2.907e+00	11
2	1.474e-04	2.347e-03	8.100e-03	7
3	3.383e-06	6.199e-05	1.559e-03	6
4	2.163e-09	4.030e-08	4.340e-05	6
5	9.061e-16	1.520e-14	2.836e-08	5
6	4.515e-21	-	1.028e-14	-

Tabelle 5.11: Iterationsprotokoll für die Trigger Schaltung. Die Startwerte wurden wie in [Sch00] gewählt, u_* entspricht den dort errechneten Werten.

i	$\|u_i - u_*\|$	$\|\Delta\mu_i\|$	$\|F(u_i)\|$	RED
0	1.506e+00	1.663e-01	2.795e+00	7
1	2.809e-02	1.411e-03	3.390e-04	6
2	1.990e-06	1.278e-07	1.673e-06	5
3	1.637e-14	1.428e-14	5.489e-11	5
4	4.184e-19	-	2.552e-15	-

Tabelle 5.12: Iterationsprotokoll für die diskretisierte Explosionsgleichung mit $h = \frac{1}{8}$. Für u_* wurden die in [PSS99] angegeben Werte für $x_9 = 4.896597271$, $\tau = 1.307368582$ und $\alpha = 0.2457832826$ verwendet. Aufgrund mangelnder Vergleichswerte wurden für die restlichen Einträge von u_* die Ergebnisse des Verfahrens 4.4 verwendet.

Anhang A

Hilfsmittel

A.1 Satz über implizite Funktionen

Satz A.1. *Sei*

$$f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m, \quad (x, y) \mapsto f(x, y)$$

stetig und bezüglich y stetig differenzierbar. Ferner sei $(x_*, y_*) \in \mathbb{R}^n \times \mathbb{R}^m$ mit

$$f(x_*, y_*) = 0 \;\; \textit{und} \;\; \partial_y f(x_*, y_*) \; \textit{nichtsingulär.}$$

Dann existiert ein $\varepsilon > 0$ so, dass es zu jedem x mit $\|x - x_*\| < \varepsilon$ genau ein $y = g(x)$ mit

$$f(x, g(x)) = 0$$

gibt. Die Zuordnung $x \mapsto g(x)$ ist stetig und $g(x_*) = y_*$. Ist ferner f k-mal stetig differenzierbar, so ist g ebenfalls k-mal stetig differenzierbar. Dabei gilt:

$$g_x(x) \equiv -\left(f_y(x, g(x))\right)^{-1} f_x(x, g(x)). \tag{A.1}$$

Beweis. Siehe z.B. [AE98, Kapitel VII, Theorem 8.2.]. □

Literaturverzeichnis

[AE98] H Amann and J Escher. *Analysis II*. Grundstudium Mathematik. Birkhäuser, Basel, 1998.

[BB98] M.C. Batholomew-Biggs. Using forward accumulation for automatic differentiation of implicitly-defined functions. *Computational Optimization and Applications*, 9:65–84, 1998.

[BCG+92] C. Bischof, G. Corliss, L. Green, A. Griewank, K. Haigler, and P. Newman. Automatic differentiation of advanced cfd codes for multidisciplinary design. *Journal on Computing Systems in Engineering*, 3:625–637, 1992.

[BF94] Thomas Beck and Herbert Fischer. The if-problem in automatic differentiation. *Journal of Computational and Applied Mathematics*, 50(1–3):119 – 131, 1994.

[Bro65] C. G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 19(92):577–593, 1965.

[Deu06] P. Deuflhard. *Newton Methods for Nonlinear Problems. Affine Invariance and Adaptive Algorithms*. Springer Series in Computational Mathematics. 2nd edition, 2006.

[FP63] R. Fletcher and M. J. D. Powell. A rapidly convergent descent method for minimization. *The Computer Journal*, 6(2):163–168, 1963.

[GBC+93] Andreas Griewank, Christian Bischof, Alan Carle, George F. Corliss, and Karen Williamson. Derivative convergence for iterative equation solvers. *Optimization Methods and Software*, 2:321–355, 1993.

[Gil92] Jean Charles Gilbert. Automatic differentiation and iterative processes. *Optimization Methods and Software*, 1:13–21, 1992.

[GJU96] Andreas Griewank, David Juedes, and Jean Utke. Algorithm 755: ADOL-C: a package for the automatic differentiation of algorithms written in C/C++. *ACM Transactions on Mathematical Software*, 22:131–167, 1996.

[Gov97] W. Govaerts. Computation of singularities in large nonlinear systems. *SIAM J. Num. Ana.*, 34(3):867–880, 1997.

[GS79] M. Golubitsky and D.G. Schaeffer. A theory for imperfect bifurcation via singularity theory. *Comm. Pure. Appl. Math.*, 32(1):21–98, 1979.

[GS85] M. Golubitsky and D.G. Schaeffer. *Singularities and Groups in Bifurcation Theory: Vol. I.* Applied Mathematical Sciences 51. Springer-Verlag, 1985.

[GUW00] Andreas Griewank, Jean Utke, and Andrea Walther. Evaluating higher derivative tensors by forward propagation of univariate Taylor series. *Math. Comput.*, 69:1117–1130, 2000.

[GW08] Andreas Griewank and Andrea Walther. *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation.* Number 105 in Other Titles in Applied Mathematics. SIAM, Philadelphia, PA, 2nd edition, 2008.

[IEE08] IEEE Standard for Floating-Point Arithmetic. Technical report, Microprocessor Standards Committee of the IEEE Computer Society, 3 Park Avenue, New York, NY 10016-5997, USA, August 2008.

[JS85] A.D. Jepson and A. Spence. The numerical solution of nonlinear equations having several parameters, part I: scalar equations. *SIAM J. Num. Ana.*, 22(4):736–759, 1985.

[Kun90] Peter Kunkel. A unified approach to the numerical treatment of singular points. *Habilitationsschrift, Universität Oldenburg*, 1990.

[OR00] James M. Ortega and Werner C. Rheinboldt. *Iterative solution of nonlinear equations in several variables.* Society for Industrial and Applied Mathematics, 2000.

[PL68] I. Prigogine and R. Lefever. Symmetry breaking instabilities in dissipative systems. ii. *The Journal of Chemical Physics*, 48(4):1695–1700, 1968.

[PSS99] G. Pönisch, U. Schnabel, and H. Schwetlick. Computing multiple turning points by using simple extended systems and computational differentiation. *Optimization Methods and Software*, 10(4):639–668, 1999.

[PTVF92] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes in C (2nd ed.): the art of scientific computing*. Cambridge University Press, New York, NY, USA, 1992.

[Sch00] Uwe Schnabel. *Berechnung singulärer Punkte nichtlinearer Gleichungssysteme*. PhD thesis, Technische Universität Dresden, 2000.

Erklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann.

Ort	Datum	Unterschrift