



TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

DIPLOMARBEIT

Regularisierungsmethoden für Differential–Algebraische Gleichungssysteme

Ausgeführt am Institut für

Analysis und Scientific Computing

der Technischen Universität Wien
unter der Anleitung von

Ao.Univ.Prof. Dipl.–Ing. Dr.techn. Felix Breitenecker

durch

Carina Pöll

Römergasse 50/1/13
1160 Wien
Österreich

Wien, im September 2014

Kurzfassung

Durch die Verwendung objektorientierter Simulationswerkzeuge zur Modellierung physikalischer oder mechanischer Systeme entstehen differential–algebraische Gleichungssysteme mit hohem differentiellm Index. Der differentielle Index gibt die minimale Anzahl an Ableitungen an, die notwendig sind, um aus dem resultierenden differenzierten System ein gewöhnliches Differentialgleichungssystem extrahieren zu können. Das numerische Lösen von differential–algebraischen Gleichungssystemen ist mit herkömmlichen Lösungsmethoden für gewöhnliche Differentialgleichungen im Allgemeinen sehr komplex und daher aufwändig bzw. kann sogar unmöglich sein. Daher sind Methoden notwendig, um diesem Problem zu begegnen, was zur sogenannten Indexreduktion führt. Das Ziel der Indexreduktion, ist das differential–algebraische Gleichungssystem in ein differential–algebraisches Gleichungssystem mit niedrigerem Index beziehungsweise ein gewöhnliches Differentialgleichungssystem umzuwandeln.

Diese Arbeit soll einen Überblick über gängige Regularisierungsmethoden bieten, wobei eine Klassifikation der verschiedenen Ansätze vorgenommen wird. Diese Klassifikation unterteilt die verschiedenen Ansätze in drei Bereiche: Indexreduktion mithilfe von Differentiation, Stabilisierung der numerischen Lösung durch Projektion und (lokale) Transformation des Zustandsraums. Nach der Klassifikation wird jeder Ansatz im Detail erklärt und vorgestellt, wobei auch Vor- und Nachteile der verschiedenen Methoden diskutiert werden. Bei der Indexreduktion mithilfe der Differentiation werden drei verschiedene Methoden betrachtet: Differentiation und Ersetzen der Zwangsbedingung, Baumgarte–Methode und Pantelides–Algorithmus. Bei dem Ansatz mithilfe von Projektionen werden zwei verschiedene Methoden betrachtet, die sogenannte orthogonale Projektionsmethode und die symmetrische Projektionsmethode. Die Idee der Transformation des Zustandsraums ist, ein gewöhnliches Differentialgleichungssystem auf einer Mannigfaltigkeit zu erhalten.

Zum Vergleich der verschiedenen Methoden werden anschließend die vorher beschriebenen Ansätze anhand von Beispielen vorgeführt. Bei den beiden Beispielen handelt es sich um mechanische Systeme: einerseits um die Bewegungsgleichungen eines Pendels auf einer Kreisbahn in kartesischen Koordinaten, andererseits um die Bewegungsgleichungen eines Doppelpendels in kartesischen Koordinaten, wobei dieses System ein chaotisches Verhalten zeigt. Für die Vergleichbarkeit der verschiedenen Methoden werden die numerisch erhaltenen Lösungen und die Abweichung zur Zwangsbedingung betrachtet. Dadurch können die unterschiedlichen Ansätze hinsichtlich der erhaltenen Lösungen gegenübergestellt werden. Schließlich kann dadurch für das gegebene differential–algebraische Gleichungssystem entschieden werden, welche Methode sich am besten für das Lösen ebendieses eignet.

Abstract

The use of object-oriented simulation tools for modelling of physical or mechanical systems leads to systems of differential-algebraic equations with a high differential index. The differential index indicates the minimal number of differentiations of the system, which are necessary to extract a system of ordinary differential equations from the differentiated system. In general the numerical solution of differential-algebraic equation systems with high index by conventional solution methods for ordinary differential equations is very complex or may even be impossible. Therefore methods for solving this problem are necessary, which leads to the so-called index reduction. The aim of the index reduction is to convert the system of differential-algebraic equations into a system of differential-algebraic equations of lower index or a system of ordinary differential equations.

This work aims to provide an overview of common regularisation methods. Additionally, a classification of these different approaches is done. This classification divides the different approaches into three areas: index reduction with the use of differentiation, stabilization of the numerical solution by projection and (local) transformation of the state space. According to the classification each approach is presented and explained in detail. Then advantages and disadvantages of the different methods are discussed. Three different methods of index reduction with the use of differentiation are considered: differentiation and replacement of the constraint, the Baumgarte-Method and the Pantelides-Algorithm. There are two different methods using projections, called the orthogonal projection method and the symmetric projection method. The idea of the method using transformation of the state space is to obtain a system of ordinary differential equations on a manifold.

In order to compare the different methods, the approaches described above are demonstrated by examples. The two examples are mechanical systems. On the one hand the equations of the motion of a pendulum on a circular path in Cartesian coordinates are considered. On the other hand, the equations of motion of the double pendulum in Cartesian coordinates, which shows chaotic behaviour, are used. For the comparison of the different methods, the obtained numerical solutions and the deviation from the constraint equations are considered. Therefore the numerical solutions of the distinct approaches can be compared. Finally it is possible to decide which method is suitable for solving the given system of differential-algebraic equations.

Danksagung

An dieser Stelle möchte ich mich bei all denjenigen bedanken, die mich während der Anfertigung dieser Diplomarbeit und während des gesamten Studiums unterstützt und motiviert haben. Besonderer Dank gebührt Professor Felix Breitenecker für die Betreuung der Diplomarbeit, die Möglichkeit als Tutorin tätig zu sein und die herzliche Atmosphäre in der Forschungsgruppe.

Weiters möchte ich Irene Hafner und Bernhard Heinzl für die fachliche Beratung und Unterstützung danken.

Ein großes Dankeschön gilt auch allen Mitgliedern der Forschungsgruppe. Dabei möchte ich mich besonders bei Stefanie Nadine Winkler und Andreas Körner bedanken – für die gute Zusammenarbeit und die dabei entstandene Freundschaft.

Außerdem möchte ich mich an dieser Stelle bei Niki Popper bedanken.

Ein besonderer Dank gilt meinen Eltern, die mir dieses Studium ermöglicht haben und mich immer unterstützt haben. Sie haben mir eine wunderschöne Kindheit geschenkt und sind immer für mich da. Diese wunderbare Kindheit war geprägt von vier großartigen Personen, meinen Großeltern Barbara und Alois Pöll und Rosa und Hubert Vogl. Diesen möchte ich für meine Kindheit und ihre ständige Begleitung und Fürsorge danken.

An dieser Stelle möchte ich mich noch bei den beiden Personen, die mich während meines ganzen Studiums begleitet haben, bedanken. Für die vielen Stunden der Vorbereitung auf Übungen, Prüfungen und schließlich auch die Zusammenarbeit bei der Bachelorarbeit möchte ich mich herzlich bei Nina Winterer und Andreas Bauer bedanken.

Abschließend möchte ich meinem Freund Matthias danken, der mir während dieser letzten Phase meines Studiums immer zur Seite stand. Vielen Dank für die motivierenden Worte, Ratschläge und die Hilfe beim Korrigieren der Diplomarbeit.

Ich widme diese Diplomarbeit meiner Großmutter,

Rosa Katharina Vogl,

die leider kurz vor Beendigung meines Studiums verstorben ist.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Aufgabenstellung	2
1.3	Ziele	2
1.4	Überblick	3
2	Mathematische Grundlagen	4
2.1	Grundlegende Begriffe und Resultate	4
2.1.1	Analysis	4
2.1.2	Lineare Algebra	6
2.1.3	Numerik	9
2.1.4	Variationsrechnung	11
2.2	Differentialgleichungen auf Mannigfaltigkeiten	14
3	Definition und Eigenschaften von DAEs	15
3.1	Grundlegende Definitionen zu DAEs und Index	15
3.2	Spezielle DAEs	16
3.3	DAEs mit Index 1	18
3.4	DAEs bei mechanischen Systemen	20
4	Regularisierungsmethoden	26
4.1	Überblick	26
4.2	Indexreduktion durch Differentiation	27
4.2.1	Differentiation der Zwangsbedingung	27
4.2.2	Baumgarte–Methode	28
4.2.3	Pantelides–Algorithmus	28

4.3	Stabilisierung durch Projektion	29
4.3.1	Standardprojektionsmethode	30
4.3.2	Symmetrische Projektionsmethode	31
4.4	Methoden basierend auf lokaler Zustandsraumtransformation	33
5	Methodenvergleich – Fallstudie Pendel	37
5.1	Analyse der DAE	38
5.2	Indexreduktionsmethoden	41
5.2.1	Indexreduktion durch Differentiation	41
5.2.1.1	Differentiation der Zwangsbedingung	41
5.2.1.2	Baumgarte-Methode	41
5.2.1.3	Pantelides-Algorithmus	42
5.2.2	Stabilisierung durch Projektion	47
5.2.2.1	Standardprojektionsmethode	48
5.2.2.2	Symmetrische Projektionsmethode	52
5.2.3	Methoden basierend auf lokaler Zustandsraumtransformation	52
5.3	Simulationen	54
5.3.1	ode-Solver	55
5.3.2	Simulationen der Index-0-Formulierung	55
5.3.3	Simulationen der Index-1-Formulierung	58
5.3.4	Simulationen der Indexreduktionsmethoden	60
5.3.4.1	Differentiation der Zwangsbedingung	60
5.3.4.2	Baumgarte-Methode	61
5.3.4.3	Pantelides-Algorithmus	63
5.3.4.4	Standardprojektionsmethode	66
5.3.4.5	Symmetrische Projektionsmethode	68
5.3.4.6	Methoden basierend auf lokaler Zustandsraumtransformation	69
5.3.5	Ergebnisse	70
6	Methodenvergleich – Fallstudie Doppelpendel	75
6.1	Analyse der DAE	76
6.2	Indexreduktionsmethoden	80
6.2.1	Indexreduktion durch Differentiation	80
6.2.1.1	Differentiation der Zwangsbedingung	80
6.2.1.2	Baumgarte-Methode	80
6.2.1.3	Pantelides-Algorithmus	81
6.2.2	Stabilisierung durch Projektion	83
6.2.2.1	Standardprojektionsmethode	84
6.2.2.2	Symmetrische Projektionsmethode	86

6.2.3	Methoden basierend auf lokaler Zustandsraumtransformation . . .	86
6.3	Simulationen	88
6.3.1	Simulationen der Index-0-Formulierung	89
6.3.2	Simulationen der Index-1-Formulierung	90
6.3.3	Simulationen der Indexreduktionsmethoden	92
6.3.3.1	Differentiation der Zwangsbedingung	92
6.3.3.2	Baumgarte-Methode	93
6.3.3.3	Pantelides-Algorithmus	98
6.3.3.4	Standardprojektionsmethode	100
6.3.3.5	Symmetrische Projektionsmethode	103
6.3.3.6	Methoden basierend auf lokaler Zustandsraumtransformation	105
6.3.4	Ergebnisse	107
7	Zusammenfassung	111
8	Ausblick	113
	Abbildungsverzeichnis	114
	Tabellenverzeichnis	117
	Literaturverzeichnis	119

In diesem Kapitel wird motiviert, warum das Thema dieser Arbeit eine interessante Aufgabenstellung bietet und welche Ziele erhofft werden. Anschließend wird ein Überblick über die folgenden Kapitel gegeben.

1.1 Motivation

Die Verwendung von objektorientierter, akausaler Modellbeschreibung besitzt verschiedene Vor- und Nachteile. Aufgrund der direkten Abbildung der Systemstruktur anstatt der Gleichungen ist die Darstellung des Systems sehr gut erkennbar. Außerdem ist durch den modularen Aufbau, bei dem jede Komponente lokal beschrieben wird und die einzelnen Komponenten zu einem Gesamtsystem zusammengefasst werden, eine Erweiterung des Systems flexibel und leicht durchführbar. Neben diesen Vorteilen gibt es auch Nachteile dieser Modellbeschreibung. Zu erwähnen ist hier, dass jegliche Einschränkung des Systems durch Zwangsbedingungen realisiert werden muss und daher ein differential-algebraisches Gleichungssystem entsteht.

Eine objektorientierte akausale Modellbeschreibung für physikalische oder mechanische Systeme, wie beispielsweise Modelica oder MATLAB/Simscape, führt in der Regel auf differential-algebraische Gleichungssysteme mit nichttrivialem differentielllem Index. Das numerische Lösen von differential-algebraischen Gleichungssystemen mit hohem differentielllem Index ist mit herkömmlichen Lösungsmethoden für gewöhnliche Differentialgleichungen im Allgemeinen sehr komplex und daher aufwändig bzw. kann sogar unmöglich sein. Diese Problematik führt auf die sogenannte Indexreduktion, bei dem das gegebene differential-algebraische Gleichungssystem in ein differential-algebraisches Gleichungssystem mit niedrigerem differentielllem Index oder ein gewöhnliches Differentialgleichungssystem umformuliert wird und dadurch schließlich mit Methoden für gewöhnliche Differentialgleichungen gelöst werden kann. Aufgrund der großen Unterschiede

hinsichtlich der Struktur und Eigenschaften von differential–algebraischen Gleichungssystemen, wie beispielsweise der differentielle Index, finden sich in der Literatur eine ganze Reihe verschiedener Ansätze und Methoden zur Reduktion des differentiellen Index bzw. der Regularisierung des Systems.

1.2 Aufgabenstellung

Am Beginn der Arbeit steht eine ausführliche Literaturrecherche zum Thema differential–algebraische Gleichungssysteme, Indexreduktion und Regularisierungsmethoden, um die grundlegenden Begriffe und Methoden kennenzulernen. Im Folgenden sollen verschiedene Ansätze für differential–algebraische Gleichungssysteme evaluiert und verglichen werden. Bei der Auswahl der Methoden muss beachtet werden, dass der entsprechende Ansatz nicht zu speziell ist und daher nur für eine sehr kleine Klasse von differential–algebraischen Gleichungssystemen verwendet werden kann. Aufgrund dessen werden im Folgenden gängige Methoden zur numerischen Lösung von differential–algebraischen Gleichungssystemen betrachtet. Im Vordergrund stehen die Ansätze von Pantelides und Baumgarte. Diese beiden Ansätze beruhen im Wesentlichen auf der Differentiation der Zwangsbedingung. Eine weitere Idee ist, die numerische Lösung, wenn sie die Zwangsbedingung nicht mehr erfüllt, auf die Lösungsmannigfaltigkeit zu projizieren. Eine andere Herangehensweise für das Lösen von differential–algebraischen Gleichungssystemen ist, die Modellbeschreibung durch Anwenden geeigneter Koordinatentransformationen zu ändern und somit das Reduzieren des Index zu umgehen. Um die Vorgehensweise und Funktionalität der Ansätze besser zu verstehen, werden die verschiedenen Ansätze anhand einfacher Beispiele durchgerechnet. Anschließend sollen diese Ansätze durch Studien an ausgewählten Beispielen miteinander verglichen werden, das heißt es soll herausgefunden werden, für welche Problemstellung welche Methode im Vergleich hinsichtlich der Performance von Vorteil bzw. Nachteil ist und bei welchen Aufgabenstellungen eine Methode versagt. Die Implementierung der Beispiele erfolgt in MATLAB und die entstehenden gewöhnlichen bzw. differential–algebraischen Gleichungssysteme werden mit entsprechenden Differentialgleichungslösern, die in MATLAB vorimplementiert sind, gelöst.

1.3 Ziele

Das Ziel dieser Masterarbeit ist die Evaluierung von Methoden zur Indexreduktion und Regularisierung von differential–algebraischen Gleichungssystemen sowie die Analyse und der Vergleich dieser Methoden. Durch Studien anhand einfacher mechanischer Beispiele werden Vor- und Nachteile und Grenzen der jeweiligen Methoden zur Reduktion des differentiellen Index aufgezeigt. Schlussendlich wäre es wünschenswert herauszufin-

den für welche Problemstellung welcher Ansatz zum numerischen Lösen eines gegebenen differential–algebraischen Gleichungssystems geeignet ist. Mithilfe dieser Erkenntnisse könnte das numerische Lösen solcher Gleichungssysteme in der Praxis effizienter durchgeführt werden.

1.4 Überblick

In Kapitel 2 werden mathematische Definitionen und Grundlagen aus verschiedenen mathematischen Themengebieten, die später verwendet werden, zitiert. In Kapitel 3 werden die Begriffe „differential–algebraisches Gleichungssystem“ und „differentieller Index“ definiert. Weiters werden Eigenschaften von differential–algebraischen Gleichungssystemen mit spezieller Gestalt präsentiert und die Bewegungsgleichungen mechanischer Systeme hergeleitet. In Kapitel 4 werden zuerst die gängigsten Ansätze in Bereiche gegliedert, um die verschiedenen Herangehensweisen abzugrenzen. Anschließend werden die einzelnen Methoden im Detail beschrieben und erklärt. Diese Methoden werden in Kapitel 5 und 6 anhand von zwei Beispielen durchgerechnet. Bei dem ersten Beispiel handelt es sich um die Bewegung eines Pendels auf einer Kreisbahn in kartesischen Koordinaten, welches ein klassisches Beispiel eines differential–algebraischen Gleichungssystems ist. Das zweite Beispiel ist die Bewegung eines Doppelpendels in kartesischen Koordinaten, wobei das Doppelpendel im Gegensatz zum einfachen Pendel ein chaotisches Verhalten zeigt, siehe [1]. Diese beiden differential–algebraischen Gleichungssysteme besitzen Index drei. Anschließend werden diese differential–algebraischen Gleichungssysteme mit MATLAB gelöst und die Simulationsergebnisse präsentiert. Abschließend werden in den beiden letzten Kapiteln die Ergebnisse zusammengefasst und diskutiert.

In diesem Kapitel werden später verwendete mathematische Begriffe und Grundlagen aus verschiedenen Themengebieten zitiert.

2.1 Grundlegende Begriffe und Resultate

In diesem Abschnitt werden Definitionen und Resultate aufgelistet.

2.1.1 Analysis

In diesem Abschnitt werden Resultate aus der Analysis (siehe [2], [3] und [4]), die später verwendet werden, zitiert.

Definition 2.1 (Taylorsches Polynom). Sei $n \in \mathbb{N}$, $I \subseteq \mathbb{R}$ ein Intervall und $f: I \rightarrow \mathbb{R}(\mathbb{C})$ liege in $C^n(I)$. Ist $y \in I$ fest, so wird das Polynom (in der Variablen x)

$$T_n(x) = \sum_{k=0}^n \frac{(x-y)^k}{k!} f^{(k)}(y), \quad (2.1)$$

das n -te Taylorsche Polynom an der Anschlussstelle y genannt.

Definition 2.2 (Norm, normierter Raum). Sei X ein Vektorraum über $\mathbb{R}(\mathbb{C})$. Eine Abbildung $\|\cdot\|: X \rightarrow [0, \infty)$ heißt Norm, wenn gilt:

- Dreiecksungleichung: $\|x + y\| \leq \|x\| + \|y\|$, $x, y \in X$.
- $\|\lambda x\| = |\lambda| \|x\|$, $x \in X$, $\lambda \in \mathbb{R}(\mathbb{C})$.
- $\|x\| > 0$ für alle $x \neq 0$.

Ist auf dem Vektorraum X eine Norm gegeben, so spricht man von $(X, \|\cdot\|)$ als einem normierten Raum.

Die sogenannte euklidische Norm $\|\cdot\|_2$ ist für einen Vektor $x \in \mathbb{R}^n$ definiert durch

$$\|x\|_2 := \sqrt{\sum_{i=1}^n |x_i|^2}. \quad (2.2)$$

$(\mathbb{R}^n, \|\cdot\|_2)$ ist ein normierter Raum über dem Vektorraum \mathbb{R} .

Definition 2.3 (Projektion auf die j -te Komponente). Die Abbildung $\pi_j : \mathbb{R}^p \rightarrow \mathbb{R}$ mit $j = 1, \dots, p$, die definiert ist durch

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_p \end{pmatrix} \mapsto x_j, \quad (2.3)$$

wird Projektion auf die j -te Komponente genannt.

Im folgenden Satz werden mit dF_1 bzw. dF_2 die Teile $\left(\frac{\partial F}{\partial x_j}\right)_{j=1}^n$ bzw. $\left(\frac{\partial F}{\partial x_j}\right)_{j=n+1}^{n+m}$ der Jacobimatrix bezeichnet.

Satz 2.4 (Hauptsatz über implizite Funktionen). Sei $D \subseteq \mathbb{R}^{n+m}$ offen und sei $F: D \rightarrow \mathbb{R}^m$, $F \in C^1$. Weiters sei $(a, b) \in D$ sodass $F(a, b) = 0$ und $dF_2(a, b)$ invertierbar ist, das heißt

$$\det \left(\frac{\partial F_i}{\partial x_{n+j}}(a, b) \right)_{i,j=1}^m \neq 0. \quad (2.4)$$

- Dann existieren offene Kugeln $U = U_\delta(a) \subseteq \mathbb{R}^n$ um a und $V = U_\rho(b) \subseteq \mathbb{R}^m$ um b mit $U \times V \subseteq D$, sowie eine stetige Funktion $g: U \rightarrow V$, sodass

$$F(x, g(x)) = 0, \quad x \in U. \quad (2.5)$$

- Die Funktion g löst die Gleichung $F(x, y) = 0$, $x \in U$, $y \in V$ vollständig in dem Sinn, dass wenn $(x, y) \in U \times V$ mit $F(x, y) = 0$ immer $y = g(x)$ gelten muss. Insbesondere ist $b = g(a)$.

- Die Funktion g hat in a partielle Ableitungen ($j = 1, \dots, n$)

$$\frac{\partial g}{\partial x_j} = -dF_2(a, g(a))^{-1} \frac{\partial F}{\partial x_j}(a, g(a)). \quad (2.6)$$

- $dF_2(u, v)$ ist für alle $(u, v) \in U \times V$ invertierbar; die Funktion g ist auf U stetig differenzierbar und es gilt

$$dg(x) = -dF_2(x, g(x))^{-1} dF_1(x, g(x)). \quad (2.7)$$

Definition 2.5 (Einbettung, Mannigfaltigkeit). Für $M \subseteq \mathbb{R}^p$, $p \geq 1$, heißt eine Abbildung $\phi: D \rightarrow M$ mit $D \subseteq \mathbb{R}^d$, $0 < d \leq p$ eine d -dimensionale Einbettung in M , wenn

- $\emptyset \neq D$ offene Teilmenge von \mathbb{R}^d ist, $\phi(D)$ offene Teilmenge von M bezüglich der Spurtopologie ist, das heißt $\phi(D) = M \cap U$ für eine in \mathbb{R}^p bezüglich der Euklidischen Topologie offene Teilmenge U , und $\phi: D \rightarrow \phi(D)$ ein Homöomorphismus ist,
- ϕ als Abbildung von D nach \mathbb{R}^p stetig differenzierbar ist,
- $d\phi(s)$ für alle $s \in D$ injektiv ist, das heißt $d\phi(s)$ hat maximalen Rang d .

Eine nichtleere Teilmenge $M \subseteq \mathbb{R}^p$ heißt d -dimensionale Mannigfaltigkeit im \mathbb{R}^p , falls es zu jedem $x \in M$ eine d -dimensionale Einbettung $\phi: D \rightarrow M$ mit $x \in \phi(D)$ gibt.

Lemma 2.6 (Implizit definierte Mannigfaltigkeit). Sei $O \subseteq \mathbb{R}^p$ offen und $F: O \rightarrow \mathbb{R}^{p-d}$ stetig differenzierbar, wobei $0 < d < p$. Sei

$$M = \{x \in O : F(x) = 0\}. \tag{2.8}$$

Hat $dF(x)$ für alle $x \in M$ vollen Rang, also $p - d$, so ist M eine d -dimensionale Mannigfaltigkeit.

Definition 2.7 (Idempotent). Eine Abbildung $f: M \rightarrow N$ ist idempotent, wenn $f^2 = f \circ f = f$ erfüllt ist.

Definition 2.8 (Projektion). Eine Projektion ist eine Abbildung $P: M \rightarrow N$, wobei M, N beliebige Mengen mit $N \subset M$ sind, die idempotent ist.

2.1.2 Lineare Algebra

Im ersten Teil dieses Abschnitts werden Resultate aus der Linearen Algebra (hauptsächlich aus [5], [6] und [7]), die später verwendet werden, zitiert und bewiesen.

Definition 2.9 (Symmetrische Matrix). Eine Matrix $A \in \mathbb{R}^{n \times n}$ ist symmetrisch genau dann, wenn $A = A^T$.

Definition 2.10 (Orthogonale Matrix). Eine Matrix $Q \in \mathbb{R}^{n \times n}$ heißt orthogonal, falls $Q^T Q = E_n$.

Definition 2.11 (Verallgemeinerte obere Dreiecksmatrix). $R \in \mathbb{R}^{n \times m}$ heißt eine verallgemeinerte obere Dreiecksmatrix, falls $r_{ij} = 0 \quad \forall i > j$.

Definition 2.12 (QR-Zerlegung). Zu $A \in \mathbb{R}^{n \times m}$ heißt eine Faktorisierung $A = QR$, wobei $Q \in \mathbb{R}^{n \times n}$ eine orthogonale Matrix und $R \in \mathbb{R}^{n \times m}$ eine verallgemeinerte obere Dreiecksmatrix ist, eine QR-Zerlegung.

Definition 2.13 (Eigenwert, Eigenvektor). Sei $A \in \mathbb{R}^{n \times n}$ eine quadratische Matrix. Ein Skalar $t \in \mathbb{C}$ heißt Eigenwert von A , wenn es einen Vektor $a \in \mathbb{C}^n \setminus \{0\}$ mit

$$Aa = ta \quad (2.9)$$

gibt. Ein Vektor $a \in \mathbb{C}^n \setminus \{0\}$, zu dem es ein $t \in \mathbb{C}$ mit $Aa = ta$ gibt, heißt Eigenvektor von A (zum Eigenwert t).

Satz 2.14. Sei $A \in \mathbb{R}^{n \times n}$ eine symmetrische Matrix. Dann sind alle Eigenwerte von A reell und die Eigenvektoren von A bilden eine Basis des \mathbb{R}^n , wobei die Matrix V spaltenweise bestehend aus den Eigenvektoren orthogonal ist.

Satz 2.15. Sei $A \in \mathbb{R}^{n \times n}$ eine symmetrische Matrix. Dann ist A genau dann positiv definit, wenn alle Eigenwerte von A positiv sind.

Satz 2.16. Sei $A \in \mathbb{R}^{n \times n}$ SPD (symmetrisch positiv definit), dann ist A^{-1} SPD.

Beweis. Da A SPD ist, gibt es eine orthogonale Matrix $V \in \mathbb{R}^{n \times n}$ (bestehend aus den Eigenvektoren) und eine Diagonalmatrix D (bestehend aus den Eigenwerten) mit $D_{ii} > 0$ für alle $i = 1, \dots, n$. Es gilt also

$$V^T A V = D \Rightarrow (V^T A V)^{-1} = V^T A^{-1} V = D^{-1} = \hat{D}, \quad (2.10)$$

wobei $\hat{D}_{ii} = \frac{1}{D_{ii}} > 0$, das heißt A^{-1} hat positive Eigenwerte.

$$(A^{-1})^T = (V D^{-1} V^T)^T = V D^{-1} V^T = A^{-1} \quad (2.11)$$

Somit ist A^{-1} SPD. □

Satz 2.17. Sei $A \in \mathbb{R}^{n \times n}$ SPD und $S \in \mathbb{R}^{n \times m}$ mit vollem Spaltenrang m . Dann ist $S^T A S$ SPD.

Beweis. $S^T A S$ ist symmetrisch, da

$$(S^T A S)^T \stackrel{A=A^T}{=} S^T A S. \quad (2.12)$$

Es bleibt zu zeigen, dass $S^T A S$ positiv definit ist. Dazu wird zuerst die Matrix S näher betrachtet. $S: \mathbb{R}^m \rightarrow \mathbb{R}^n$ ist injektiv, da

$$m = \dim(\mathbb{R}^m) = \dim \ker(S) + \text{Rang}(S) = 0 + m. \quad (2.13)$$

Also wird durch $x \mapsto Sx$ ausschließlich der Nullvektor auf Null abgebildet.

$$x^T (S^T A S) x = x^T S^T A S x = (Sx)^T A (Sx) \stackrel{A \text{ SPD}}{>} 0 \quad \forall x \in \mathbb{R}^m \setminus \{0\} \quad (2.14)$$

Somit folgt insgesamt, dass $S^T A S$ SPD ist. □

Definition 2.18 (Moore–Penrose–pseudoinverse Matrix). Sei $A \in \mathbb{R}^{m \times n}$. Eine Matrix $B \in \mathbb{R}^{m \times n}$ heißt Moore–Penrose–pseudoinverse Matrix von A , wenn

$$AB \quad \text{und} \quad BA \tag{2.15}$$

symmetrisch sind und

$$A = ABA \quad \text{und} \quad B = BAB \tag{2.16}$$

gilt. Im Folgenden wird die Matrix B mit A^+ bezeichnet.

Satz 2.19. Sei $A \in \mathbb{R}^{m \times n}$, so dass $A^T A$ invertierbar ist. Dann gilt

$$A^+ = (A^T A)^{-1} A^T. \tag{2.17}$$

Im zweiten Teil dieses Abschnitts werden speziell Resultate der Tensorrechnung behandelt, die aus [8] und [9] entnommen wurden.

Im Folgenden wird die Einsteinsche Summenkonvention verwendet, das heißt $b_i = t_{ij} a_j$ bedeutet $b_i = \sum_{j=1}^n t_{ij} a_j$. Für die folgenden Definitionen sei $R = (r_{ij})_{i,j=1}^n$ eine orthogonale Matrix.

Definition 2.20 (Tensor zweiter Stufe). Ist T eine Menge von zweifach indizierten Termen t_{ij} mit dem Transformationsverhalten $t'_{kl} = r_{ki} r_{lj} t_{ij}$, dann ist T ein Tensor zweiter Stufe.

Definition 2.21 (Tensor dritter Stufe). Ist S eine Menge von dreifach indizierten Termen s_{ijk} mit dem Transformationsverhalten $s'_{ijk} = r_{ir} r_{js} r_{kt} s_{rst}$, dann ist S ein Tensor dritter Stufe.

Eigenschaften von Tensoren:

- Ein Tensor 0. Stufe ist ein Skalar (kein Index).
- Ein Tensor 1. Stufe ist ein Vektor (1 Index).
- Ein Tensor 2. Stufe ist eine Matrix (2 Indizes).
- Ein Tensor 3. Stufe besitzt 3 Indizes.

Rechenregeln für Tensoren:

- Tensorprodukt: Das Tensorprodukt zweier Tensoren führt stets (außer bei Tensoren nullter Stufe) zu einem Tensor höherer Ordnung. Die Stufe des Produkttensors ist dabei durch die Summe der Stufen der beteiligten Tensoren gegeben.

- **Verjüngung:** Ein Tensor (höherer Stufe) kann einer Verjüngung unterworfen werden. Dabei wird über zwei Indizes summiert. Dies wird durch das Gleichsetzen zweier Indizes und anschließender Summation über diese ausgedrückt. Als Ergebnis wird ein Tensor mit einer um zwei niedrigeren Stufe erhalten.
- **Überschiebung:** Bei der Überschiebung von zwei Tensoren bildet man zuerst das Tensorprodukt und führt danach eine Verjüngung aus.

2.1.3 Numerik

Im ersten Teil dieses Abschnitts werden Resultate aus der Numerik (siehe [5]), die später verwendet werden, zitiert.

Gegeben sei eine Funktion $F: \mathbb{R}^n \rightarrow \mathbb{R}^n$. Von dieser Funktion wird die Nullstelle x^* , also $F(x^*) = 0$, gesucht. Diese Nullstelle kann mit dem Newton-Verfahren gefunden werden.

Definition 2.22 (Newton-Verfahren). Ein Iterationsschritt des Newton-Verfahrens angewendet auf die Funktion F lautet

$$x_{n+1} = x_n - \left(\frac{\partial F}{\partial x} \right)^{-1} (x_n) F(x_n). \quad (2.18)$$

Im zweiten Teil dieses Abschnitts werden Resultate aus der Numerik von Differentialgleichungen (siehe [10]), die später verwendet werden, zitiert.

Sei $G \subseteq \mathbb{R}^{n+1}$ ein Gebiet, $(t_0, y_0) \in G$, $f \in C(G, \mathbb{R}^n)$ und $y_0 \in \mathbb{R}^n$. Es wird das Differentialgleichungssystem

$$\dot{y} = f(t, y) \quad \text{mit} \quad y(t_0) = y_0 \quad (2.19)$$

betrachtet. Es gibt verschiedene numerische Verfahren zur Lösung solcher Differentialgleichungssysteme. Zuerst wird auf Einschrittverfahren näher eingegangen. Im Folgenden bezeichnet y_i die numerische Lösung zum Zeitpunkt t_i , wobei $i = 0, \dots, N$, und $h_i = t_{i+1} - t_i$ die Schrittweite, wobei $i = 0, \dots, N - 1$.

Definition 2.23 (Einschrittverfahren). Ein numerisches Verfahren, bei dem zu gegebenem y_0 die Werte y_i , $i = 1, \dots, N$ durch eine Rekursion

$$y_{i+1} = y_i + h_i \Psi(t_i, y_i, y_{i+1}, h_i), \quad i = 0, \dots, N - 1 \quad (2.20)$$

bestimmt werden, heißt Einschrittverfahren. Die Funktion Ψ heißt Inkrementfunktion. Hängt die Funktion Ψ nicht explizit von y_{i+1} ab, so wird von einem expliziten Einschrittverfahren gesprochen, andernfalls von einem impliziten.

Beispiele für Einschrittverfahren:

- Das explizite und das implizite Euler–Verfahren sind gegeben durch

$$\Psi(t_i, y_i, y_{i+1}, h_i) = f(t_i, y_i) \quad \text{und} \quad \Psi(t_i, y_i, y_{i+1}, h_i) = f(t_i + h_i, y_{i+1}). \quad (2.21)$$

- Die Trapezregel ist ebenfalls ein implizites Verfahren mit der Inkrementfunktion

$$\Psi(t_i, y_i, y_{i+1}, h_i) = \frac{f(t_i + h_i, y_{i+1}) + f(t_i, y_i)}{2}. \quad (2.22)$$

Im Folgenden wird ein explizites Einschrittverfahren der Form

$$y_{i+1} = y_i + h_i \Psi(t_i, y_i, h_i) \quad (2.23)$$

betrachtet.

Definition 2.24 (Konsistenzfehler, Ordnung). Sei $G \subseteq \mathbb{R}^2$ ein Gebiet, $f \in C(G)$ lokal lipschitzstetig im zweiten Argument. Sei die Inkrementfunktion Ψ für ein $h > 0$ auf der Menge $\mathcal{G} := G \times [0, h] \subset \mathbb{R}^3$ definiert. Für $(\tilde{t}, \tilde{y}, h) \in \mathcal{G}$ ist der Konsistenzfehler $\tau(\tilde{t}, \tilde{y}, h)$ definiert als

$$\tau(\tilde{t}, \tilde{y}, h) = y_{\tilde{t}, \tilde{y}}(\tilde{t} + h) - (\tilde{y} + h\Psi(\tilde{t}, \tilde{y}, h)), \quad (2.24)$$

wobei die Funktion $t \mapsto y_{\tilde{t}, \tilde{y}}$ die Lösung von

$$\dot{y}_{\tilde{t}, \tilde{y}}(t) = f(t, y_{\tilde{t}, \tilde{y}}(t)), \quad y_{\tilde{t}, \tilde{y}}(\tilde{t}) = \tilde{y} \quad (2.25)$$

ist. Gilt für jedes $(\tilde{t}, \tilde{y}) \in G$

$$\lim_{h \rightarrow 0^+} \frac{\tau(\tilde{t}, \tilde{y}, h)}{h} = 0 \quad (2.26)$$

so heißt das explizite Einschrittverfahren konsistent auf G . Das Verfahren heißt konsistent von der Ordnung $p > 0$, falls es für jede kompakte Teilmenge $K \subset G$ eine Konstante $C > 0$ und ein $h' > 0$ gibt, sodass

$$|\tau(\tilde{t}, \tilde{y}, h)| \leq Ch^{p+1} \quad \forall (\tilde{t}, \tilde{y}) \in K \text{ und alle } h \in [0, h']. \quad (2.27)$$

Definition 2.25 (Explizite Runge–Kutta–Verfahren). Eine Inkrementfunktion $\Psi(t, y, h)$ gehört zu einem s –stufigen expliziten Runge–Kutta–Verfahren, falls sie für Zahlen a_{ij} , $b_i \in \mathbb{R}$, $c_i \in [0, 1]$ von der folgenden Form ist

$$\begin{aligned} \Psi(t, y, h) &= \sum_{i=1}^s b_i k_i \\ k_1 &= f(t, y) \\ k_2 &= f(t + c_2 h, y + h a_{21} k_1) \\ &\vdots \\ k_s &= f(t + c_s h, y + h(a_{s1} k_1 + \cdots + a_{s, s-1} k_{s-1})). \end{aligned} \quad (2.28)$$

Zudem müssen die Koeffizienten b_i die Konsistenzbedingung

$$\sum_{i=1}^s b_i = 1 \tag{2.29}$$

erfüllen.

Bemerkung 2.26. Runge–Kutta–Verfahren sind explizite Einschrittverfahren. Die maximal erreichbare Ordnung p eines s –stufigen Runge–Kutta–Verfahrens ist s .

Definition 2.27 (BDF–Verfahren). BDF–Verfahren (backward differentiation formulas) werden dadurch erhalten, dass durch die $k + 1$ Punkte (t_{i+1-j}, y_{i+1-j}) , $j = 0, \dots, k$, ein Polynom k –ten Grades gelegt wird und dann für den unbekanntes Wert y_{i+1} gefordert wird, dass die Differentialgleichung in t_{i+1} gilt, das heißt es wird

$$P_k(t) = \sum_{j=0}^k y_{i+1-j} L_j(t), \quad L_j(t) = \prod_{m=0, m \neq j}^k \frac{t - t_{i+1-m}}{t_{i+1-j} - t_{i+1-m}} \tag{2.30}$$

definiert und gefordert, dass

$$\dot{P}_k(t_{i+1}) = f(t_{i+1}, y_{i+1}) \tag{2.31}$$

gilt.

Bemerkung 2.28. BDF–Verfahren sind Mehrschrittverfahren.

2.1.4 Variationsrechnung

In diesem Abschnitt werden Resultate aus der Variationsrechnung (siehe [11]), die später verwendet werden, zitiert.

In der Variationsrechnung geht es um die Untersuchung von Extremalstellen von Variationsintegralen (oder Energie–Funktionalen)

$$u \mapsto \mathcal{F} := \int_{\Omega} F(x, u(x), Du(x)) dx: U \rightarrow \mathbb{R} \tag{2.32}$$

wobei $U \subset \{u: \Omega \subset \mathbb{R}^n \rightarrow \mathbb{R}^N\}$. $F(x, z, p)$ heißt Lagrange–Funktion.

Definition 2.29. Sei $\mathcal{F}: U \rightarrow \mathbb{R}$. Der „Punkt“ $u \in U$ heißt

- (globale) Minimalstelle von \mathcal{F} , falls $\mathcal{F}(u) \leq \mathcal{F}(v) \forall v \in U$
- strikte (globale) Minimalstelle von \mathcal{F} , falls $\mathcal{F}(u) < \mathcal{F}(v) \forall v \in U \setminus \{u\}$

- (strikte) lokale Minimalstelle von \mathcal{F} , falls U topologischer Raum ist und eine offene Umgebung $V \subset U$ von u existiert, so dass u (strikte) Minimalstelle von $\mathcal{F}|_V$ ist.

Der Wert $\mathcal{F}(u) \in \mathbb{R}$ heißt dann (striktes) Minimum. Die Definitionen für (strikte) Maximalstellen sind analog.

Es wird ein Funktional (sogenanntes Variationsfunktional) $\mathcal{F}: U \rightarrow \mathbb{R}$ mit $U \subset X$, X linearer Raum über \mathbb{R} betrachtet. Sei $u_0 \in U$, $\zeta \in X$, sodass $\{u_0 + \varepsilon\zeta: |\varepsilon| < \varepsilon_0\} \subset U$ für ein $\varepsilon_0 > 0$ gelte. Dann sei $\phi: \mathbb{R} \rightarrow \mathbb{R}$ definiert durch

$$\phi(\varepsilon) := \mathcal{F}(u_0 + \varepsilon\zeta), \quad |\varepsilon| < \varepsilon_0. \quad (2.33)$$

Definition 2.30 (Erste Variation). Falls $\phi'(0)$ existiert, heißt $\delta\mathcal{F}(u_0, \zeta) := \phi'(0)$ erste Variation von \mathcal{F} an u_0 in Richtung ζ .

Definition 2.31 (Stationärer Punkt). Sei $\mathcal{F}: U \subset X \rightarrow \mathbb{R}$. $u_0 \in U$ heißt stationärer Punkt von \mathcal{F} , falls

$$\delta\mathcal{F}(u_0, \zeta) = 0 \quad \forall \zeta \in X, \text{ für die } \delta\mathcal{F}(u_0, \zeta) \text{ existiert.} \quad (2.34)$$

Im Folgenden wird ein sogenanntes Variationsintegral betrachtet. Generelle Annahmen:

- $\Omega \subset \mathbb{R}^n$ offen, beschränkt; $u \in X := C^1(\overline{\Omega}, \mathbb{R}^N)$
- $1 - \text{Graph}(u) := \{(x, u(x), Du(x)): x \in \overline{\Omega}\} \dots$ kompakt
- $V \subset \mathbb{R}^n \times \mathbb{R}^N \times \mathbb{R}^{N \times n}$ offen, Lagrange-Funktion $F \in C^1(V, \mathbb{R})$
- $U = \{u \in X: 1 - \text{Graph}(u) \subset V\}$

Definition 2.32 (Variationsintegral). Sei $\mathcal{F}: U \rightarrow \mathbb{R}$ und

$$\mathcal{F}(u) := \int_{\Omega} F(x, u(x), Du(x)) dx, \quad (2.35)$$

dann heißt \mathcal{F} Variationsintegral.

Definition 2.33. Sei $C \subset U \subset X = C^1(\overline{\Omega}, \mathbb{R}^N)$. Dann heißt $u \in C$

- eine schwache (lokale) Minimalstelle von \mathcal{F} in C , falls $\exists \delta_0 > 0$ mit

$$\mathcal{F}(u) \leq \mathcal{F}(v) \quad \forall v \in C \cap B_{\delta_0}(u; X). \quad (2.36)$$

Dabei bezeichnet $B_{\delta_0}(u; X)$ den Kreis um u mit Radius δ_0 in der Norm von X .

- eine starke (lokale) Minimalstelle von \mathcal{F} in C , falls $\exists \delta_0 > 0$ mit

$$\mathcal{F}(u) \leq \mathcal{F}(v) \quad \forall v \in C \cap B_{\delta_0}(u; C(\overline{\Omega}, \mathbb{R}^N)). \quad (2.37)$$

Satz 2.34 (Euler–Lagrange Gleichung). *Zusätzlich zu den generellen Annahmen gelte $\frac{\partial F}{\partial p_{ij}} \in C^1(V)$. Für eine Funktion $u \in C^2(\Omega, \mathbb{R}^N) \cap C^1(\bar{\Omega}, \mathbb{R}^N)$ gelte*

$$\delta \mathcal{F}(u, \zeta) = 0 \quad \forall \zeta \in C_0^\infty(\Omega, \mathbb{R}^N). \quad (2.38)$$

Dann erfüllt u die Euler–Lagrange Gleichung von \mathcal{F} :

$$\frac{\partial F}{\partial z_i}(x, u(x), Du(x)) - \sum_{j=1}^n \frac{\partial}{\partial x_j} \left(\frac{\partial F}{\partial p_{ij}}(x, u(x), Du(x)) \right) = 0 \quad (2.39)$$

$$x \in \Omega, i = 1, \dots, N.$$

Im Folgenden wird ein sogenanntes Variationsproblem mit Nebenbedingung betrachtet, genauer: ein Variationsproblem mit holonomen Nebenbedingungen. Generelle Annahmen:

- $\Omega \subset \mathbb{R}^n$ offen, beschränkt; $u \in X := C^1(\bar{\Omega}, \mathbb{R}^N)$
- $V \subset \mathbb{R}^n \times \mathbb{R}^N \times \mathbb{R}^{N \times n}$ offen mit $1 - \text{Graph}(u) \subset V$
- Lagrange–Funktion $F \in C^2(V, \mathbb{R})$

Es wird das Problem

$$\mathcal{F}(v) := \int_{\Omega} F(x, v(x), Dv(x)) dx \rightarrow \min! \quad (2.40)$$

mit einer (holonomen) Nebenbedingung $0 = G(x, v(x)) = (G_1, \dots, G_r) \forall x \in \Omega$ (also $G = G(x, z)$) betrachtet. Für G gilt: $G \in C^2(\mathbb{R}^n \times \mathbb{R}^N, \mathbb{R}^r)$ mit $1 \leq r \leq N-1$, sodass für festes $x_0 \in \Omega$ $G(x_0, z) = 0$ eine $(N-r)$ –dimensionale Mannigfaltigkeit $M(x_0)$ von \mathbb{R}^N definiert. Das Problem $\mathcal{F}(v) := \int_{\Omega} F(x, v(x), Dv(x)) dx \rightarrow \min!$ ist ein Minimierungsproblem für Abbildungen $v: \bar{\Omega} \rightarrow M$.

Satz 2.35 (Euler–Lagrange Gleichung mit Nebenbedingung). *Sei $u \in C^2(\Omega, \mathbb{R}^N)$ schwache lokale Minimalstelle von \mathcal{F} in*

$$C_\delta := \{v \in X : \text{supp}(u - v) \subset \Omega, \|u - v\|_X < \delta, G(x, v(x)) = 0 \forall x \in \Omega\} \quad (2.41)$$

$\Rightarrow \exists$ eindeutig bestimmte Funktionen $\lambda_1, \dots, \lambda_r \in C(\Omega)$ (Lagrange–Multiplikatoren) mit

$$\frac{\partial}{\partial z_i} \left(F + \sum_{l=1}^r \lambda_l G_l \right) (x, u(x), Du(x)) - \sum_{j=1}^n \frac{\partial}{\partial x_j} \left(\frac{\partial F}{\partial p_{ij}}(x, u(x), Du(x)) \right) = 0, \quad (2.42)$$

$$x \in \Omega, i = 1, \dots, N.$$

2.2 Differentialgleichungen auf Mannigfaltigkeiten

Dieser Abschnitt ist aus [12] entnommen. Sei $M \subseteq \mathbb{R}^n$ eine gegebene Mannigfaltigkeit.

Ein System von gewöhnlichen Differentialgleichungen

$$\dot{y} = f(t, y), \quad y \in \mathbb{R}^n \quad \text{und} \quad y(0) = y_0 \quad (2.43)$$

mit $f: \mathbb{R}_0^+ \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ ist eine Differentialgleichung auf einer Mannigfaltigkeit M , wenn

$$y_0 \in M \quad \text{impliziert, dass} \quad y(t) \in M \quad \forall t. \quad (2.44)$$

Das ist äquivalent zur Forderung

$$f(t, y) \in T_y M \quad \text{für} \quad y \in M, \quad (2.45)$$

wobei $T_y M$ der Tangentialraum von M im Punkt $y \in M$ ist. Für m -dimensionale Mannigfaltigkeiten gegeben durch

$$M = \{y \in \mathbb{R}^n : h(y) = 0\} \quad (2.46)$$

mit einer Funktion $h: \mathbb{R}^n \rightarrow \mathbb{R}^{n-m}$ mit $m < n$ ist der Tangentialraum im Punkt $a \in M$ von der Form

$$T_a M = \left\{ v \in \mathbb{R}^n : \frac{\partial h}{\partial y}(a)v = 0 \right\} \quad (2.47)$$

und besitzt Dimension m .

Definition und Eigenschaften von DAEs

In diesem Kapitel werden grundlegende Begriffe und Fakten, die in Zusammenhang mit differential–algebraischen Gleichungssystemen und differentiellm Index stehen, beschrieben.

3.1 Grundlegende Definitionen zu DAEs und Index

Eine allgemeine Form eines differential–algebraischen Gleichungssystems ist in [13] zu finden.

Definition 3.1 (Differential–Algebraische Gleichung, DAE). Eine differential–algebraische Gleichung (DAE) ist gegeben durch die implizite Gleichung

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

mit einer Funktion $F: I \times D_x \times D_{\dot{x}} \rightarrow \mathbb{R}^n$, wobei $I \subseteq \mathbb{R}$ ein reelles Intervall ist und $D_x, D_{\dot{x}} \subseteq \mathbb{R}^n$ offene Mengen sind, $n \in \mathbb{N}$ und $x: I \rightarrow \mathbb{R}^n$ eine differenzierbare Funktion ist, wobei mit \dot{x} die Ableitung von x nach t bezeichnet wird.

Bemerkung 3.2. Als DAE wird im Allgemeinen auch ein differential–algebraisches Gleichungssystem bezeichnet.

Bemerkung 3.3. Ist die Matrix $\frac{\partial F}{\partial \dot{x}}$ regulär, dann kann mit dem Hauptsatz über implizite Funktionen $F = 0$ nach \dot{x} aufgelöst werden.

Bemerkung 3.4. Der Zustandsraum ist gegeben durch x .

Variablen bzw. Gleichungen werden mithilfe der folgenden Bezeichnung unterschieden:

- algebraische Variable: eine algebraische Variable zeichnet sich dadurch aus, dass in der DAE keine Ableitungen dieser vorkommen.

- differentielle Variable: eine differentielle Variable ist eine Variable deren Ableitungen ebenfalls in der DAE vorkommen.
- algebraische Gleichung: eine algebraische Gleichung einer DAE ist eine Gleichung, in der keine Ableitungen vorkommen.
- differentielle Gleichung: eine differentielle Gleichung einer DAE ist eine Gleichung, in der Ableitungen auftreten.

Definition 3.5 (Zwangsbedingung). Die algebraischen Gleichungen des differential-algebraischen Gleichungssystems besitzen die Gestalt $g(x) = 0$, mit einer Funktion $g: \mathbb{R}^n \rightarrow \mathbb{R}^m$ mit $m < n$, und werden Zwangsbedingung genannt.

Jedes differential-algebraische Gleichungssystem besitzt einen differentiellen Index, der in [14] definiert ist.

Definition 3.6 (Differentieller Index). Ein differential-algebraisches Gleichungssystem besitzt differentiellen Index $k \in \mathbb{N}_0$, wenn k die minimale Anzahl an Ableitungen darstellt, so dass sich aus dem System

$$F(t, x, \dot{x}) = 0, \frac{dF(t, x, \dot{x})}{dt} = 0, \dots, \frac{d^k F(t, x, \dot{x})}{dt^k} = 0 \quad (3.2)$$

ein gewöhnliches Differentialgleichungssystem extrahieren lässt, das durch algebraische Umformungen auf die Form $\dot{x} = \varphi(t, x)$ mit einer Funktion $\varphi: I \times D_x \rightarrow \mathbb{R}^n$ gebracht werden kann.

Bemerkung 3.7. Ein gewöhnliches Differentialgleichungssystem besitzt differentiellen Index Null.

Es gibt im Zusammenhang mit DAEs noch weitere Indexbegriffe wie beispielsweise „perturbation index“, „strangeness index“, „geometric index“, „tractability index“ oder „structural index“, siehe [13] und [15].

Im Folgenden wird der differentielle Index meist als Index bezeichnet.

3.2 Spezielle DAEs

Dieser Abschnitt ist aus [12] entnommen.

Es wird die Gestalt von bestimmten DAEs mit Index eins, zwei bzw. drei betrachtet.

- DAEs mit Index 1: Die DAE besitzt die Gestalt

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

wobei $f: U \times V \rightarrow \mathbb{R}^n$, $g: U \times V \rightarrow \mathbb{R}^m$ und $U \subset \mathbb{R}^n$, $V \subset \mathbb{R}^m$ offen. Die Variable y ist eine differentielle Variable und z eine algebraische Variable. Differentiation von $g(y, z) = 0$ nach t ergibt

$$g_y \dot{y} + g_z \dot{z} = g_y f + g_z \dot{z} = 0.\tag{3.4}$$

Wenn g_z invertierbar ist kann obige Gleichung umgeformt werden zu

$$\dot{z} = -(g_z)^{-1} g_y f.\tag{3.5}$$

Im Falle der Invertierbarkeit von g_z besitzt diese DAE Index 1.

- DAEs mit Index 2: Die DAE besitzt die Gestalt

$$\begin{aligned}\dot{y} &= f(y, z) \\ g(y) &= 0,\end{aligned}\tag{3.6}$$

wobei $f: U \times V \rightarrow \mathbb{R}^n$, $g: U \rightarrow \mathbb{R}^m$ und $U \subset \mathbb{R}^n$, $V \subset \mathbb{R}^m$ offen. Die Variable y ist eine differentielle Variable und z eine algebraische Variable. Differentiation von $g(y) = 0$ nach t ergibt

$$g_y \dot{y} = g_y f = 0.\tag{3.7}$$

Zweimaliges Ableiten von $g(y) = 0$ nach t ergibt

$$f^T \otimes g_{yy} \otimes f + g_y f_y \dot{y} + g_y f_z \dot{z} = f^T \otimes g_{yy} \otimes f + g_y f_y f + g_y f_z \dot{z} = 0,\tag{3.8}$$

wobei \otimes das Tensorprodukt bezeichnet. Wenn $g_y f_z$ invertierbar ist, kann obige Gleichung umgeformt werden zu

$$\dot{z} = -(g_y f_z)^{-1} (f^T \otimes g_{yy} \otimes f + g_y f_y f).\tag{3.9}$$

Im Falle der Invertierbarkeit von $g_y f_z$ besitzt diese DAE Index 2 und die DAE

$$\begin{aligned}\dot{y} &= f(y, z) \\ g_y f &= 0\end{aligned}\tag{3.10}$$

besitzt Index 1. Wenn $g_y(y) f_z(y, z)$ invertierbar ist, gilt mit dem Hauptsatz über implizite Funktionen (Satz 2.4), dass $g_y(y) f(y, z) = 0$ nach z aufgelöst werden kann, das heißt es existiert eine Funktion $h: \mathbb{R}^n \rightarrow \mathbb{R}^m$ mit $z = h(y)$. Zusammengefasst gilt: Wenn $g_y(y) f_z(y, z)$ invertierbar ist, besitzt die DAE (3.6) Index 2 und aus der ersten Ableitung der Zwangsbedingung $g(y) = 0$ kann die algebraische Variable z ausgedrückt werden.

- DAEs mit Index 3: Die DAE besitzt die Gestalt

$$\begin{aligned} \dot{y} &= f(y, z) \\ \dot{z} &= k(y, z, u) \\ g(y) &= 0, \end{aligned} \tag{3.11}$$

wobei $f: U \times V \rightarrow \mathbb{R}^n$, $k: U \times V \times W \rightarrow \mathbb{R}^m$, $g: U \rightarrow \mathbb{R}^p$ und $U \subset \mathbb{R}^n$, $V \subset \mathbb{R}^m$, $W \subset \mathbb{R}^p$ offen. Die Variablen y und z sind differentielle Variablen und u ist eine algebraische Variable. Differentiation von $g(y) = 0$ nach t ergibt

$$g_y \dot{y} = g_y f = 0. \tag{3.12}$$

Zweimaliges Ableiten von $g(y) = 0$ nach t ergibt

$$f^T \otimes g_{yy} \otimes f + g_y f_y \dot{y} + g_y f_z \dot{z} = f^T \otimes g_{yy} \otimes f + g_y f_y f + g_y f_z k = 0, \tag{3.13}$$

wobei \otimes das Tensorprodukt bezeichnet. Dreimaliges Ableiten von $g(y) = 0$ nach t ergibt einen Term mit $g_y f_z k_u \dot{u}$. Falls $g_y f_z k_u$ invertierbar ist, besitzt die DAE Index 3 und die DAE

$$\begin{aligned} \dot{y} &= f(y, z) \\ \dot{z} &= k(y, z, u) \\ f^T \otimes g_{yy} \otimes f + g_y f_y f + g_y f_z k &= 0 \end{aligned} \tag{3.14}$$

besitzt Index 1. Wenn $g_y(y) f_z(y, z) k_u(y, z, u)$ invertierbar ist, gilt mit Satz 2.4, dass $g_y(y) f_z(y, z) k(y, z, u)$ nach u aufgelöst werden kann, das heißt es existiert eine Funktion $h: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^p$ mit $u = h(y, z)$. Zusammengefasst gilt: Wenn $g_y(y) f_z(y, z) k_u(y, z, u)$ invertierbar ist, besitzt die DAE (3.11) Index 3 und aus der zweiten Ableitung der Zwangsbedingung $g(y) = 0$ kann die algebraische Variable u ausgedrückt werden.

Zusammenfassend gilt für DAEs mit Index 2 bzw. 3, dass unter gewissen Voraussetzungen die algebraischen Variablen aus der ersten bzw. zweiten Ableitung der Zwangsbedingung $g(y) = 0$ ausgedrückt werden können und in die differentielle Gleichungen eingesetzt werden können. Dieses Vorgehen wird bei einigen Methoden zur Indexreduktion in Kapitel 4 verwendet.

3.3 DAEs mit Index 1

Es wird die Gestalt einer DAE mit Index 1 aus Abschnitt 3.2 verwendet, also

$$\begin{aligned} \dot{y} &= f(y, z) \\ g(y, z) &= 0, \end{aligned} \tag{3.15}$$

wobei die Funktionen definiert sind wie oben. Für DAEs mit Index 1 werden im Folgenden zwei Möglichkeiten betrachtet, um diese zu lösen:

- Analytischer Weg:

Da die DAE Index 1 besitzt, muss die Ableitung von g nach z , das heißt $\frac{\partial g}{\partial z}$, regulär sein. Daher kann $g(y, z) = 0$ laut dem Hauptsatz über implizite Funktionen 2.4 nach z aufgelöst werden. Somit wird z als Funktion von y erhalten, also $z = \tilde{g}(y)$. Eingesetzt in die Gleichung $\dot{y} = f(y, z)$ ergibt das $\dot{y} = f(y, \tilde{g}(y))$. Mit dieser Vorgehensweise kann eine DAE mit Index 1 in eine gewöhnliche Differentialgleichung umgeformt werden.

- Numerischer Weg:

Wenn ein expliziter ode-Solver zum Lösen der Differentialgleichung $\dot{y} = f(y, z)$ verwendet wird, dann entspricht die Vorgehensweise dem folgenden Algorithmus. Ein Schritt von $(y_n, z_n) \mapsto (y_{n+1}, z_{n+1})$ folgt dem Schema:

Algorithmus 3.8 (Lösen einer Index 1 DAE mit einem expliziten ode-Solver).

- $y_{i+1} = \Phi_h(y_i, z_i)$, wobei Φ_h ein explizites numerisches Verfahren ist, das auf $\dot{y} = f(y, z)$ angewendet wird.
- Aus der Gleichung $g(y_{i+1}, z_{i+1}) = 0$ wird mithilfe eines Newton-Verfahrens z_{i+1} berechnet.

In der Gleichung $g(y_{i+1}, z_{i+1}) = 0$ besitzt y_{i+1} einen fixen Wert, das heißt g ist nur von z_{i+1} abhängig. Daher ist die Jacobimatrix von g gegeben durch $\frac{\partial g}{\partial z}$ und diese ist regulär. Ein Schritt des Newton-Verfahrens für die Gleichung $g(y_{i+1}, z_{i+1}) = 0$ hat die Form

$$z_{i+1,n+1} = z_{i+1,n} - \left(\frac{\partial g}{\partial z} \right)^{-1} (y_{i+1}, z_{i+1,n}) g(y_{i+1}, z_{i+1,n}), \quad (3.16)$$

wobei n der Laufindex des Newton-Verfahrens ist.

Wenn ein impliziter ode-Solver zum Lösen der Differentialgleichung $\dot{y} = f(y, z)$ verwendet wird, der nicht von z_{i+1} abhängt, dann entspricht die Vorgehensweise dem folgenden Algorithmus. Ein Schritt von $(y_n, z_n) \mapsto (y_{n+1}, z_{n+1})$ folgt dem Schema:

Algorithmus 3.9 (Lösen einer Index 1 DAE mit einem impliziten ode-Solver (Spezialfall)).

- $y_{i+1} = \Phi_h(y_i, y_{i+1}, z_i)$, wobei Φ_h ein implizites numerisches Verfahren ist, das auf $\dot{y} = f(y, z)$ angewendet wird.
- Aus der Gleichung $g(y_{i+1}, z_{i+1}) = 0$ wird mithilfe eines Newton-Verfahrens z_{i+1} berechnet.

Das Newtonverfahren für die Gleichung $g(y_{i+1}, z_{i+1}) = 0$ verläuft genauso wie im Fall eines expliziten ode-Solvers.

Wenn ein impliziter ode-Solver zum Lösen der Differentialgleichung $\dot{y} = f(y, z)$ verwendet wird, der von z_{i+1} abhängt, dann entspricht die Vorgehensweise dem folgenden Algorithmus. Ein Schritt von $(y_n, z_n) \mapsto (y_{n+1}, z_{n+1})$ folgt dem Schema:

Algorithmus 3.10 (Lösen einer Index 1 DAE mit einem impliziten ode-Solver).

- Aus den beiden Gleichungen $y_{i+1} = \Phi_h(y_i, y_{i+1}, z_i, z_{i+1})$, wobei Φ_h ein implizites numerisches Verfahren ist, das auf $\dot{y} = f(y, z)$ angewendet wird, und $g(y_{i+1}, z_{i+1}) = 0$ wird mithilfe eines Newton-Verfahrens y_{i+1} und z_{i+1} berechnet.

In diesem Fall ergibt sich mit

$$F(y_i, y_{i+1}, z_i, z_{i+1}) = \begin{pmatrix} y_{i+1} - \Phi_h(y_i, y_{i+1}, z_i, z_{i+1}) \\ g(y_{i+1}, z_{i+1}) \end{pmatrix} \quad (3.17)$$

für das Newtonverfahren

$$\begin{pmatrix} y_{i+1, n+1} \\ z_{i+1, n+1} \end{pmatrix} = \begin{pmatrix} y_{i+1, n} \\ z_{i+1, n} \end{pmatrix} - J^{-1}(y_{i+1, n}, z_{i+1, n}) \begin{pmatrix} y_{i+1, n} - \Phi_h(y_{i, n}, y_{i+1, n}, z_{i, n}, z_{i+1, n}) \\ g(y_{i+1, n}, z_{i+1, n}) \end{pmatrix}, \quad (3.18)$$

wobei J die Jacobimatrix von F ist, das heißt $J = \frac{\partial F}{\partial (y, z)}$.

Dieser Abschnitt ist der Grund dafür, dass bei den Regularisierungsmethoden in Kapitel 4 die Indexreduktion bis Index 1 durchgeführt wird und nicht bis Index 0.

3.4 DAEs bei mechanischen Systemen

Dieser Abschnitt ist aus [12] und [16] entnommen.

Herleitung der DAE mechanischer Gleichungssysteme mithilfe der Variationsrechnung (siehe 2.1.4):

Es bezeichnet q die generalisierten Koordinaten und \dot{q} die generalisierten Geschwindigkeiten. Weiters ist eine Funktion $L: \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$, die von t , q und \dot{q} abhängt, gegeben. Es wird das Variationsproblem

$$\int_{t_1}^{t_2} L(t, q(t), \dot{q}(t)) dt \rightarrow \min! \quad (3.19)$$

betrachtet. Die Euler-Lagrange Gleichungen berechnen sich laut Satz 2.34 zu

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = 0, \quad i = 1, \dots, n. \quad (3.20)$$

Für die Bewegung eines Systems in einem konservativen Kraftfeld, also ein System, bei dem sich die Gesamtenergie zufolge der Bewegung nicht ändert, zum Beispiel ein ungedämpftes Feder–Masse–System oder ein ungedämpftes Pendel (siehe [17]), berechnet sich L als

$$L = E_{kin} - E_{pot}, \quad (3.21)$$

wobei E_{kin} die kinetische und E_{pot} die potentielle Energie bezeichnet. Das Variationsproblem hat daher die Gestalt

$$\int_{t_1}^{t_2} (E_{kin} - E_{pot}) dt \rightarrow \min!. \quad (3.22)$$

Der nächste Schritt ist, eine oder mehrere Nebenbedingungen bzw. eine Zwangsbedingung mit in das System aufzunehmen. Das weitere Vorgehen kann ab diesem Schritt auf zwei verschiedenen Arten geschehen. Die eine Variante ist der Theorie der Variationsrechnung zufolge, die zweite ist der Lagrange–Formalismus der klassischen Mechanik.

- Variationsrechnung:

Es handelt sich hierbei um den Fall sogenannter holonomer Nebenbedingungen aus Abschnitt 2.1.4. Dabei berechnen sich die Euler–Lagrange Gleichungen zu

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial \left(L + \sum_{j=1}^m \lambda_j g_j \right)}{\partial q_i} = 0, \quad i = 1, \dots, n. \quad (3.23)$$

Zu diesen Gleichungen wird noch die Zwangsbedingung g_j , $j = 1, \dots, m$ hinzugefügt. Somit entsteht insgesamt ein Gleichungssystem mit $n + m$ Gleichungen. Diese Gleichungen können umgeformt werden zu

$$\frac{d}{dt} \left(\frac{\partial (E_{kin} - E_{pot})}{\partial \dot{q}_i} \right) - \frac{\partial \left((E_{kin} - E_{pot}) + \sum_{j=1}^m \lambda_j g_j \right)}{\partial q_i} = 0, \quad i = 1, \dots, n \quad (3.24)$$

und die Zwangsbedingung bildet die $(n + 1)$. bis $(n + m)$. Gleichung.

- Lagrange–Formalismus der klassischen Mechanik:

Um Nebenbedingungen mit aufzunehmen, wird die Zwangsbedingung $g_1(q) = 0, \dots, g_m(q) = 0$ in die Funktion L mit aufgenommen, das heißt

$$L = E_{kin} - E_{pot} - \lambda_1 g_1 - \dots - \lambda_m g_m, \quad (3.25)$$

wobei λ_k , $k = 1, \dots, m$ als Lagrangesche Multiplikatoren bezeichnet werden. Die Lagrangeschen Multiplikatoren λ_k werden zu den generalisierten Koordinaten und $\dot{\lambda}_k$ zu den generalisierten Geschwindigkeiten hinzugefügt, das heißt es gibt anstatt

vorher n generalisierte Koordinaten/Geschwindigkeiten $n+m$. Die Euler–Lagrange Gleichungen berechnen sich zu

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = 0, \quad i = 1, \dots, n+m. \quad (3.26)$$

Dieses Gleichungssystem besteht aus $n+m$ Gleichungen. Diese Gleichungen können umgeformt werden zu

$$\frac{d}{dt} \left(\frac{\partial \left(E_{kin} - E_{pot} - \sum_{j=1}^m \lambda_j g_j \right)}{\partial \dot{q}_i} \right) - \frac{\partial \left(E_{kin} - E_{pot} - \sum_{j=1}^m \lambda_j g_j \right)}{\partial q_i} = 0, \quad i = 1, \dots, n+m. \quad (3.27)$$

Da die Zwangsbedingung nicht von \dot{q} abhängt, ergeben sich die Gleichungen

$$\frac{d}{dt} \left(\frac{\partial (E_{kin} - E_{pot})}{\partial \dot{q}_i} \right) - \frac{\partial (E_{kin} - E_{pot} - \sum_{j=1}^m \lambda_j g_j)}{\partial q_i} = 0, \quad i = 1, \dots, n+m. \quad (3.28)$$

Werden aus q und \dot{q} die Lagrangeschen Multiplikatoren wieder entfernt und da $\frac{\partial(\lambda_j g_j)}{\partial \lambda_j} = g_j$ ergeben sich die Gleichungen

$$\frac{d}{dt} \left(\frac{\partial (E_{kin} - E_{pot})}{\partial \dot{q}_i} \right) - \frac{\partial (E_{kin} - E_{pot} - \sum_{j=1}^m \lambda_j g_j)}{\partial q_i} = 0, \quad i = 1, \dots, n \quad (3.29)$$

und die m Gleichungen der Zwangsbedingung. Das sind bis auf das Vorzeichen von den λ_j , $j = 1, \dots, m$ die selben $m+n$ Gleichungen, die auch beim Ansatz mit Variationsrechnung erhalten wurden, das heißt $\lambda_{mech} = -\lambda_{var}$.

Im Folgenden wird mit dem Vorzeichen, das beim Ansatz mithilfe der klassischen Mechanik erhalten wurde, weitergerechnet. Insgesamt führt dieses Vorgehen zum Gleichungssystem

$$\frac{d}{dt} \left(\frac{\partial (E_{kin} - E_{pot})}{\partial \dot{q}_i} \right) - \frac{\partial (E_{kin} - E_{pot} - \sum_{j=1}^m \lambda_j g_j)}{\partial q_i} = 0, \quad i = 1, \dots, n \quad (3.30)$$

$$g_j(q) = 0, \quad j = 1, \dots, m.$$

Durch $E_{kin} = E_{kin}(q, \dot{q})$, $E_{pot} = E_{pot}(q, \dot{q})$ und Verwendung der Linearität der Differentiation wird das Gleichungssystem

$$\frac{d}{dt} \left(\frac{\partial E_{kin}}{\partial \dot{q}_i} \right) - \frac{d}{dt} \left(\frac{\partial E_{pot}}{\partial \dot{q}_i} \right) - \frac{\partial E_{kin}}{\partial q_i} + \frac{\partial E_{pot}}{\partial q_i} + \frac{\partial \sum_{j=1}^m \lambda_j g_j}{\partial q_i} = 0, \quad i = 1, \dots, n$$

$$g_j(q) = 0, \quad j = 1, \dots, m \quad (3.31)$$

erhalten. Weitere Umformungen und $\frac{d}{dt} \left(\frac{\partial E_{kin}}{\partial \dot{q}_i} \right) =: \frac{d}{dt} (m(q_i) \dot{q}_i)$, $i = 1, \dots, n$ ergeben das Gleichungssystem

$$\begin{aligned} \frac{d}{dt} (m(q_i) \dot{q}_i) - \frac{d}{dt} \left(\frac{\partial E_{pot}}{\partial \dot{q}_i} \right) - \frac{\partial E_{kin}}{\partial q_i} + \frac{\partial E_{pot}}{\partial q_i} + \sum_{j=1}^m \lambda_j \frac{\partial g_j}{\partial q_i} &= 0, \quad i = 1, \dots, n \\ g_j(q) &= 0, \quad j = 1, \dots, m. \end{aligned} \quad (3.32)$$

Weitere Umformungen ergeben

$$\begin{aligned} m(q_i) \ddot{q}_i + \frac{d}{dt} (m(q_i)) \dot{q}_i - \frac{d}{dt} \left(\frac{\partial E_{pot}}{\partial \dot{q}_i} \right) - \frac{\partial E_{kin}}{\partial q_i} + \frac{\partial E_{pot}}{\partial q_i} + \sum_{j=1}^m \lambda_j \frac{\partial g_j}{\partial q_i} &= 0, \quad i = 1, \dots, n \\ g_j(q) &= 0, \quad j = 1, \dots, m. \end{aligned} \quad (3.33)$$

Aus diesen Gleichungen wird durch $M(q) := (E_{kin})_{\dot{q}\dot{q}}$ mit $(M(q))_{ii} = m(q_i)$, Umschreiben von $\left(\sum_{j=1}^m \lambda_j \frac{\partial g_j}{\partial q_i} \right)_i$ als Matrix-Vektor-Multiplikation $\left(\frac{\partial g}{\partial q} \right)^T \lambda$ und $-F_i(t, q, \dot{q}) = \frac{d}{dt} (m(q_i)) \dot{q}_i - \frac{d}{dt} \left(\frac{\partial E_{pot}}{\partial \dot{q}_i} \right) - \frac{\partial E_{kin}}{\partial q_i} + \frac{\partial E_{pot}}{\partial q_i}$ das Gleichungssystem

$$\begin{aligned} M(q) \ddot{q} - F(t, q, \dot{q}) + \left(\frac{\partial g}{\partial q} \right)^T \lambda &= 0 \\ g_j(q) &= 0, \quad j = 1, \dots, m \end{aligned} \quad (3.34)$$

erhalten. Somit wird durch Umformung des obigen Gleichungssystems die allgemeine Form der Gleichungssysteme mechanischer Systeme erhalten.

Definition 3.11 (Gleichungssystem mechanischer Systeme). Die Bewegungsgleichungen mechanischer Systeme sind gegeben durch

$$\begin{aligned} M(q) \ddot{q} &= F(t, q, \dot{q}) - G^T(q) \lambda \\ g(q) &= 0, \end{aligned} \quad (3.35)$$

wobei $q \in \mathbb{R}^n$ die generalisierten Koordinaten sind, $M \in \mathbb{R}^{n \times n}$ die symmetrisch positiv definite Massenmatrix, $F: \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ die Kräfte bezeichnet, $g: \mathbb{R}^n \rightarrow \mathbb{R}^m$ die Zwangsbedingung ist, $G = \frac{\partial g}{\partial q} \in \mathbb{R}^{m \times n}$ vollen Rang besitzt und $\lambda \in \mathbb{R}^m$ der Lagrangesche Multiplikator ist.

Im Folgenden wird die Gleichung $g(q) = 0$ aus dem differential-algebraischen Gleichungssystem (3.35) betrachtet. Zuerst wird der Fall $m = 1$, das heißt $g: \mathbb{R}^n \rightarrow \mathbb{R}$, diskutiert. Die Ableitung von g nach der Zeit t errechnet sich zu

$$\dot{g}(q) = G(q) \dot{q}, \quad (3.36)$$

wobei $G(q) \in \mathbb{R}^n$. Die zweite Ableitung von g nach t wird nach Differentiation der ersten Ableitung erhalten als

$$\ddot{g}(q) = \dot{q}^T G_q(q) \dot{q} + G(q) \ddot{q}, \quad (3.37)$$

wobei $G_q(q) \in \mathbb{R}^{n \times n}$. Da M symmetrisch und positiv definit ist, ist M regulär und daher invertierbar. Aus der Gleichung $M(q)\ddot{q} = F(t, q, \dot{q}) - G^T(q)\lambda$ des differentialalgebraischen Gleichungssystems (3.35) kann daher \ddot{q} ausgedrückt werden als

$$\ddot{q} = M(q)^{-1}(F(t, q, \dot{q}) - G^T(q)\lambda) \quad (3.38)$$

und Einsetzen in die Gleichung $\ddot{q}(q) = 0$ ergibt mit (3.37)

$$\dot{q}^T G_q(q)\dot{q} + G(q)M(q)^{-1}(F(t, q, \dot{q}) - G^T(q)\lambda) = 0. \quad (3.39)$$

Umformungen ergeben

$$\dot{q}^T G_q(q)\dot{q} + G(q)M(q)^{-1}F(t, q, \dot{q}) = G(q)M(q)^{-1}G^T(q)\lambda. \quad (3.40)$$

Die reelle Zahl $G(q)M(q)^{-1}G^T(q)$ ist ungleich 0, da $M(q)^{-1}$ eine symmetrisch positiv definite Matrix ist, daher n positive Eigenwerte besitzt, die Eigenvektoren eine Basis des \mathbb{R}^n bilden (V ist die Matrix mit den Eigenvektoren in den Spalten, V orthogonal) und dadurch $G(q)M(q)^{-1}G^T(q) = H(q)^T D H(q) = \sum_{i=1}^n \lambda_i (h_i)^2 > 0$ mit $V^T G^T = H$, wobei D die Diagonalmatrix mit den Eigenwerten in der Diagonale ist und h_i die Spalten von H bezeichnet. Daher kann mit $(G(q)M(q)^{-1}G^T(q))^{-1}$ multipliziert werden. Dies führt auf

$$(G(q)M(q)^{-1}G^T(q))^{-1}(\dot{q}^T G_q(q)\dot{q} + G(q)M(q)^{-1}F(t, q, \dot{q})) = \lambda. \quad (3.41)$$

Es ist bemerkenswert, dass aus der zweiten Ableitung der Zwangsbedingung $\ddot{q}(q) = 0$ der Lagrangesche Multiplikator λ ausgedrückt werden kann. Mit Gleichung (3.41) ist weiter zu erkennen, dass die DAE (3.35) Index 3 besitzt, da durch nochmaliges Ableiten eine Gleichung für $\dot{\lambda}$ entsteht. Weiters kann der in Gleichung (3.41) ausgedrückte Lagrangesche Multiplikator in die differentiellen Gleichungen der DAE (3.35) eingesetzt werden. Diese Gleichungen werden das zugrundeliegende gewöhnliche Differentialgleichungssystem genannt.

Nun wird die obige Herleitung verallgemeinert auf Funktionen g , die in den \mathbb{R}^m abbilden, das heißt $g: \mathbb{R}^n \rightarrow \mathbb{R}^m$. Die erste Ableitung von g nach t errechnet sich genauso wie im reellwertigen Fall. Daher

$$\dot{g}(q) = G(q)\dot{q}, \quad (3.42)$$

wobei $G(q) \in \mathbb{R}^{m \times n}$. Die zweite Ableitung von g nach t wird nach Differentiation der ersten Ableitung erhalten als

$$\ddot{g}(q) = \dot{q}^T \otimes G_q(q) \otimes \dot{q} + G(q)\ddot{q}, \quad (3.43)$$

wobei $G_q(q)$ ein Tensor 3. Stufe ist und \otimes das Tensorprodukt bezeichnet. Das Tensorprodukt $G_q(q) \otimes \dot{q}$ ist das Produkt eines Tensors 3. Stufe und eines Tensors 1. Stufe, somit ist $G_q(q) \otimes \dot{q}$ ein Tensor 4. Stufe. Durch Verjüngung wird daraus ein Tensor 2. Stufe.

Bildet man das Produkt des Tensors 1. Stufe \dot{q}^T mit dem Tensor 2. Stufe $G_q(q) \otimes \dot{q}$ so ist das Produkt ein Tensor 3. Stufe. Durch Verjüngung wird daraus ein Tensor 1. Stufe. Insgesamt ist daher $\dot{q}^T \otimes G_q(q) \otimes \dot{q}$ ein Tensor 1. Stufe, das heißt ein Vektor. Analoge Umformungen zum reellwertigen Fall ergeben die Gleichung

$$\dot{q}^T \otimes G_q(q) \otimes \dot{q} + G(q)M(q)^{-1}F(t, q, \dot{q}) = G(q)M(q)^{-1}G^T(q)\lambda. \quad (3.44)$$

Die Matrix $G(q)M(q)^{-1}G^T(q) \in \mathbb{R}^{m \times m}$ ist regulär, da $M(q)^{-1}$ eine symmetrisch positiv definite Matrix ist und $S := G^T \in \mathbb{R}^{n \times m}$ vollen Rang m besitzt, gilt laut Satz 2.17, dass $G(q)M(q)^{-1}G^T(q)$ SPD ist und daher auch regulär ist. Da $G(q)M(q)^{-1}G^T(q)$ eine reguläre Matrix ist, ergibt sich

$$(G(q)M(q)^{-1}G^T(q))^{-1} (\dot{q}^T \otimes G_q(q) \otimes \dot{q} + G(q)M(q)^{-1}F(t, q, \dot{q})) = \lambda. \quad (3.45)$$

Wie im reellwertigen Fall besitzt die DAE Index 3 und aus der zweiten Ableitung der Zwangsbedingung g kann der Lagrangesche Multiplikator λ ausgedrückt werden. Analog zu oben kann der so ausgedrückte Lagrangesche Multiplikator in die differentiellen Gleichungen eingesetzt werden.

Das System

$$\begin{aligned} M(q)\ddot{q} &= F(t, q, \dot{q}) - G^T(q)\lambda \\ g(q) &= 0, \end{aligned} \quad (3.46)$$

kann durch

$$\dot{q} = p \quad (3.47)$$

auf die Gestalt

$$\begin{aligned} \dot{q} &= p \\ M(q)\dot{p} &= F(t, q, p) - G^T(q)\lambda \\ g(q) &= 0, \end{aligned} \quad (3.48)$$

gebracht werden, also auf die Gestalt einer DAE mit Index 3 wie in Abschnitt 3.2 mit $y = q$, $z = p$, $u = \lambda$, $f(y, z) = z$ und $k(y, z, u) = M^{-1}(F - G^T u)$.

Regularisierungsmethoden

In diesem Kapitel werden die verschiedenen Regularisierungsmethoden klassifiziert und erklärt.

4.1 Überblick

Es gibt zahlreiche Methoden der Indexreduktion. In diesem Abschnitt werden diese Methoden aufgelistet und in Klassen eingeteilt. In der Literatur werden auch immer wieder spezielle Methoden, beispielsweise Indexreduktion ausschließlich für lineare DAEs oder Gleichungssysteme mechanischer Systeme mit Index 3, ausführlich diskutiert, siehe [13] und [16]. In dieser Arbeit werden eher allgemeine Ansätze diskutiert, um eine möglichst breite Anwendung dieser gewährleisten zu können. Zum Vergleich wird eine Methode, die speziell für die ausgewählten Fallbeispiele konzipiert ist, betrachtet. Die gängigsten Methoden zur Indexreduktion lassen sich im Allgemeinen in folgende Bereiche teilen:

- Regularisierung durch Differentiation
Bei diesen Methoden wird in der Regel mit einer oder mehreren Ableitungen der Zwangsbedingungen gearbeitet.
- Regularisierung durch Projektionsmethoden
Bei Projektionsmethoden wird die numerische Lösung, wenn sie die Zwangsbedingungen nicht mehr erfüllt, auf die Mannigfaltigkeit, die durch diese gegeben ist, projiziert.
- Regularisierung durch Zustandsraumtransformation
Bei Methoden mit Koordinatentransformation wird der Zustandsraum (lokal, wenn möglich global) umkoordinatisiert und dann dieses System gelöst.

Diese weit verbreiteten Methoden werden in den folgenden Abschnitten im Detail diskutiert. In Kapitel 5 und 6 wird anhand von Beispielen die genaue Vorgehensweise präsen-

tiert und erklärt. Im Weiteren werden noch ein paar Ansätze aus der Literatur erwähnt, die aber nicht näher betrachtet werden:

- DAETS (Differential Algebraic Equations by Taylor Series) ist ein Verfahren zum Lösen von DAEs mittels Taylorreihenentwicklung der Funktionen und mithilfe derer wird dann die Integration vorgenommen. Anschließend wird die Lösung, wenn sie die Zwangsbedingung nicht mehr erfüllt, auf die Lösungsmannigfaltigkeit projiziert (siehe [18]). Diese Methode wird nicht näher betrachtet, da die Indexreduktion durch Projektion erfolgt und dies im Abschnitt 4.3 behandelt wird.
- Das Simulationstool APMonitor kann zum Lösen von DAEs herangezogen werden. DAEs werden gelöst, indem die differentiellen Gleichungen in algebraische umgewandelt werden (siehe [19]), wobei auch ein Optimierungstool herangezogen wird. Dieses Tool wird nicht näher betrachtet, da es keine eigenständige Methode repräsentiert.
- Die Grundidee der Linear Graph Theory (siehe [20]) ist, jede Komponente des Gesamtsystems einzeln zu modellieren. Somit können für verschiedene Elemente des mechanischen Systems verschiedene Koordinatisierungen gewählt werden. Dieser Ansatz wird nicht näher betrachtet, da die Indexreduktion durch Differentiation der Zwangsbedingung und durch lokale Koordinatisierungen durchgeführt wird und dies in Abschnitt 4.2 und 4.4 behandelt wird.
- DynaFlexPro (siehe [21]) ist ein Tool von Maple, das zur Formulierung der Gleichungen mechanischer Systeme verwendet werden kann. DynaFlexPro basiert auf der Linear Graph Theory. Die Koordinaten der Gleichungen werden so gewählt, dass die Gleichungen eine möglichst einfache Gestalt besitzen. Dieses Tool wird nicht näher betrachtet, da es keine eigenständige Methode repräsentiert.

4.2 Indexreduktion durch Differentiation

Im Folgenden werden drei Methoden betrachtet, die die Indexreduktion mittels Differentiation durchführen.

4.2.1 Differentiation der Zwangsbedingung

Ein erster Ansatz zur Reduktion des Index, der in [14] zu finden ist, besteht darin, die Zwangsbedingung $g(x) = 0$ zu differenzieren und sie durch ihre Ableitung zu ersetzen. Dieses Vorgehen wird solange durchgeführt, bis das System differentiellen Index 1 besitzt, siehe Abschnitt 3.3. Ein Problem, das sich durch dieses Vorgehen ergibt, ist die Tatsache, dass durch das Ableiten Information verloren geht und daher die für die Rückintegration

notwendigen Anfangswerte unbekannt sind. Dadurch kann es zum sogenannten numerischen „Drift-off“ kommen, bei dem sich die numerisch berechnete Lösung von der exakten Lösung entfernt. Dieses Phänomen ist in den Kapiteln 5 und 6 zu beobachten.

4.2.2 Baumgarte–Methode

Ein etwas anderer Ansatz ist die Baumgarte–Methode (siehe [16]). Als Ausgangspunkt wird die Index–1–Formulierung eines Index–3–Systems verwendet und die algebraischen Variablen werden aus der zweiten Ableitung der Zwangsbedingung $g(x) = 0$ ausgedrückt und in die differentiellen Gleichungen eingesetzt (siehe Abschnitt 3.2). Diese Methode ersetzt die Zwangsbedingung $\ddot{g}(x) = 0$ der Index–1–Formulierung durch eine Linearkombination von \ddot{g} , \dot{g} und g , das heißt statt $\ddot{g}(x) = 0$ wird folgende Gleichung

$$\ddot{g} + 2\alpha\dot{g} + \beta^2g = 0 \quad (4.1)$$

verwendet. Die Methode wird dadurch motiviert, dass durch das Auftreten der ursprünglichen Zwangsbedingung in Gleichung (4.1) keine Information verloren geht. Die Konstanten α und β in Gleichung (4.1) sollen so gewählt werden, dass die Differentialgleichung asymptotisch stabil ist, das heißt die Eigenwerte des charakteristischen Polynoms müssen einen Realteil besitzen, der strikt kleiner Null ist. Das charakteristische Polynom errechnet sich zu

$$\lambda^2 + 2\alpha\lambda + \beta^2 \quad (4.2)$$

und daraus ergibt sich

$$\lambda_{1,2} = -\alpha \pm \sqrt{\alpha^2 - \beta^2}. \quad (4.3)$$

Daher muss $\alpha > 0$ sein. Ein Problem dieser Methode stellt die Wahl der beiden Konstanten dar, da es außer der Voraussetzung $\alpha > 0$ keine genaueren Spezifikationen gibt. Eine genauere Untersuchung des Parameters α ist in [22] zu finden.

Je größer α und β betragsmäßig (im Fall $\alpha \leq \beta$ ist der Realteil von $\lambda_{1,2}$ gleich $-\alpha$) werden, desto größer werden $\lambda_{1,2}$ betragsmäßig. Das heißt das Differentialgleichungssystem (4.1) wird steifer, je größer α und β betragsmäßig werden.

4.2.3 Pantelides–Algorithmus

Der folgende Abschnitt ist aus [23] entnommen.

Es sei eine DAE gegeben, dann ist die Vorgehensweise des Pantelides–Algorithmus für jede Zwangsbedingung die folgende.

Algorithmus 4.1 (Pantelides–Algorithmus).

1. Sobald eine Zwangsbedingung in einer DAE gefunden wird, muss diese differenziert werden.

2. Diese differenzierte Zwangsbedingung wird zum vorhandenen Gleichungssystem hinzugefügt. Kommt in der Zwangsbedingung eine algebraische Variable vor, so ist das Differential dieser ein sogenanntes Dummy Derivative. Das bedeutet, wenn y eine algebraische Variable ist, die in der Zwangsbedingung vorkommt, so wird die Ableitung mit dy bezeichnet anstatt mit \dot{y} . Die Variable dy wird als Dummy Derivative bezeichnet.
3. In diesem Schritt wird ein Integrator einer differentiellen Variable, der mit der Zwangsbedingung bzw. genauer mit der Ableitung der Zwangsbedingung in Zusammenhang steht, aufgelöst. Bezeichnet x eine differentielle Variable, deren Ableitung \dot{x} in der Ableitung der Zwangsbedingung vorkommt, so kann der Integrator $\dot{x} \xrightarrow{f} x$ aufgebrochen werden und somit die Verbindung von \dot{x} und x eliminiert werden. Das bedeutet, dass anstatt von $\dot{x} dx$ im Weiteren verwendet wird.
4. Beim Ableiten der Zwangsbedingung kann es passieren, dass neue Variablen erzeugt werden, beispielsweise wird aus der in der Zwangsbedingung vorkommenden algebraischen Variable y durch Differenzieren dy (siehe Schritt 2). Die Variable y muss auch in einer anderen Gleichung vorkommen (sonst wäre die Zwangsbedingung keine Zwangsbedingung, da y unbekannt wäre). Aus dieser Gleichung kann y berechnet werden.
5. Daher muss diese Gleichung ebenfalls differenziert und zum Gleichungssystem hinzugefügt werden.
6. Die Vorgehensweise der Punkte 4–5 wird so lange fortgeführt bis durch das Ableiten keine neue Variable entsteht.

Bemerkung 4.2. Im Zuge des Pantelides–Algorithmus taucht die Frage auf, wie Zwangsbedingungen gefunden werden können. Dies passiert durch andere Algorithmen, zum Beispiel im Rahmen der Gleichungssortierung, siehe [24].

Ein Nachteil des Pantelides–Algorithmus ist, dass im Zuge dieses Verfahrens schnell sehr viele Gleichungen und Unbekannte erzeugt werden und dadurch das Gleichungssystem unübersichtlich wird.

4.3 Stabilisierung durch Projektion

Diese Methode ist aus [12] und [14] entnommen.

Es sei eine DAE mit differentiellem Index $k > 1$ gegeben. Bei diesem Ansatz wird die numerische Lösung, wenn sie die Zwangsbedingung nicht mehr erfüllt, das heißt von der tatsächlichen Lösung „wegdriftet“, auf die durch die Zwangsbedingung $g(x) = 0$ und

deren $1, \dots, (k-2)$. Ableitungen nach t implizit definierte Mannigfaltigkeit projiziert. Diese sogenannte Lösungsmannigfaltigkeit ist daher gegeben durch

$$M = \left\{ x \in \mathbb{R}^n : g(x) = 0, \frac{d^i g(x)}{dt^i} = 0 \text{ mit } i = 1, \dots, k-2 \right\}. \quad (4.4)$$

Die DAE hat eine Gestalt derart, dass sich aus der $(k-1)$. Ableitung der Zwangsbedingung die algebraischen Variablen ausdrücken lassen und diese in die Differentialgleichungen eingesetzt werden können. Diese Differentialgleichungen ohne algebraische Variablen werden dann in $f(t, y)$ zusammengefasst und stellen somit die Index-1-Formulierung dar. Im Folgenden wird die Differentialgleichung $\dot{y} = f(t, y)$ auf einer Mannigfaltigkeit M betrachtet, siehe Abschnitt 2.2.

Zunächst wird die Gestalt der Mannigfaltigkeit M speziell für DAEs mit Index 2 und 3 wie in Abschnitt 3.2 beschrieben.

- DAEs mit Index 2: Die Mannigfaltigkeit wird implizit definiert durch $g(x) = 0$, das heißt

$$M = \{x \in \mathbb{R}^n : g(x) = 0\}. \quad (4.5)$$

Die algebraischen Variablen z können aus der ersten Ableitung von $g(x) = 0$, also $g_x f = 0$, ausgedrückt werden, falls $g_x f_z$ invertierbar ist.

- DAEs mit Index 3: Die Mannigfaltigkeit wird implizit definiert durch $g(x) = 0$ und $g_x f = 0$, das heißt

$$\begin{aligned} M &= \left\{ (x, z) \in \mathbb{R}^{n+m} : g(x) = 0, \frac{\partial g(x)}{\partial t} = 0 \right\} = \\ &= \left\{ (x, z) \in \mathbb{R}^{n+m} : g(x) = 0, g_x(x) f(x, z) = 0 \right\}. \end{aligned} \quad (4.6)$$

Die algebraischen Variablen u können aus der zweiten Ableitung von $g(x) = 0$, also $f^T \otimes g_{xx} \otimes f + g_x f_x f + g_x f_z k = 0$, ausgedrückt werden, falls $g_x f_z k_u$ invertierbar ist.

Projektionsmethoden sind ein Standardverfahren zur numerischen Lösung von Differentialgleichungen auf Mannigfaltigkeiten. In dieser Arbeit wird die sogenannte Standardprojektionsmethode und die symmetrische Projektionsmethode betrachtet.

4.3.1 Standardprojektionsmethode

Bei der Standardprojektionsmethode wird nach jedem Lösungsschritt orthogonal auf die Lösungsmannigfaltigkeit projiziert. Ein Schritt $y_n \mapsto y_{n+1}$ der Standardprojektionsmethode folgt folgendem Schema:

Algorithmus 4.3 (Standardprojektionsmethode).

- Es wird $\hat{y}_{n+1} = \Phi_h(y_n)$, wobei Φ_h einen numerischen Integrator angewandt auf $\dot{y} = f(t, y)$ bezeichnet (beispielsweise ein Runge–Kutta–Verfahren), berechnet.
- Um $y_{n+1} \in M$ zu erhalten, wird \hat{y}_{n+1} orthogonal auf die Mannigfaltigkeit M projiziert.

In Abbildung 4.1 ist die Standardprojektionsmethode schematisch dargestellt.

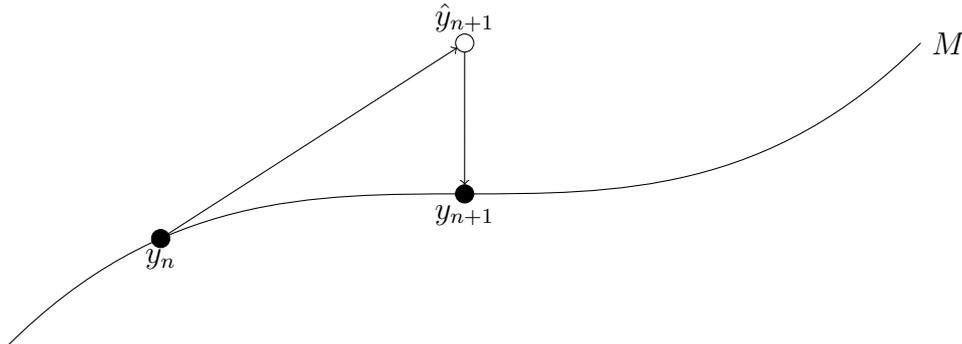


Abbildung 4.1: Illustration der Standardprojektionsmethode (Quelle: [12])

Die Standardprojektionsmethode wird auch orthogonale Projektionsmethode genannt.

4.3.2 Symmetrische Projektionsmethode

Die Idee der symmetrischen Projektionsmethode ist, den Wert y_n zu stören und anschließend ein symmetrisches Einschrittverfahren auf diesen „gestörten“ Wert anzuwenden, sodass der nächste Wert y_{n+1} entsprechend der betragsmäßigen Störung von y_n von der Mannigfaltigkeit entfernt ist.

Definition 4.4 (Symmetrisches Einschrittverfahren). Ein Einschrittverfahren Φ_h ist symmetrisch, wenn $\Phi_h = \Phi_{-h}^{-1}$.

Ein Schritt der symmetrischen Projektionsmethode $y_n \mapsto y_{n+1}$ folgt folgendem Schema:

Algorithmus 4.5 (Symmetrische Projektionsmethode).

- $\hat{y}_n = y_n + \frac{\partial g}{\partial y}^T(y_n)\mu$ mit $g(y_n) = 0$.
- Es wird $\hat{y}_{n+1} = \Phi_h(\hat{y}_n)$, wobei Φ_h ein symmetrisches Einschrittverfahren angewandt auf $\dot{y} = f(t, y)$ bezeichnet, berechnet.
- $y_{n+1} = \hat{y}_{n+1} + \frac{\partial g}{\partial y}^T(y_{n+1})\mu$ mit μ so, dass $g(y_{n+1}) = 0$.

Es ist wichtig, dass bei einem Schritt von $y_n \rightarrow y_{n+1}$ dasselbe μ verwendet wird.

Die Vorgehensweise zur Berechnung, wie in [25] beschrieben, für einen Schritt $y_n \rightarrow y_{n+1}$ lautet wie folgt:

1. Für y_{n+1} und μ werden Schätzwerte angenommen. Weiters wird angenommen, dass $\frac{\partial g^T}{\partial y}(y_{n+1}) \approx \frac{\partial g^T}{\partial y}(y_n)$ ist.
2. Mit diesen Schätzungen wird aus

$$\hat{y}_n = y_n + \frac{\partial g^T}{\partial y}(y_n)\mu \quad (4.7)$$

\hat{y}_n und aus

$$\hat{y}_{n+1} = y_{n+1} - \frac{\partial g^T}{\partial y}(y_{n+1})\mu \approx y_{n+1} - \frac{\partial g^T}{\partial y}(y_n)\mu \quad (4.8)$$

\hat{y}_{n+1} berechnet.

3. Die Abweichungen von den Schätzwerten werden mit Δy_{n+1} und $\Delta \mu$ bezeichnet, sie berechnen sich durch folgendes Gleichungssystem

$$\begin{pmatrix} I & -2\frac{\partial g^T}{\partial y}(y_n) \\ \frac{\partial g}{\partial y}(y_n) & 0 \end{pmatrix} \begin{pmatrix} \Delta y_{n+1} \\ \Delta \mu \end{pmatrix} = - \begin{pmatrix} d \\ g(y_{n+1}) \end{pmatrix}, \quad (4.9)$$

wobei $d := \hat{y}_n + h\Psi(h, \hat{y}_n, \hat{y}_{n+1}) - \hat{y}_{n+1}$ und Ψ die Inkrementfunktion des numerischen Verfahrens Φ_h ist.

4. Die so erhaltenen neuen Werte für y_{n+1} und μ sind $y_{n+1} + \Delta y_{n+1}$ und $\mu + \Delta \mu$.

Das gesamte Vorgehen wird so lange wiederholt, bis die Fehler von y_{n+1} und μ kleiner als eine vorgegebene Toleranz sind.

Durch die rechte Seite des Gleichungssystems (4.9) wird erreicht, dass der Fehler des Einschrittverfahrens d verschwindend klein ist und die Zwangsbedingung $g(y_{n+1}) = 0$ erfüllt wird. Gleichung (4.9) lässt sich über das Newton-Verfahren

$$\begin{pmatrix} \tilde{y}_{n+1} \\ \tilde{\mu} \end{pmatrix} = \begin{pmatrix} y_{n+1} \\ \mu \end{pmatrix} - J(y_{n+1}, \mu)^{-1}k(y_{n+1}, \mu) \quad (4.10)$$

motivieren, wobei $J(y_{n+1}, \mu) = \begin{pmatrix} \frac{\partial k}{\partial y, \partial z}(y_{n+1}, \mu) \end{pmatrix}$. Umformungen von (4.10) ergeben mit $\tilde{y}_{n+1} - y_{n+1} = \Delta y_{n+1}$ und $\tilde{\mu} - \mu = \Delta \mu$

$$J(y_{n+1}, \mu) \begin{pmatrix} \Delta y_{n+1} \\ \Delta \mu \end{pmatrix} = -k(y_{n+1}, \mu). \quad (4.11)$$

Gleichung (4.7) von Gleichung (4.8) subtrahiert ergibt unter Verwendung von $\hat{y}_{n+1} = \hat{y}_n + h\Psi(h, \hat{y}_n, \hat{y}_{n+1})$

$$h\Psi(h, \hat{y}_n, \hat{y}_{n+1}) = y_{n+1} - y_n - 2\frac{\partial g^T}{\partial y}(y_n)\mu. \quad (4.12)$$

Somit ergibt sich

$$d = \hat{y}_n - \hat{y}_{n+1} + y_{n+1} - y_n - 2\frac{\partial g^T}{\partial y}(y_n)\mu. \quad (4.13)$$

Da y_{n+1} und μ berechnet werden sollen, wird d als Funktion ausgewertet an y_{n+1} und μ aufgefasst, also

$$d(y, z) = \hat{y}_n - \hat{y}_{n+1} + y - y_n - 2\frac{\partial g^T}{\partial y}(y_n)z. \quad (4.14)$$

Daher ergibt sich für die Ableitungen

$$\frac{\partial d}{\partial y}(y_{n+1}, \mu) = I \quad \text{und} \quad \frac{\partial d}{\partial z}(y_{n+1}, \mu) = -2\frac{\partial g^T}{\partial y}(y_n). \quad (4.15)$$

Für die zweite Zeile von (4.9) gilt

$$\frac{\partial g}{\partial y}(y_{n+1}) \approx \frac{\partial g}{\partial y}(y_n) \quad \text{und} \quad \frac{\partial g}{\partial z}(y_{n+1}) = 0. \quad (4.16)$$

Somit wird das Gleichungssystem (4.9) erreicht.

In Abbildung 4.2 ist die symmetrische Projektionsmethode schematisch dargestellt.

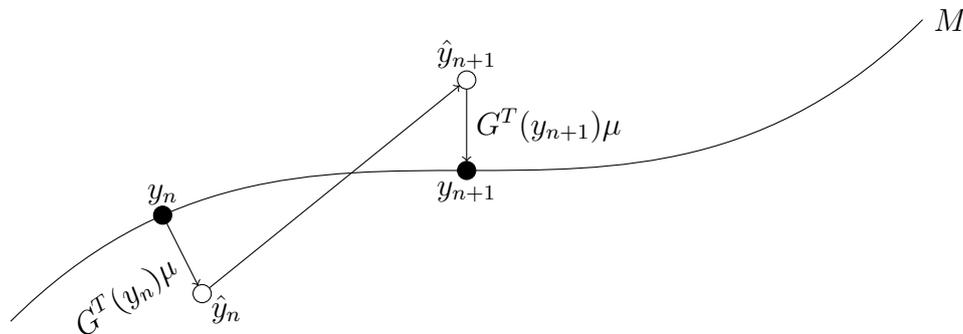


Abbildung 4.2: Illustration der symmetrischen Projektionsmethode (Quelle: [12])

4.4 Methoden basierend auf lokaler Zustandsraumtransformation

Diese Methode ist aus [12] und [14] entnommen.

Es sei eine DAE mit differentiellem Index $k > 1$ gegeben. Bei dieser Methode wird die

Lösung der DAE anstatt auf dem gesamten Zustandsraum nur auf einer Mannigfaltigkeit gesucht, die implizit definiert ist durch die Zwangsbedingung $g(x) = 0$ und deren $1, \dots, (k-2)$. Ableitungen nach t , siehe Abschnitt 2.2. Die DAE hat eine Gestalt derart, dass sich aus der $(k-1)$. Ableitung der Zwangsbedingung die algebraischen Variablen ausdrücken lassen und diese in die Differentialgleichungen eingesetzt werden können. Diese Differentialgleichungen ohne algebraische Variablen werden dann in $f(t, y)$ zusammengefasst und stellen somit die Index-1-Formulierung dar. Dadurch vereinfacht sich das Lösen einer DAE zum Lösen einer ODE auf der Mannigfaltigkeit (4.4), die durch Einführung lokaler Koordinatentransformationen leichter gelöst werden kann. Der Tangentialraum ist für ein fixes $y \in M$ von der Form

$$T_y M = \{v \in \mathbb{R}^n : tg_i(y, v) = 0 \text{ mit } i = 1, \dots, k-1\} \quad (4.17)$$

mit

$$tg_i(y, \dot{y}) = \left(\frac{d^{i-1}g}{dt^{i-1}} \right)_t, \quad (4.18)$$

wobei $\left(\frac{d^{i-1}g}{dt^{i-1}} \right) = \left(\frac{d^{i-1}g}{dt^{i-1}} \right)(y)$, das heißt nur von y und nicht von \dot{y} abhängt und t im Index für die Zeitableitung steht.

Im Folgenden wird die Gestalt des Tangentialraumes $T_y M$ speziell für DAEs mit Index 2 und 3 wie in Abschnitt 3.2 beschrieben.

- DAEs mit Index 2: Die Mannigfaltigkeit ist in diesem Fall gegeben durch $M = \{x \in \mathbb{R}^n : g(x) = 0\}$. Der Tangentialraum wird dann definiert durch

$$T_y M = \{v \in \mathbb{R}^n : tg_1(y, v) = 0\} = \{v \in \mathbb{R}^n : g_x(y)v = 0\}. \quad (4.19)$$

- DAEs mit Index 3: Die Mannigfaltigkeit ist in diesem Fall gegeben durch $M = \{(x, z) \in \mathbb{R}^{n+m} : g(x) = 0, g_x f = 0\}$. Der Tangentialraum wird dann definiert durch

$$\begin{aligned} T_y M &= \{v \in \mathbb{R}^n : tg_1(y, v) = 0, tg_2(y, v) = 0\} = \\ &= \{(v, w) \in \mathbb{R}^n : g_x(y)v = 0, f^T \otimes g_{xx}(y)v + g_x(y)(f_x v + f_z w) = 0\}. \end{aligned} \quad (4.20)$$

Angenommen

$$\psi : U \rightarrow V, \quad U \subset \mathbb{R}^m \text{ offen, } V \subseteq M \subset \mathbb{R}^n \quad (4.21)$$

ist eine lokale Koordinatisierung (hinreichend glatt, bijektiv und $\frac{\partial \psi(z)}{\partial z}$ hat vollen Rang m) auf der m -dimensionalen Mannigfaltigkeit M . Die Koordinatentransformation $y = \psi(z)$ transformiert die Differentialgleichung $\dot{y} = f(t, y)$ in

$$\frac{\partial \psi(z)}{\partial z} \dot{z} = f(t, \psi(z)). \quad (4.22)$$

Durch die Voraussetzung $f(t, y) \in T_y M$ (Dimension von $T_y M$ ist m) ist die Differentialgleichung (4.22) durch Verwendung der Pseudoinversen von $\frac{\partial \psi(z)}{\partial z}$ äquivalent zu

$$\dot{z} = \frac{\partial \psi(z)}{\partial z}^+ f(t, \psi(z)), \quad (4.23)$$

wobei die Pseudoinverse gegeben ist durch $\frac{\partial \psi(z)}{\partial z}^+ = \left(\frac{\partial \psi(z)}{\partial z}^T \frac{\partial \psi(z)}{\partial z} \right)^{-1} \frac{\partial \psi(z)}{\partial z}^T$. Somit ist die Differentialgleichung (4.22) äquivalent zu

$$\dot{z} = F(z) := \frac{\partial \psi(z)}{\partial z}^+ f(t, \psi(z)). \quad (4.24)$$

Die Idee ist, einen Schritt einer numerischen Methode angewandt auf Gleichung (4.24) zu berechnen und das Ergebnis mit der Koordinatentransformation ψ auf die Mannigfaltigkeit zu transformieren. Ein Schritt $y_n \mapsto y_{n+1}$ der Methode basierend auf lokaler Zustandsraumtransformation folgt folgendem Schema:

Algorithmus 4.6 (Verwendung lokaler Koordinaten).

- Es wird eine lokale Koordinatisierung gewählt und z_n mithilfe von $y_n = \psi(z_n)$ berechnet.
- Es wird $\hat{z}_{n+1} = \Phi_h(z_n)$ mithilfe eines numerischen Verfahrens Φ_h angewendet auf (4.24) berechnet.
- $y_{n+1} = \psi(\hat{z}_{n+1})$

Bemerkung 4.7.

- Die im obigen Algorithmus verwendete Koordinatisierung $y = \psi(z)$ kann in jedem Schritt geändert werden.
- Am effizientesten ist das Verfahren, wenn eine globale Koordinatentransformation gefunden werden kann, da dann die Koordinatisierung nicht gewechselt werden muss.

Es gibt einige verschiedene Möglichkeiten, die lokalen Koordinaten zu wählen, wenn eine Mannigfaltigkeit durch $M = \{y \in \mathbb{R}^n : h(y) = 0\}$ gegeben ist. Zwei dieser Möglichkeiten werden im Folgenden kurz vorgestellt:

- Verallgemeinerte Koordinatenzerlegung:
Basierend auf einer QR-Zerlegung der Matrix $\frac{\partial h}{\partial y}$ wird für z eine geeignete Teilmenge der Komponenten von y gewählt.

- Tangentialraumkoordinatisierung:

Sei Q_0 eine Basis des Tangentialraums $T_{y_0}M$ im Punkt y_0 . Es wird

$$y - y_0 = \underbrace{Q_0 z}_{\in T_{y_0}M} + \underbrace{\frac{\partial h}{\partial y}(y_0)^T u}_{\in (T_{y_0}M)^\perp}, \quad (4.25)$$

mit $u = u(z)$ so, dass $h(y) = 0$, gesetzt. Damit gilt

$$y = y_0 + Q_0 z + \frac{\partial h}{\partial y}(y_0)^T u = \psi(z) = \psi_{y_0}(z). \quad (4.26)$$

In Abbildung 4.3 ist die Vorgehensweise der Tangentialraumkoordinatisierung schematisch dargestellt.

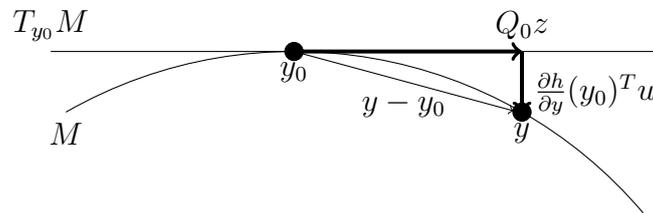


Abbildung 4.3: Illustration der Tangentialraumkoordinatisierung (Quelle: [12])

Methodenvergleich – Fallstudie Pendel

In diesem Kapitel werden die verschiedenen Methoden aus Kapitel 4 anhand eines Beispiels miteinander verglichen. Als Anschauungsbeispiel wird die Bewegung eines Pendels auf einer Kreisbahn in kartesischen Koordinaten verwendet. Der Einfachheit halber werden die Masse und die Länge des Seils mit eins angenommen. Da gewöhnlich sowohl die die Zwangsbedingung definierende Funktion als auch die Erdbeschleunigung mit g bezeichnet werden, wird hier die die Zwangsbedingung definierende Funktion mit g und die Erdbeschleunigung mit \mathbf{g} (fett) bezeichnet. Die Bewegung des Pendels ist in expliziter Form durch die folgenden Gleichungen gegeben und in Abbildung 5.1 schematisch illustriert.

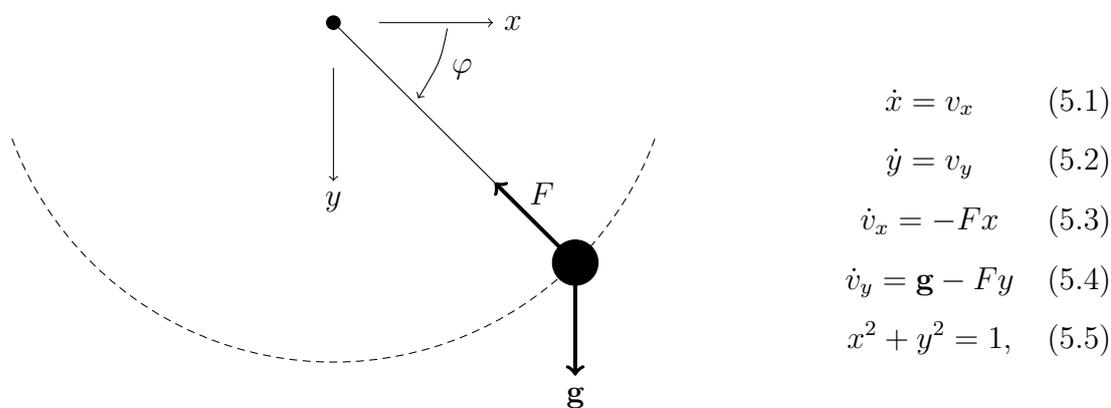


Abbildung 5.1: Schaubild und Gleichungen des Pendels (Quelle: [24])

In den obigen Gleichungen bezeichnet F die Rückführkraft (die Kraft, die das Pendel auf der Kreisbahn hält) und \mathbf{g} die Erdbeschleunigung. Mathematisch gesehen kann F als Lagrange-Multiplikator interpretiert werden. Die Zwangsbedingung ist gegeben durch Gleichung (5.5), das heißt $g: \mathbb{R}^2 \rightarrow \mathbb{R}$ ist definiert als $g(x, y) = x^2 + y^2 - 1$. Die vier Differentialgleichungen erster Ordnung können auch als zwei Differentialgleichungen

zweiter Ordnung geschrieben werden, das heißt

$$\ddot{x} = -Fx \quad \text{und} \quad \ddot{y} = \mathbf{g} - Fy. \quad (5.6)$$

5.1 Analyse der DAE

Im Folgenden werden die beiden Vorgehensweisen aus Abschnitt 3.4 diskutiert. Dazu werden zuerst die kinetische Energie E_{kin} und die potentielle Energie E_{pot} berechnet als

$$E_{kin} = \frac{\dot{x}^2 + \dot{y}^2}{2} \quad \text{und} \quad E_{pot} = -\mathbf{g}y. \quad (5.7)$$

- Variationsrechnung:

Die Funktion $L: \mathbb{R}^4 \rightarrow \mathbb{R}$ ist definiert als

$$L(x, y, \dot{x}, \dot{y}) = E_{kin} - E_{pot} = \frac{\dot{x}^2 + \dot{y}^2}{2} + \mathbf{g}y. \quad (5.8)$$

Die Euler–Lagrange Gleichungen lauten

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}} \right) - \frac{\partial (L + \lambda g(x, y))}{\partial x} &= 0 \\ \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{y}} \right) - \frac{\partial (L + \lambda g(x, y))}{\partial y} &= 0. \end{aligned} \quad (5.9)$$

Einsetzen und Ableiten ergibt

$$\begin{aligned} \frac{d}{dt} (\dot{x}) - 2\lambda x &= 0 \\ \frac{d}{dt} (\dot{y}) - \mathbf{g} - 2\lambda y &= 0, \end{aligned} \quad (5.10)$$

woraus durch erneutes Auflösen der Ableitung die Gleichungen

$$\begin{aligned} \ddot{x} - 2\lambda x &= 0 \\ \ddot{y} - \mathbf{g} - 2\lambda y &= 0, \end{aligned} \quad (5.11)$$

folgen. Insgesamt ergibt sich

$$\ddot{x} = 2x\lambda \quad (5.12)$$

$$\ddot{y} = \mathbf{g} + 2y\lambda \quad (5.13)$$

$$x^2 + y^2 - 1 = 0. \quad (5.14)$$

Die beiden Differentialgleichungen zweiter Ordnung können durch Einführung von zwei Variablen auf vier Differentialgleichungen erster Ordnung transformiert werden. Durch $F = -2\lambda$ resultiert das am Beginn des Abschnitts genannte System (5.1)–(5.5).

- Lagrange-Formalismus der klassischen Mechanik:

Die Funktion $L: \mathbb{R}^6 \rightarrow \mathbb{R}$ ist definiert als

$$L(x, y, \lambda, \dot{x}, \dot{y}, \dot{\lambda}) = E_{kin} - E_{pot} - \lambda g(x, y) = \frac{\dot{x}^2 + \dot{y}^2}{2} + \mathbf{g}y - \lambda(x^2 + y^2 - 1) \quad (5.15)$$

und daraus resultieren mit $\frac{d}{dt} \frac{\partial L}{\partial \dot{x}} - \frac{\partial L}{\partial x} = 0$, $\frac{d}{dt} \frac{\partial L}{\partial \dot{y}} - \frac{\partial L}{\partial y} = 0$ und $\frac{d}{dt} \frac{\partial L}{\partial \dot{\lambda}} - \frac{\partial L}{\partial \lambda} = 0$ die Gleichungen

$$\ddot{x} = -2x\lambda \quad (5.16)$$

$$\ddot{y} = \mathbf{g} - 2y\lambda \quad (5.17)$$

$$x^2 + y^2 - 1 = 0. \quad (5.18)$$

Bemerkenswert ist, dass die Gleichung $\frac{d}{dt} \frac{\partial L}{\partial \dot{\lambda}} - \frac{\partial L}{\partial \lambda} = 0$ nach Ausrechnen die Zwangsbedingung $g = 0$ liefert. Die beiden Differentialgleichungen zweiter Ordnung können durch Einführung zweier Variablen auf vier Differentialgleichungen erster Ordnung transformiert werden. Durch $F = 2\lambda$ resultiert das am Beginn des Abschnitts genannte System (5.1)–(5.5).

Auf diese Art und Weise können die Bewegungsgleichungen des Pendels in kartesischen Koordinaten hergeleitet werden.

In impliziter Form kann die DAE (5.1)–(5.5) mit der Funktion $f: \mathbb{R}_0^+ \times \mathbb{R}^5 \times \mathbb{R}^5 \rightarrow \mathbb{R}^5$ mit $f(t, u, \dot{u}) = 0$ beschrieben werden, wobei $u = (x, y, v_x, v_y, F)$ und

$$f(t, u, \dot{u}) = \begin{pmatrix} \dot{x} - v_x \\ \dot{y} - v_y \\ v_x + Fx \\ v_y - \mathbf{g} + Fy \\ x^2 + y^2 - 1 \end{pmatrix}. \quad (5.19)$$

Die Funktion f partiell nach \dot{u} abgeleitet ergibt

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (5.20)$$

Diese Matrix ist singulär und daher kann laut dem Hauptsatz über implizite Funktionen 2.4 das differential-algebraische Gleichungssystem nicht direkt in ein gewöhnliches Differentialgleichungssystem aufgelöst werden. Die gegebene DAE besitzt daher einen differentiellen Index, der größer Null ist.

Im Folgenden werden die Zwangsbedingung, gegeben durch Gleichung (5.5), und deren Ableitungen betrachtet:

$$x^2 + y^2 - 1 = 0 \quad (5.21)$$

$$xv_x + yv_y = 0 \quad (5.22)$$

$$v_x^2 + v_y^2 - F(x^2 + y^2) + \mathbf{g}y = 0 \quad (5.23)$$

$$2v_x(-Fx) + 2v_y(\mathbf{g} - Fy) - \dot{F}(x^2 + y^2) - F(2xv_x + 2yv_y) + \mathbf{g}v_y = 0. \quad (5.24)$$

Aus diesen vier Gleichungen ist zu erkennen, dass nach dreimaligem Ableiten der algebraischen Gleichung eine Differentialgleichung entsteht. Die DAE ist daher ein Index–3–Problem. Dies ist eine typische Eigenschaft von mechanischen Systemen. Bemerkenswert ist, dass aus der zweiten Ableitung der algebraischen Gleichung die Kraft F explizit ausgedrückt werden kann als

$$F = \frac{v_x^2 + v_y^2 + \mathbf{g}y}{x^2 + y^2}. \quad (5.25)$$

Dieser Ausdruck für F kann in die vier Differentialgleichungen eingesetzt werden. Durch diese Vorgehensweise ergibt sich das Index–1–System der gegebenen DAE. Eine andere Herangehensweise ist, aus der dritten Ableitung der Zwangsbedingung die erste Ableitung der Kraft auszudrücken und dies ergibt

$$\dot{F} = \frac{-4F(xv_x + yv_y) + 3\mathbf{g}v_y}{x^2 + y^2}. \quad (5.26)$$

Wird diese Differentialgleichung zu den anderen hinzugefügt, so wird das Index–0–System der gegebenen DAE erhalten. Somit werden zwei verschiedene Systeme erhalten, die in Tabelle 5.1 dargestellt sind.

Index–0–Formulierung	Index–1–Formulierung
$\dot{x} = v_x$	$\dot{x} = v_x$
$\dot{y} = v_y$	$\dot{y} = v_y$
$\dot{v}_x = -Fx$	$\dot{v}_x = -\frac{v_x^2 + v_y^2 + \mathbf{g}y}{x^2 + y^2}x$
$\dot{v}_y = \mathbf{g} - Fy$	$\dot{v}_y = \mathbf{g} - \frac{v_x^2 + v_y^2 + \mathbf{g}y}{x^2 + y^2}y$
$\dot{F} = \frac{-4F(xv_x + yv_y) + 3\mathbf{g}v_y}{x^2 + y^2}$	

Tabelle 5.1: Links: Index–0–Formulierung der Bewegung des Pendels; Rechts: Index–1–Formulierung der Bewegung des Pendels

Es ist zu beachten, dass der Zustandsraum bei der Index–0–Formulierung fünfdimensional mit den Zustandsgrößen x, y, v_x, v_y, F ist, im Gegensatz dazu ist dieser bei der Index–1–Formulierung vierdimensional mit den Zustandsgrößen x, y, v_x, v_y .

5.2 Indexreduktionsmethoden

In diesem Abschnitt werden die verschiedenen Methoden aus Kapitel 4 für das einfache Pendel durchgerechnet.

5.2.1 Indexreduktion durch Differentiation

In diesem Abschnitt werden die drei Methoden mithilfe von Differentiation betrachtet.

5.2.1.1 Differentiation der Zwangsbedingung

Die Zwangsbedingung zweimal abgeleitet ergibt

$$v_x^2 + v_y^2 - F(x^2 + y^2) + \mathbf{g}y = 0. \quad (5.27)$$

Somit ergibt das System (5.1)–(5.4) mit Gleichung (5.27) ein Index-1-System. Dieses kann mit ode-Solvern von MATLAB gelöst werden, indem die singuläre Massenmatrix den Optionen für den ode-Solver übergeben wird.

5.2.1.2 Baumgarte-Methode

Die Baumgarte-Methode angewendet auf die Zwangsbedingung (5.5) liefert

$$2(v_x^2 + v_y^2 - Fx^2 + \mathbf{g}y - Fy^2) + 4\alpha(xv_x + yv_y) + \beta^2(x^2 + y^2 - 1) = 0, \quad (5.28)$$

wobei $\alpha > 0$. Diese Gleichung kann nach F aufgelöst werden, was

$$F = \frac{2(v_x^2 + v_y^2 + \mathbf{g}y) + 4\alpha(xv_x + yv_y) + \beta^2(x^2 + y^2 - 1)}{2(x^2 + y^2)} \quad (5.29)$$

ergibt. Wird $\alpha = \beta = 0$ gesetzt, so ergibt sich für F genau die Gleichung, die in der Index-1-Formulierung auftritt. Bei Anwendung der Baumgarte-Methode werden die Gleichungen

$$\dot{x} = v_x \quad (5.30)$$

$$\dot{y} = v_y \quad (5.31)$$

$$\dot{v}_x = -\frac{2(v_x^2 + v_y^2 + \mathbf{g}y) + 4\alpha(xv_x + yv_y) + \beta^2(x^2 + y^2 - 1)}{2(x^2 + y^2)}x \quad (5.32)$$

$$\dot{v}_y = g - \frac{2(v_x^2 + v_y^2 + \mathbf{g}y) + 4\alpha(xv_x + yv_y) + \beta^2(x^2 + y^2 - 1)}{2(x^2 + y^2)}y \quad (5.33)$$

erhalten, die mit dem ode-Solver ode45 von MATLAB gelöst werden können.

5.2.1.3 Pantelides–Algorithmus

Die Vorgehensweise des Pantelides–Algorithmus wurde in Abschnitt 4.2.3 in Algorithmus 4.1 erklärt. Der Ausgangspunkt für den Pantelides–Algorithmus ist das System (5.1)–(5.5), welches aus fünf Gleichungen und fünf Unbekannten \dot{x} , \dot{y} , \dot{v}_x , \dot{v}_y und F besteht.

- Schritt 1:
Im Gleichungssystem (5.1)–(5.5) wird die Zwangsbedingung (5.5) gefunden.
- Schritt 2:
Durch Differentiation der Zwangsbedingung (5.5) wird das Gleichungssystem

$$\begin{aligned}
 \dot{x} &= v_x \\
 \dot{y} &= v_y \\
 \dot{v}_x &= -Fx \\
 \dot{v}_y &= \mathbf{g} - Fy \\
 x^2 + y^2 &= 1 \\
 x\dot{x} + y\dot{y} &= 0
 \end{aligned} \tag{5.34}$$

erhalten. Dieses Gleichungssystem besteht aus sechs Gleichungen und den fünf Unbekannten \dot{x} , \dot{y} , \dot{v}_x , \dot{v}_y und F .

- Schritt 3:
In diesem Schritt muss ein Integrator, der in Zusammenhang mit der Zwangsbedingung bzw. deren Ableitung steht, aufgebrochen werden. Dies geschieht durch Auflösen des Integrators für x , das heißt die Variable dx wird eingeführt und ersetzt \dot{x} , das heißt $dx := \dot{x}$. Dadurch gibt es im Folgenden \dot{x} nicht mehr. Obiges Gleichungssystem wird zu

$$\begin{aligned}
 dx &= v_x \\
 \dot{y} &= v_y \\
 \dot{v}_x &= -Fx \\
 \dot{v}_y &= \mathbf{g} - Fy \\
 x^2 + y^2 &= 1 \\
 xdx + y\dot{y} &= 0.
 \end{aligned} \tag{5.35}$$

Dieses Gleichungssystem besteht aus sechs Gleichungen und sechs Unbekannten x , dx , \dot{y} , \dot{v}_x , \dot{v}_y und F . Durch das Aufbrechen eines Integrators entspricht die Anzahl der Gleichungen der Anzahl der Unbekannten. Da in diesem Gleichungssystem eine Zwangsbedingung enthalten ist, muss der Pantelides–Algorithmus noch einmal angewendet werden.

- Schritt 1:
Im Gleichungssystem (5.35) tritt die Zwangsbedingung $dx = v_x$ auf.

- Schritt 2:

Daher muss die Zwangsbedingung, die für obiges System gegeben ist durch $dx = v_x$, differenziert werden. Die Differentiation führt auf die Gleichung $d2x = \dot{v}_x$, da \dot{dx} nicht existiert, errechnet sich die Ableitung von dx zu $d2x$, das heißt $d2x := \dot{dx}$. Dadurch entsteht das folgende Gleichungssystem

$$\begin{aligned}
 dx &= v_x \\
 d2x &= \dot{v}_x \\
 \dot{y} &= v_y \\
 \dot{v}_x &= -Fx \\
 \dot{v}_y &= \mathbf{g} - Fy \\
 x^2 + y^2 &= 1 \\
 xdx + y\dot{y} &= 0.
 \end{aligned} \tag{5.36}$$

Dieses Gleichungssystem besteht aus sieben Gleichungen und sieben Unbekannten x , dx , $d2x$, \dot{y} , \dot{v}_x , \dot{v}_y und F .

- Schritt 3:

In diesem Schritt muss ein Integrator, der in Zusammenhang mit der Zwangsbedingung bzw. deren Ableitungen steht, aufgelöst werden. Dies geschieht durch Auflösen des Integrators für \dot{v}_x , das heißt die Variable dv_x wird eingeführt und ersetzt \dot{v}_x , das heißt $dv_x := \dot{v}_x$. Dadurch gibt es im Folgenden \dot{v}_x nicht mehr. Obiges Gleichungssystem wird zu

$$\begin{aligned}
 dx &= v_x \\
 d2x &= dv_x \\
 \dot{y} &= v_y \\
 dv_x &= -Fx \\
 \dot{v}_y &= \mathbf{g} - Fy \\
 x^2 + y^2 &= 1 \\
 xdx + y\dot{y} &= 0.
 \end{aligned} \tag{5.37}$$

Dieses Gleichungssystem besteht aus sieben Gleichungen und acht Unbekannten x , dx , $d2x$, \dot{y} , v_x , dv_x , \dot{v}_y und F .

- Schritt 4:

Aufgrund der Einführung von $d2x$ muss auch die Gleichung, aus der dx ausgedrückt wird, das heißt $xdx + y\dot{y} = 0$, abgeleitet werden.

- Schritt 5:

Das Ableiten der Gleichung $xdx + y\dot{y} = 0$ führt mit $d2y := \dot{y}$ auf das Gleichungssystem

$$dx = v_x \tag{5.38a}$$

$$d^2x = dv_x \quad (5.38b)$$

$$\dot{y} = v_y \quad (5.38c)$$

$$dv_x = -Fx \quad (5.38d)$$

$$\dot{v}_y = \mathbf{g} - Fy \quad (5.38e)$$

$$x^2 + y^2 = 1 \quad (5.38f)$$

$$xdx + y\dot{y} = 0 \quad (5.38g)$$

$$(dx)^2 + xd^2x + (\dot{y})^2 + d^2y = 0. \quad (5.38h)$$

Dieses Gleichungssystem besteht aus acht Gleichungen und neun Unbekannten x , dx , d^2x , d^2y , \dot{y} , v_x , dv_x , \dot{v}_y und F .

- Schritt 4:
Aufgrund der Einführung von d^2y muss auch die Gleichung, aus der \dot{y} ausgedrückt wird, das heißt $\dot{y} = v_y$, abgeleitet werden.
- Schritt 5:
Das Ableiten der Gleichung $\dot{y} = v_y$ führt auf das Gleichungssystem

$$dx = v_x \quad (5.39a)$$

$$d^2x = dv_x \quad (5.39b)$$

$$\dot{y} = v_y \quad (5.39c)$$

$$d^2y = \dot{v}_y \quad (5.39d)$$

$$dv_x = -Fx \quad (5.39e)$$

$$\dot{v}_y = \mathbf{g} - Fy \quad (5.39f)$$

$$x^2 + y^2 = 1 \quad (5.39g)$$

$$xdx + y\dot{y} = 0 \quad (5.39h)$$

$$(dx)^2 + xd^2x + (\dot{y})^2 + yd^2y = 0. \quad (5.39i)$$

Dieses Gleichungssystem besteht aus neun Gleichungen und neun Unbekannten x , dx , d^2x , d^2y , \dot{y} , v_x , dv_x , \dot{v}_y und F .

Anwenden des Pantelides-Algorithmus führt somit schließlich auf ein Gleichungssystem mit neun Gleichungen und neun Unbekannten. Diese Gleichungen können umgeformt werden zu

$$\dot{y} = v_y \quad (5.40a)$$

$$x = \pm\sqrt{1 - y^2} \quad (5.40b)$$

$$dx = -\frac{y}{x}\dot{y} \quad (5.40c)$$

$$dv_x = d2x \quad (5.40d)$$

$$F = -\frac{dv_x}{x} \quad (5.40e)$$

$$\dot{v}_y = \mathbf{g} - Fy \quad (5.40f)$$

$$d2y = \dot{v}_y \quad (5.40g)$$

$$d2x = -\frac{(dx)^2 + (\dot{y})^2 + yd2y}{x} \quad (5.40h)$$

$$v_x = dx. \quad (5.40i)$$

Wie aus den Gleichungen zu erkennen ist, können diese nicht global verwendet werden, um die DAE zu lösen, da in der Gleichung $x = \pm\sqrt{1 - y^2}$ entschieden werden muss, ob x positiv oder negativ ist.

Wird im ersten Schritt des Pantelides-Algorithmus nicht, wie oben, die Variable dx , sondern dy eingeführt, so wird das System

$$dy = v_y \quad (5.41a)$$

$$d2y = dv_y \quad (5.41b)$$

$$\dot{x} = v_x \quad (5.41c)$$

$$d2x = \dot{v}_x \quad (5.41d)$$

$$dv_y = \mathbf{g} - Fy \quad (5.41e)$$

$$\dot{v}_x = -Fx \quad (5.41f)$$

$$x^2 + y^2 = 1 \quad (5.41g)$$

$$x\dot{x} + ydy = 0 \quad (5.41h)$$

$$(\dot{x})^2 + xd2x + (dy)^2 + yd2y = 0 \quad (5.41i)$$

erhalten. Dieses Gleichungssystem besteht aus neun Gleichungen und neun Unbekannten \dot{x} , $d2x$, y , dy , $d2y$, \dot{v}_x , v_y , dv_y und F . Diese Gleichungen können umgeformt werden zu

$$\dot{x} = v_x \quad (5.42a)$$

$$y = \pm\sqrt{1 - x^2} \quad (5.42b)$$

$$dy = -\frac{x}{y}\dot{x} \quad (5.42c)$$

$$dv_y = d^2y \quad (5.42d)$$

$$F = -\frac{dv_y - \mathbf{g}}{y} \quad (5.42e)$$

$$\dot{v}_x = -Fx \quad (5.42f)$$

$$d^2x = \dot{v}_x \quad (5.42g)$$

$$d^2y = -\frac{(dy)^2 + (\dot{x})^2 + xd^2x}{y} \quad (5.42h)$$

$$v_y = dy. \quad (5.42i)$$

Wie oben schon erwähnt, können diese Gleichungen ebenfalls nicht global zum Lösen verwendet werden, da $y = \pm\sqrt{1-x^2}$. Daher werden die Gleichungssysteme (5.40) und (5.42) in vier Gleichungssysteme aufgeteilt:

- Gleichungssystem (5.40) mit $x = \sqrt{1-y^2}$
- Gleichungssystem (5.40) mit $x = -\sqrt{1-y^2}$
- Gleichungssystem (5.42) mit $y = \sqrt{1-x^2}$
- Gleichungssystem (5.42) mit $y = -\sqrt{1-x^2}$.

Diese vier Gleichungssysteme werden jeweils auf bestimmten Teilbereichen der Einheitskreislinie verwendet. Diese Teilbereiche sind in Abbildung 5.2 dargestellt.

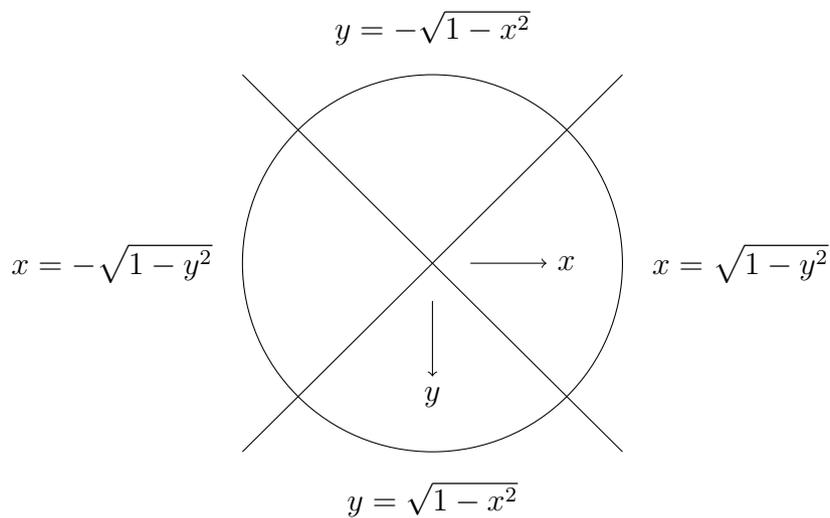


Abbildung 5.2: Einheitskreislinie mit der entsprechenden Aufteilung für das Lösen der DAE mithilfe des Pantelides-Algorithmus

5.2.2 Stabilisierung durch Projektion

Bei diesen Methoden wird die Index–1–Formulierung, siehe Tabelle 5.1, mithilfe von Einschrittverfahren gelöst. Durch die Verwendung einer Projektion wird sichergestellt, dass die numerische Lösung auf der Mannigfaltigkeit liegt. Diese Mannigfaltigkeit ist gegeben durch

$$M = \{(x, y, v_x, v_y) \in \mathbb{R}^4 : h(x, y, v_x, v_y) = 0\}, \quad (5.43)$$

wobei $h: \mathbb{R}^4 \rightarrow \mathbb{R}^2$ definiert ist als

$$h(x, y, v_x, v_y) = \begin{pmatrix} x^2 + y^2 - 1 \\ xv_x + yv_y \end{pmatrix}. \quad (5.44)$$

Die Funktion h errechnet sich aus der Zwangsbedingung $g = 0$ und deren erster Ableitung, da das System Index 3 besitzt. Die Jacobimatrix $\frac{\partial h}{\partial(x,y,v_x,v_y)}$ von h lautet

$$\begin{pmatrix} 2x & 2y & 0 & 0 \\ v_x & v_y & x & y \end{pmatrix}. \quad (5.45)$$

Diese hat für alle $(x, y, v_x, v_y) \in \mathbb{R}^4 \setminus \{(0, 0, a, b)\}$ mit $a, b \in \mathbb{R}$ ($(0, 0, a, b)$ kann nicht auftreten, da immer $x^2 + y^2 = 1$ gelten muss) Rang 2 und ist daher nach Lemma 2.6 eine implizit definierte Mannigfaltigkeit. M ist eine 2–dimensionale Mannigfaltigkeit im \mathbb{R}^4 .

Der Tangentialraum von M errechnet sich mit $z = (x, y, v_x, v_y)$ und $w = (x, y)$ in diesem Fall als

$$T_z M = \left\{ v \in \mathbb{R}^4 : \frac{\partial g}{\partial w}(z) \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = 0, (v_x, v_y) \frac{\partial^2 g}{\partial w^2}(z) \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} + \frac{\partial g}{\partial w}(z) \frac{\partial \dot{w}}{\partial \dot{w}} \begin{pmatrix} v_3 \\ v_4 \end{pmatrix} = 0 \right\}. \quad (5.46)$$

Umformungen ergeben

$$T_z M = \{v \in \mathbb{R}^4 : xv_1 + yv_2 = 0, v_x v_1 + v_y v_2 + xv_3 + yv_4 = 0\}. \quad (5.47)$$

Eine Basis des 2–dimensionalen Tangentialraumes $T_z M$ ist daher gegeben durch

$$\begin{pmatrix} -y & 0 \\ x & 0 \\ -v_y & -y \\ v_x & x \end{pmatrix}. \quad (5.48)$$

Im Folgenden werden zwei verschiedene Methoden, die diesen Ansatz verwenden, betrachtet. Diese sind die sogenannte Standardprojektionsmethode oder orthogonale Projektionsmethode und die symmetrische Projektionsmethode.

5.2.2.1 Standardprojektionsmethode

Die Projektion wird mittels der Abbildung $P: \mathbb{R}^4 \rightarrow M \subset \mathbb{R}^4$

$$\begin{aligned}
 x &\mapsto \frac{x}{\sqrt{x^2 + y^2}} \\
 y &\mapsto \frac{y}{\sqrt{x^2 + y^2}} \\
 v_x &\mapsto \frac{-yv_x + xv_y}{x^2 + y^2}(-y) \\
 v_y &\mapsto \frac{-yv_x + xv_y}{x^2 + y^2}x
 \end{aligned} \tag{5.49}$$

durchgeführt. Um einzusehen, dass P eine Projektion darstellt, muss $P^2 = P$ nachgerechnet werden, wobei $P^2 = P \circ P$ ist.

$$P^2 \begin{pmatrix} x \\ y \\ v_x \\ v_y \end{pmatrix} = P \begin{pmatrix} \frac{x}{\sqrt{x^2 + y^2}} \\ \frac{y}{\sqrt{x^2 + y^2}} \\ \frac{-yv_x + xv_y}{x^2 + y^2}(-y) \\ \frac{-yv_x + xv_y}{x^2 + y^2}x \end{pmatrix} = \begin{pmatrix} \frac{x}{\sqrt{x^2 + y^2}} \\ \frac{y}{\sqrt{x^2 + y^2}} \\ \frac{-yv_x + xv_y}{x^2 + y^2}(-y) \\ \frac{-yv_x + xv_y}{x^2 + y^2}x \end{pmatrix} = P \begin{pmatrix} x \\ y \\ v_x \\ v_y \end{pmatrix} \tag{5.50}$$

Die vorletzte Gleichheit wird nachfolgend exemplarisch für die erste und dritte Komponente gezeigt, die beiden anderen Komponenten folgen analog.

$$\begin{aligned}
 \pi_1 \circ P \left(\left(\frac{x}{\sqrt{x^2 + y^2}}, \frac{y}{\sqrt{x^2 + y^2}}, \frac{-yv_x + xv_y}{x^2 + y^2}(-y), \frac{-yv_x + xv_y}{x^2 + y^2}x \right)^T \right) &= \\
 &= \frac{\frac{x}{\sqrt{x^2 + y^2}}}{\sqrt{\left(\frac{x}{\sqrt{x^2 + y^2}} \right)^2 + \left(\frac{y}{\sqrt{x^2 + y^2}} \right)^2}} = \frac{x}{\sqrt{x^2 + y^2}}
 \end{aligned} \tag{5.51}$$

$$\begin{aligned}
 \pi_3 \circ P \left(\left(\frac{x}{\sqrt{x^2 + y^2}}, \frac{y}{\sqrt{x^2 + y^2}}, \frac{-yv_x + xv_y}{x^2 + y^2}(-y), \frac{-yv_x + xv_y}{x^2 + y^2}x \right)^T \right) &= \\
 = \left(-\frac{y}{\sqrt{x^2 + y^2}} \frac{-yv_x + xv_y}{x^2 + y^2}(-y) + \frac{x}{\sqrt{x^2 + y^2}} \frac{-yv_x + xv_y}{x^2 + y^2}x \right) \left(-\frac{y}{\sqrt{x^2 + y^2}} \right) &= \\
 = \left(\frac{-yv_x + xv_y}{x^2 + y^2} \right) (-y)
 \end{aligned} \tag{5.52}$$

Daher gilt $P^2 = P$. Um auch noch die Orthogonalität von P zu beweisen muss

$$(z - Pz, \tau)_{\mathbb{R}^4} = 0 \quad \forall \tau \in T_z M \tag{5.53}$$

nachgerechnet werden, das heißt

$$(z - Pz, b_i)_{\mathbb{R}^4} = 0 \quad i = 1, 2, \quad (5.54)$$

wobei (b_1, b_2) die Basis von $T_z M$ ist (siehe (5.48)).

Für den Basisvektor b_1 errechnet sich das Skalarprodukt zu

$$\begin{aligned} - \left(x - \frac{x}{\sqrt{x^2 + y^2}} \right) y + \left(y - \frac{y}{\sqrt{x^2 + y^2}} \right) x - \left(v_x - \frac{-yv_x + xv_y}{x^2 + y^2}(-y) \right) v_y + \\ + \left(v_y - \frac{-yv_x + xv_y}{x^2 + y^2}x \right) v_x \stackrel{!}{=} 0 \\ \frac{-yv_x + xv_y}{x^2 + y^2}(-y)v_y - \frac{-yv_x + xv_y}{x^2 + y^2}xv_x \stackrel{!}{=} 0 \\ \frac{-yv_x + xv_y}{x^2 + y^2}(yv_y + xv_x) \stackrel{!}{=} 0 \\ (-yv_x + xv_y)(yv_y + xv_x) \stackrel{!}{=} 0. \end{aligned} \quad (5.55)$$

Die letzte Gleichheit ist nicht erfüllt, daher ist die angegebene Projektion nicht orthogonal auf die gesamte Mannigfaltigkeit.

Für den Basisvektor b_2 errechnet sich das Skalarprodukt zu

$$\begin{aligned} - \left(v_x - \frac{-yv_x + xv_y}{x^2 + y^2}(-y) \right) y + \left(v_y - \frac{-yv_x + xv_y}{x^2 + y^2}x \right) x \stackrel{!}{=} 0 \\ - \frac{-yv_x + xv_y}{x^2 + y^2}(x^2 + y^2) - v_x y + xv_y \stackrel{!}{=} 0 \\ yv_x - xv_y - v_x y + xv_y \stackrel{!}{=} 0 \\ 0 = 0. \end{aligned} \quad (5.56)$$

Da die Projektion nicht auf die gesamte Mannigfaltigkeit orthogonal ist, wird im Folgenden die Projektion aufgeteilt. Zuerst werden die Positionskomponenten projiziert, anschließend die Geschwindigkeitskomponenten. Dieses Vorgehen wird im Weiteren als orthogonale Projektion bezeichnet. Die Menge

$$M_1 = \{(x, y)^T \in \mathbb{R}^2 : h_1 = x^2 + y^2 - 1 = 0\} \quad (5.57)$$

ist laut Lemma 2.6 eine eindimensionale Mannigfaltigkeit im \mathbb{R}^2 , da der Gradient die Gestalt $dh_1 = (2x, 2y)$ und somit Rang 1 für alle $x, y \in \mathbb{R}^2 \setminus \{(0, 0)^T\}$ hat. Die Abbildung $p_1 : \mathbb{R}^2 \rightarrow M_1$ wird folgendermaßen definiert

$$\begin{aligned} x &\mapsto \frac{x}{\sqrt{x^2 + y^2}} \\ y &\mapsto \frac{y}{\sqrt{x^2 + y^2}}. \end{aligned} \quad (5.58)$$

Die Menge

$$M_2 = \{(v_x, v_y)^T \in \mathbb{R}^2 : h_2 = \pi_1 \circ p_1((x, y)^T)v_x + \pi_2 \circ p_1((x, y)^T)v_y = 0\} \quad (5.59)$$

ist laut Lemma 2.6 eine eindimensionale Mannigfaltigkeit im \mathbb{R}^2 , da der Gradient die Gestalt $dh_2 = (\pi_1 \circ p_1((x, y)^T), \pi_2 \circ p_1((x, y)^T))$ und somit Rang 1 für alle $v_x, v_y \in \mathbb{R}^2$ hat.

Die Abbildung $p_2 : \mathbb{R}^2 \rightarrow M_2$ wird definiert wie folgt

$$\begin{aligned} v_x &\mapsto (-\pi_2 \circ p_1((x, y)^T)v_x + \pi_1 \circ p_1((x, y)^T)v_y)(-\pi_2 \circ p_1((x, y)^T)) \\ v_y &\mapsto (-\pi_2 \circ p_1((x, y)^T)v_x + \pi_1 \circ p_1((x, y)^T)v_y)\pi_1 \circ p_1((x, y)^T). \end{aligned} \quad (5.60)$$

Die Abbildung p_1 erfüllt $p_1^2 = p_1$, da

$$p_1^2 \begin{pmatrix} x \\ y \end{pmatrix} = p_1 \left(p_1 \begin{pmatrix} x \\ y \end{pmatrix} \right) = p_1 \left(\begin{pmatrix} \frac{x}{\sqrt{x^2 + y^2}} \\ \frac{y}{\sqrt{x^2 + y^2}} \end{pmatrix} \right) = \begin{pmatrix} \frac{x}{\sqrt{x^2 + y^2}} \\ \frac{y}{\sqrt{x^2 + y^2}} \end{pmatrix} = p_1 \begin{pmatrix} x \\ y \end{pmatrix}. \quad (5.61)$$

Die Tangentialebene von M_1 ist durch

$$T_{(x,y)^T} M_1 = \{v \in \mathbb{R}^2 : xv_1 + yv_2 = 0\} \quad (5.62)$$

und die Basis der Tangentialebene durch $(-y, x)^T$ gegeben.

Die Abbildung p_1 ist orthogonal auf M_1 , das heißt

$$\left(\begin{pmatrix} x \\ y \end{pmatrix} - p_1 \begin{pmatrix} x \\ y \end{pmatrix}, \begin{pmatrix} -y \\ x \end{pmatrix} \right)_{\mathbb{R}^2} = 0, \quad (5.63)$$

da

$$\begin{aligned} - \left(x - \frac{x}{\sqrt{x^2 + y^2}} \right) y + \left(y - \frac{y}{\sqrt{x^2 + y^2}} \right) x &\stackrel{!}{=} 0 \\ -xy + \frac{xy}{\sqrt{x^2 + y^2}} + xy - \frac{xy}{\sqrt{x^2 + y^2}} &\stackrel{!}{=} 0 \\ 0 &= 0. \end{aligned} \quad (5.64)$$

Die Abbildung p_2 erfüllt $p_2^2 = p_2$, da

$$\begin{aligned} p_2^2 \begin{pmatrix} v_x \\ v_y \end{pmatrix} &= p_2 \left(p_2 \begin{pmatrix} v_x \\ v_y \end{pmatrix} \right) = \\ &= p_2 \begin{pmatrix} (-\pi_2 \circ p_1((x, y)^T)v_x + \pi_1 \circ p_1((x, y)^T)v_y)(-\pi_2 \circ p_1((x, y)^T)) \\ (-\pi_2 \circ p_1((x, y)^T)v_x + \pi_1 \circ p_1((x, y)^T)v_y)\pi_1 \circ p_1((x, y)^T) \end{pmatrix} = \\ &= \begin{pmatrix} (-\pi_2 \circ p_1((x, y)^T)v_x + \pi_1 \circ p_1((x, y)^T)v_y)(-\pi_2 \circ p_1((x, y)^T)) \\ (-\pi_2 \circ p_1((x, y)^T)v_x + \pi_1 \circ p_1((x, y)^T)v_y)\pi_1 \circ p_1((x, y)^T) \end{pmatrix} = p_2 \begin{pmatrix} v_x \\ v_y \end{pmatrix}, \end{aligned} \quad (5.65)$$

wobei die Beziehung $(\pi_1 \circ p_1((x, y)^T))^2 + (\pi_2 \circ p_1((x, y)^T))^2 = 1$ bei den Umformungen verwendet wird.

Die Tangentialebene von M_2 ist gegeben durch

$$T_{(v_x, v_y)^T} M_2 = \{v \in \mathbb{R}^2 : \pi_1 \circ p_1((x, y)^T)v_1 + \pi_2 \circ p_1((x, y)^T)v_2 = 0\} \quad (5.66)$$

und die Basis der Tangentialebene durch $(-\pi_2 \circ p_1((x, y)^T), \pi_1 \circ p_1((x, y)^T))^T$. Die Abbildung p_2 ist orthogonal auf M_2 , das heißt

$$\left(\begin{pmatrix} v_x \\ v_y \end{pmatrix} - p_2 \begin{pmatrix} v_x \\ v_y \end{pmatrix}, \begin{pmatrix} -\pi_2 \circ p_1 \begin{pmatrix} x \\ y \end{pmatrix} \\ \pi_1 \circ p_1 \begin{pmatrix} x \\ y \end{pmatrix} \end{pmatrix} \right)_{\mathbb{R}^2} = 0, \quad (5.67)$$

da

$$\begin{aligned} & -(v_x - (-\pi_2 \circ p_1((x, y)^T)v_x + \pi_1 \circ p_1((x, y)^T)v_y)(-\pi_2 \circ p_1((x, y)^T)))\pi_2 \circ p_1((x, y)^T) + \\ & +(v_y - (-\pi_2 \circ p_1((x, y)^T)v_x + \pi_1 \circ p_1((x, y)^T)v_y)\pi_1 \circ p_1((x, y)^T))\pi_1 \circ p_1((x, y)^T) \stackrel{!}{=} 0 \\ & \quad -\pi_2 \circ p_1((x, y)^T)v_x + \pi_1 \circ p_1((x, y)^T)v_y + \\ & +(\pi_2 \circ p_1((x, y)^T)v_x - \pi_1 \circ p_1((x, y)^T)v_y) \underbrace{((\pi_2 \circ p_1((x, y)^T))^2 + (\pi_1 \circ p_1((x, y)^T))^2)}_{=1} \stackrel{!}{=} 0 \\ & 0 = 0. \end{aligned} \quad (5.68)$$

Das Vorgehen ist in Abbildung 5.3 schematisch dargestellt.

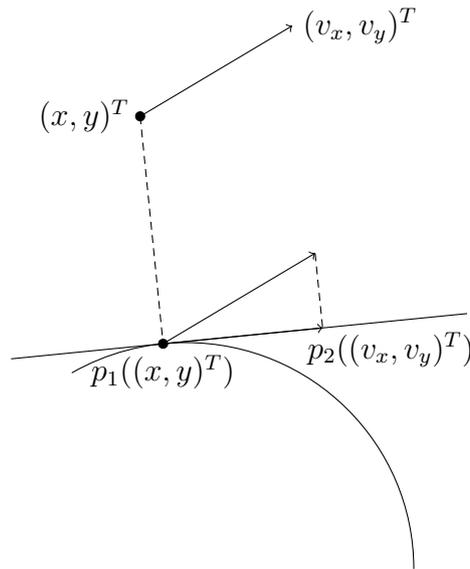


Abbildung 5.3: Orthogonale Projektion

Die orthogonale Projektion $p : \mathbb{R}^4 \rightarrow M$ wird definiert als

$$p \begin{pmatrix} x \\ y \\ v_x \\ v_y \end{pmatrix} = \begin{pmatrix} p_1 \begin{pmatrix} x \\ y \end{pmatrix} \\ p_2 \begin{pmatrix} v_x \\ v_y \end{pmatrix} \end{pmatrix}. \quad (5.69)$$

5.2.2.2 Symmetrische Projektionsmethode

Für das zu lösende Gleichungssystem (4.9) ergibt sich

$$\begin{pmatrix} 1 & 0 & 0 & 0 & -4x_{n+1} & -2v_{x_{n+1}} \\ 0 & 1 & 0 & 0 & -4y_{n+1} & -2v_{y_{n+1}} \\ 0 & 0 & 1 & 0 & 0 & -2x_{n+1} \\ 0 & 0 & 0 & 1 & 0 & -2y_{n+1} \\ 2x_{n+1} & 2y_{n+1} & 0 & 0 & 0 & 0 \\ v_{x_{n+1}} & v_{y_{n+1}} & x_{n+1} & y_{n+1} & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta x_{n+1} \\ \Delta y_{n+1} \\ \Delta v_{x_{n+1}} \\ \Delta v_{y_{n+1}} \\ \Delta \mu_1 \\ \Delta \mu_2 \end{pmatrix} = \begin{pmatrix} \hat{x}_n + h\Psi(h, \hat{x}_n, \hat{x}_{n+1}) - \hat{x}_{n+1} \\ \hat{y}_n + h\Psi(h, \hat{y}_n, \hat{y}_{n+1}) - \hat{y}_{n+1} \\ \hat{v}_{x_n} + h\Psi(h, \hat{v}_{x_n}, \hat{v}_{x_{n+1}}) - \hat{v}_{x_{n+1}} \\ \hat{v}_{y_n} + h\Psi(h, \hat{v}_{y_n}, \hat{v}_{y_{n+1}}) - \hat{v}_{y_{n+1}} \\ x_{n+1}^2 + y_{n+1}^2 - 1 \\ x_{n+1}v_{x_{n+1}} + y_{n+1}v_{y_{n+1}} \end{pmatrix} \quad (5.70)$$

Als Verfahren wird die Trapezregel (exemplarisch für \hat{x} gezeigt)

$$\hat{x}_{n+1} = \hat{x}_n + h\Psi(h, \hat{x}_n, \hat{x}_{n+1}) = \Phi_h(\hat{x}_n) \quad \text{mit} \quad \Psi = \frac{\hat{x}_n + \hat{x}_{n+1}}{2} \quad (5.71)$$

verwendet. Diese ist symmetrisch, da (\hat{x}_{n+1} und \hat{x}_n , sowie h mit $-h$ austauschen)

$$\hat{x}_n = \hat{x}_{n+1} - h\Psi(-h, \hat{x}_{n+1}, \hat{x}_n) = \hat{x}_{n+1} - h\frac{\hat{x}_n + \hat{x}_{n+1}}{2}, \quad (5.72)$$

woraus

$$\hat{x}_{n+1} = \hat{x}_n + h\Psi(h, \hat{x}_n, \hat{x}_{n+1}) \quad \text{mit} \quad \Psi = \frac{\hat{x}_n + \hat{x}_{n+1}}{2} \quad (5.73)$$

folgt und somit $\Phi_h = \Phi_{-h}^{-1}$.

5.2.3 Methoden basierend auf lokaler Zustandsraumtransformation

Ausgangspunkt ist die Index-1-Formulierung, siehe Tabelle 5.1, aus der F berechnet werden kann. Das Lösen der DAE mithilfe einer Lösungsmannigfaltigkeit und lokaler Zustandsraumtransformation wird mit einer globalen Transformation umgesetzt. Die Mannigfaltigkeit ist gegeben durch

$$M = \{(x, y, v_x, v_y) \in \mathbb{R}^4 : x^2 + y^2 - 1 = 0, xv_x + yv_y = 0\}. \quad (5.74)$$

Die Transformation mit $(x, y, v_x, v_y) = \psi(\varphi, \eta)$ und

$$\begin{aligned} x &= \cos \varphi \\ y &= \sin \varphi \\ v_x &= -\eta \sin \varphi \\ v_y &= \eta \cos \varphi \end{aligned} \quad (5.75)$$

ergibt die folgende Differentialgleichung

$$\frac{\partial \psi(\varphi, \eta)}{\partial(\varphi, \eta)} \begin{pmatrix} \dot{\varphi} \\ \dot{\eta} \end{pmatrix} = f(\psi(\varphi, \eta)), \quad (5.76)$$

mit

$$\begin{pmatrix} v_x \\ v_y \\ -\frac{v_x^2 + v_y^2 + \mathbf{g}y}{x^2 + y^2}x \\ g - \frac{v_x^2 + v_y^2 + \mathbf{g}y}{x^2 + y^2}y \end{pmatrix} = f(x, y, v_x, v_y) = f(\psi(\varphi, \eta)) = \begin{pmatrix} -\eta \sin \varphi \\ \eta \cos \varphi \\ -\eta^2 \cos \varphi - \mathbf{g} \sin \varphi \cos \varphi \\ -\eta^2 \sin \varphi + \mathbf{g} \cos^2 \varphi \end{pmatrix}. \quad (5.77)$$

Die Ableitung von ψ nach (φ, η) berechnet sich zu

$$\frac{\partial \psi(\varphi, \eta)}{\partial(\varphi, \eta)} = \begin{pmatrix} -\sin \varphi & 0 \\ \cos \varphi & 0 \\ -\eta \cos \varphi & -\sin \varphi \\ -\eta \sin \varphi & \cos \varphi \end{pmatrix}. \quad (5.78)$$

Die Pseudoinverse von $A := \frac{\partial \psi(\varphi, \eta)}{\partial(\varphi, \eta)}$, $A^+ = (A^T A)^{-1} A^T$, berechnet sich mit

$$A^T A = \begin{pmatrix} 1 + \eta^2 & 0 \\ 0 & 1 \end{pmatrix} \quad (5.79)$$

beziehungsweise

$$(A^T A)^{-1} = \frac{1}{1 + \eta^2} \begin{pmatrix} 1 & 0 \\ 0 & 1 + \eta^2 \end{pmatrix} \quad (5.80)$$

zu

$$A^+ = \frac{1}{1 + \eta^2} \begin{pmatrix} -\sin \varphi & \cos \varphi & -\eta \cos \varphi & -\eta \sin \varphi \\ 0 & 0 & -(1 + \eta^2) \sin \varphi & (1 + \eta^2) \cos \varphi \end{pmatrix}. \quad (5.81)$$

Daraus folgt folgende Gestalt des Differentialgleichungssystems

$$\begin{pmatrix} \dot{\varphi} \\ \dot{\eta} \end{pmatrix} = A^+ f(\psi(\varphi, \eta)). \quad (5.82)$$

Das Differentialgleichungssystem errechnet sich somit zu

$$\begin{aligned} \dot{\varphi} &= \eta \\ \dot{\eta} &= \mathbf{g} \cos \varphi. \end{aligned} \quad (5.83)$$

Die Zwangsbedingung und deren Ableitung sind durch die gewählte Koordinatentransformation erfüllt.

Als alternative Möglichkeit kann die Transformation $x = \cos(\varphi)$ und $y = \sin(\varphi)$ direkt in die Differentialgleichungen zweiter Ordnung eingesetzt werden. Diese Koordinatentransformationen führen zu den folgenden Gleichungen für die Geschwindigkeitskomponenten

$$\begin{aligned} v_x &= -\dot{\varphi} \sin \varphi \\ v_y &= \dot{\varphi} \cos \varphi. \end{aligned} \quad (5.84)$$

Diese Transformation ergibt eingesetzt in die ursprünglichen zwei Differentialgleichungen zweiter Ordnung

$$\begin{aligned} \ddot{x} &= -Fx \\ \ddot{y} &= \mathbf{g} - Fy \end{aligned} \quad (5.85)$$

die beiden Differentialgleichungen

$$\begin{aligned} -\dot{\varphi}^2 \cos \varphi - \ddot{\varphi} \sin \varphi &= -F \cos \varphi \\ -\dot{\varphi}^2 \sin \varphi + \ddot{\varphi} \cos \varphi &= \mathbf{g} - F \sin \varphi. \end{aligned} \quad (5.86)$$

Aus der ersten und zweiten Gleichung ergibt sich (erste Gleichung multipliziert mit $-\sin \varphi$, zweite Gleichung multipliziert mit $\cos \varphi$ und anschließende Addition)

$$\ddot{\varphi} = \mathbf{g} \cos \varphi. \quad (5.87)$$

Es werden also mit der Transformation auf Polarkoordinaten dieselben Differentialgleichungen wie oben erhalten.

5.3 Simulationen

Alle folgenden Simulationen wurden in MATLAB durchgeführt. Es wurden verschiedene numerische Verfahren bzw. ode-Solver zum Lösen der Index-0-Formulierung, der Index-1-Formulierung und der nach Anwenden der Indexreduktionsmethoden entstandenen Gleichungssysteme verwendet. In [26] werden verschiedene Mehrschrittverfahren zur numerischen Lösung der Bewegungsgleichungen des einfachen Pendels diskutiert.

Alle Laufzeiten wurden mithilfe der MATLAB-Befehle `tic` und `toc`, über 10 Durchläufe gemittelt, berechnet und in Sekunden (auf drei Nachkommastellen gerundet) angegeben. Es wurde der Fehler von (x, y) zur Einheitskreislinie berechnet. Der Fehler e_i errechnet sich daher zu

$$e_i = x_i^2 + y_i^2 - 1, \quad (5.88)$$

wobei x_i, y_i die numerisch berechneten Werte für x und y sind. Es wurde auch zu jeder Simulation der maximale Fehler e

$$e = \max_i (x_i^2 + y_i^2 - 1) \quad (5.89)$$

berechnet. Sowohl bei dem expliziten und impliziten Euler–Verfahren als auch bei der Trapezregel wurde die Schrittweite 10^{-3} gewählt.

Die verwendeten Werte für die Parameter und die Simulationszeit wurden für die folgenden Simulationen folgendermaßen gewählt:

- Erdbeschleunigung: $g = 9.81$
- Endzeit der Simulation: 10 bzw. 100.

Die Berechnungen wurden auf einem Laptop mit Intel Core i5 1.6GHz Prozessor und 4GB RAM durchgeführt. Es wurde die MATLAB–Version R2012b verwendet.

5.3.1 ode–Solver

Der ode–Solver `ode45` von MATLAB verwendet explizite Einschrittverfahren, genauer: der ode–Solver verwendet Runge–Kutta–Verfahren der Ordnung 4 und 5 (RK5(4)–Verfahren von Dormand und Prince), siehe [27], und eignet sich zum Lösen von nicht steifen Differentialgleichungen. Im Gegensatz dazu sind der `ode15s` und `ode23t` zum Lösen steifer Differentialgleichungen und DAEs geeignet. Der `ode15s` verwendet NDF–Verfahren (Numerische Differentiationsverfahren, englisch: numerical differentiation formulas), die Modifikationen von BDF–Verfahren (Rückwärtsdifferenzierungsverfahren, englisch: backward differentiation formulas) sind (siehe [28]), aber NDF–Verfahren sind effizienter als BDF–Verfahren, siehe [27]. Der `ode15s` erzeugt numerisch die Jacobimatrix, siehe [27]. Dem `ode23t` liegt die Trapezregel zugrunde. Der ode–Solver `ode15i` verwendet zum Lösen BDF–Verfahren und eignet sich zum Lösen implizit gegebener Systeme, siehe [29].

5.3.2 Simulationen der Index–0–Formulierung

Für die Simulationen des Gleichungssystems der Index–0–Formulierung in Tabelle 5.1 wurden die in Tabelle 5.2 aufgelisteten Werte verwendet.

Anfangswerte					
	x	y	v_x	v_y	F
Szenario 1	1	0	0	0	0
Szenario 2	1	0	0	10	100

Tabelle 5.2: Anfangswerte für die Simulation der Bewegungsgleichungen des Pendels mit Index 0

Die Anfangswerte für F wurden so gewählt, dass sie mit den Werten von F in den Simulationen der Index–1–Formulierung zusammenpassen. Ansonsten haben die gewählten Anfangswerte keine spezielle Bedeutung.

In Abbildung 5.4 ist das Ergebnis der Simulation von Szenario 1 dargestellt.

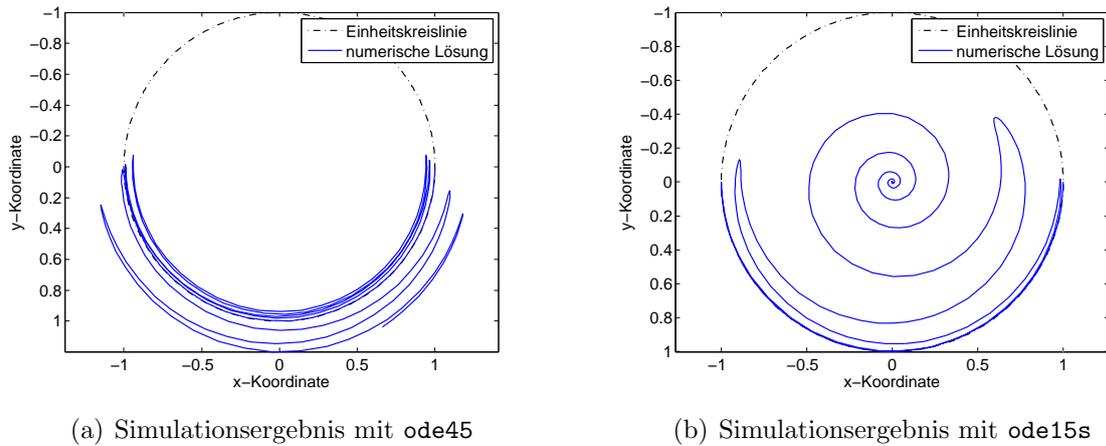


Abbildung 5.4: Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) für Szenario 1 der Index-0-Formulierung mit `ode45` und `ode15s` und Endzeit 10

Es ist zu erkennen, dass bei Verwendung des ode-Solvers `ode45` die numerische Lösung nach außen von der Einheitskreislinie wegdriftet. Im Gegensatz dazu läuft die numerische Lösung bei Verwendung des ode-Solvers `ode15s` in das Innere der Einheitskreislinie und sobald die numerische Lösung in $(0, 0)$ ankommt, bricht der Solver ab.

Zur weiteren Analyse der Simulationen der Index-0-Formulierung werden die Fehler von Szenario 1 mit `ode45` und `ode15s` berechnet. Diese sind in Abbildung 5.5 dargestellt.

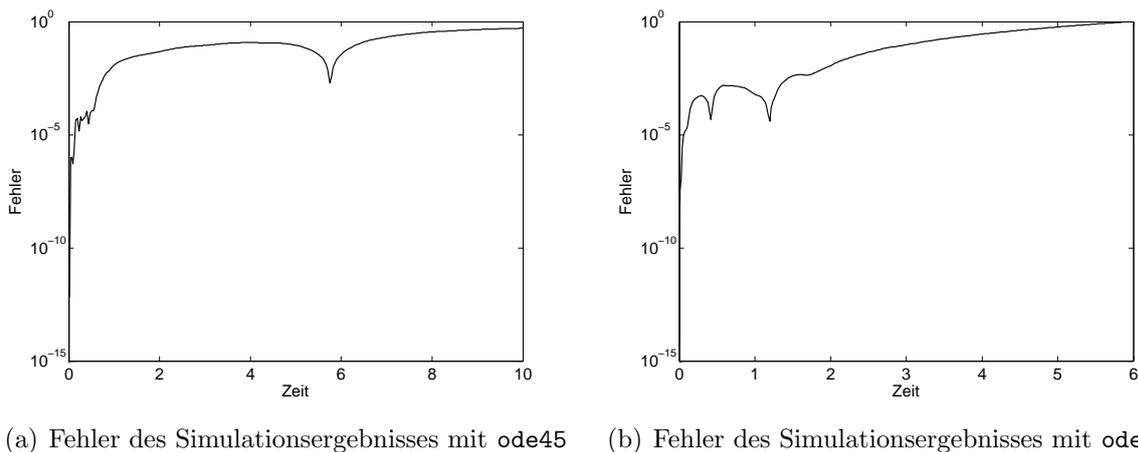


Abbildung 5.5: Fehler der numerischen Lösung zur Einheitskreislinie für Szenario 1 der Index-0-Formulierung mit `ode45` und `ode15s` und Endzeit 10

Der Fehler der Simulation mit dem `ode45` ist bei Ende der Simulation fast 1. Dieser wird bei längerer Simulationszeit immer größer. Im Gegensatz dazu ist der Fehler bei der Simulation mit dem `ode15s` maximal 1. Dies tritt genau dann ein, wenn die numerische Lösung $(0, 0)$ wird. Zu diesem Zeitpunkt bricht der ode-Solver auch ab, das passiert bei fast 6 Sekunden.

Die dargestellten Resultate aus Abbildung 5.4 und 5.5 treten auch bei Verwendung anderer ode-Solver bzw. bei dem anderen Szenario auf. Für Szenario 2 mit dem `ode45` läuft die Lösung ebenfalls in den Ursprung, genauso wie für beide Szenarien mit `ode23t`. Bei Szenario 2 mit dem Solver `ode15s` bewegt sich die numerische Lösung zuerst ins Innere der Einheitskreislinie, driftet aber dann doch nach außen weg. Dasselbe tritt für das implizite Euler-Verfahren auf. Beim expliziten Euler-Verfahren driftet die numerische Lösung nach außen weg. Im Gegensatz dazu liefert die Trapezregel für Szenario 1 ein Ergebnis, das auf der Kreisbahn bleibt. Der Grund dafür, dass die Trapezregel hier gute Resultate liefert kann vermutet werden in der Tatsache, dass für Szenario 1 das explizite Euler-Verfahren nach außen driftet wohingegen das implizite Euler-Verfahren zuerst nach innen driftet.

In Abbildung 5.6 ist das Ergebnis der Simulation von Szenario 1 und Szenario 2 mit der Trapezregel dargestellt.

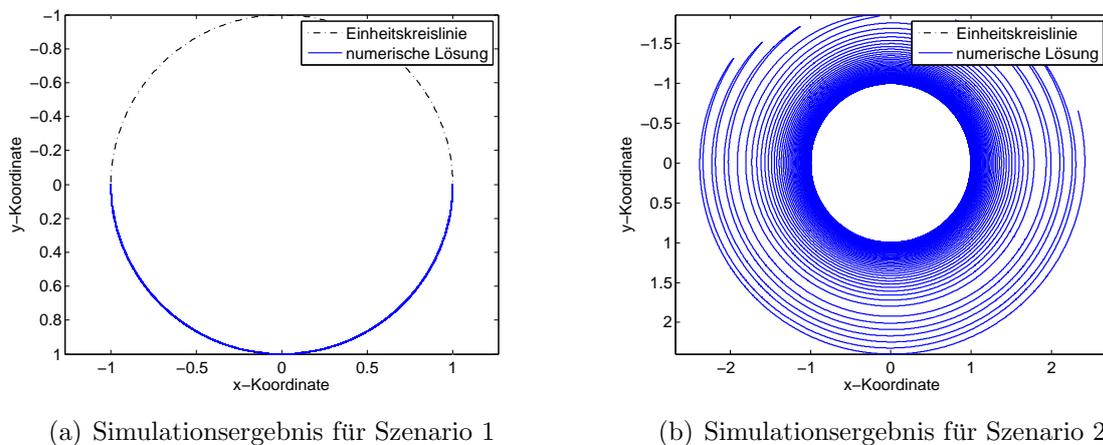


Abbildung 5.6: Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) von der Simulation der Index-0-Formulierung mit der Trapezregel und Endzeit 100

Für Szenario 1 bleibt die numerische Lösung auf der Einheitskreislinie. Für Szenario 2 driftet die numerische Lösung nach außen weg. Dieses Verhalten ist auch in Abbildung 5.7 zu beobachten. Der Fehler bei der Simulation von Szenario 1 bleibt beschränkt,

wohingegen er bei Szenario 2 steigt. Daher ist die Trapezregel zum Lösen der gegebenen DAE nur für bestimmte Anfangswerte geeignet und deshalb nicht als allgemeine Lösungsmethode zu bevorzugen.

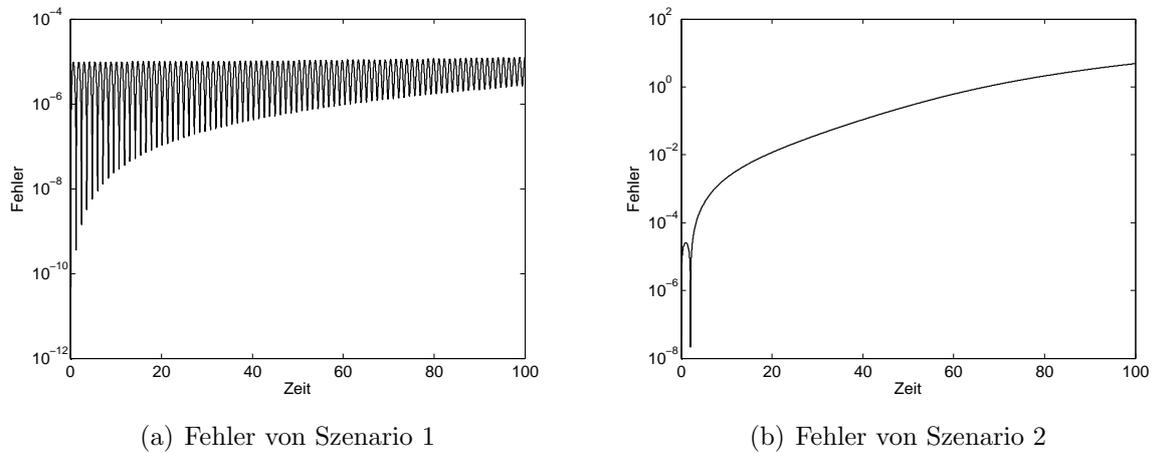


Abbildung 5.7: Fehler der numerischen Lösung zur Einheitskreislinie von der Simulation der Index-0-Formulierung mit der Trapezregel und Endzeit 100

Der ode-Solver `ode23t` liefert wie bereits erwähnt Ergebnisse, die von der Lösungsmanigfaltigkeit wegdriften. Da dieser Solver auf der Trapezregel beruht, ist das im ersten Moment überraschend, aber es wird vermutet, dass das Versagen des Solvers im Vergleich zur Trapezregel an der Schrittweitensteuerung liegt.

5.3.3 Simulationen der Index-1-Formulierung

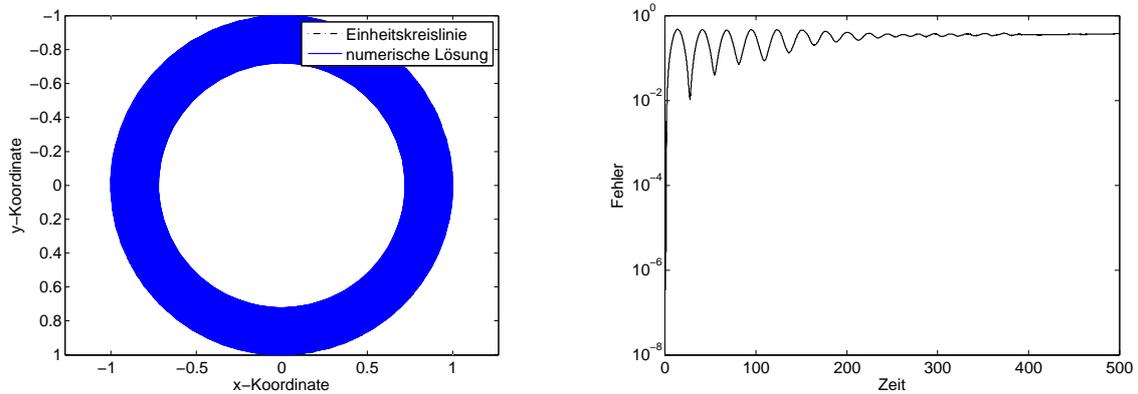
Für die Simulationen des Gleichungssystems der Index-1-Formulierung in Tabelle 5.1 wurden die in Tabelle 5.3 aufgelisteten Werte verwendet.

Anfangswerte				
	x	y	v_x	v_y
Szenario 1	1	0	0	0
Szenario 2	1	0	0	10

Tabelle 5.3: Anfangswerte für die Simulation der Bewegungsgleichungen des Pendels mit Index 1

Bei Verwendung von `ode45`, `ode15s` und explizitem Euler-Verfahren driftet die numerische Lösung nach außen von der Einheitskreislinie weg. Bei Verwendung des impliziten Euler-Verfahrens bewegt sich die numerische Lösung zuerst in das Innere der Einheitskreislinie und driftet dann nach außen weg.

In Abbildung 5.8 ist das Ergebnis der Simulation von Szenario 2 und Endzeit 500 mit dem ode-Solver `ode23t` dargestellt.

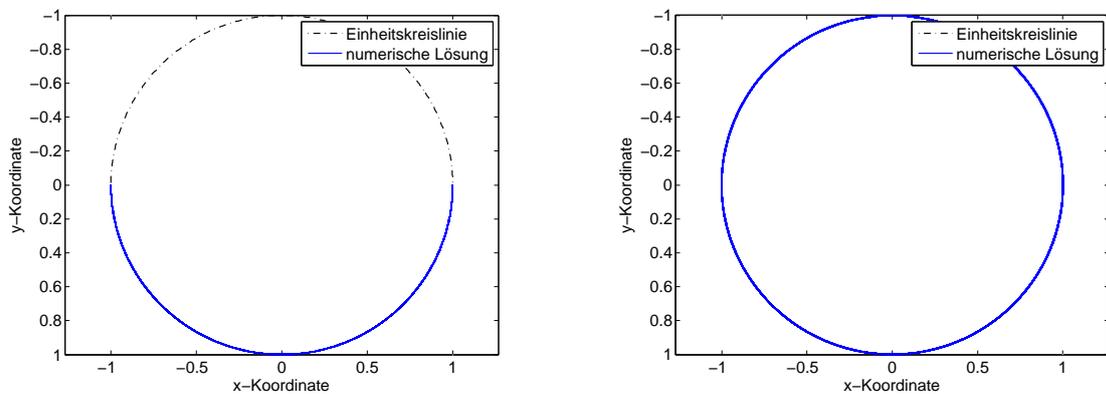


(a) Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) (b) Fehler der numerischen Lösung zur Einheitskreislinie

Abbildung 5.8: Szenario 2 der Index-1-Formulierung mit `ode23t` und Endzeit 500

Im Gegensatz zu den anderen Fällen läuft die Lösung nicht ins Unendliche und auch nicht in den Ursprung. Der Fehler pendelt sich auf einem konstanten Wert ein.

In Abbildung 5.9 ist das Ergebnis der Simulation von Szenario 1 und Szenario 2 mit der Trapezregel dargestellt.



(a) Simulationsergebnis von Szenario 1

(b) Simulationsergebnis von Szenario 2

Abbildung 5.9: Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) der Index-1-Formulierung mit der Trapezregel und Endzeit 100

Die Trapezregel liefert numerische Ergebnisse, die nahe an der Einheitskreislinie bleiben. In Abbildung 5.10 ist der Fehler der Simulationen mit der Trapezregel dargestellt.

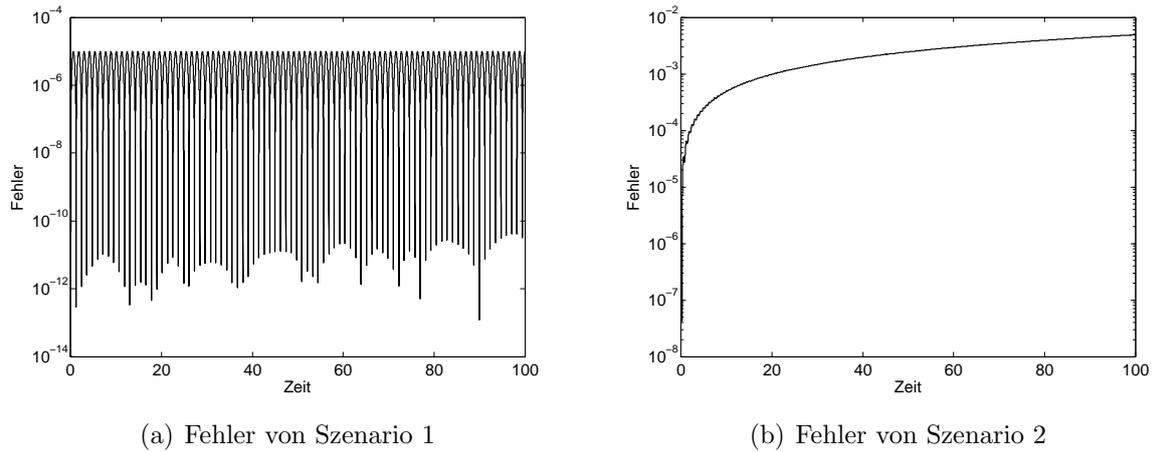


Abbildung 5.10: Fehler der numerischen Lösung zur Einheitskreislinie der Index-1-Formulierung mit der Trapezregel und Endzeit 100

Im Vergleich zum Fehler bei der Index-0-Formulierung, siehe Abbildung 5.7, ist zu erkennen, dass für Szenario 2 der Fehler ebenfalls steigt, allerdings langsamer.

5.3.4 Simulationen der Indexreduktionsmethoden

Im Folgenden werden die Lösungen der Simulationen mit den verschiedenen Indexreduktionsmethoden präsentiert.

5.3.4.1 Differentiation der Zwangsbedingung

Diese Vorgehensweise führt auf das Gleichungssystem (5.1)–(5.4) mit der Zwangsbedingung

$$v_x^2 + v_y^2 - F(x^2 + y^2) + \mathbf{g}y = 0. \quad (5.90)$$

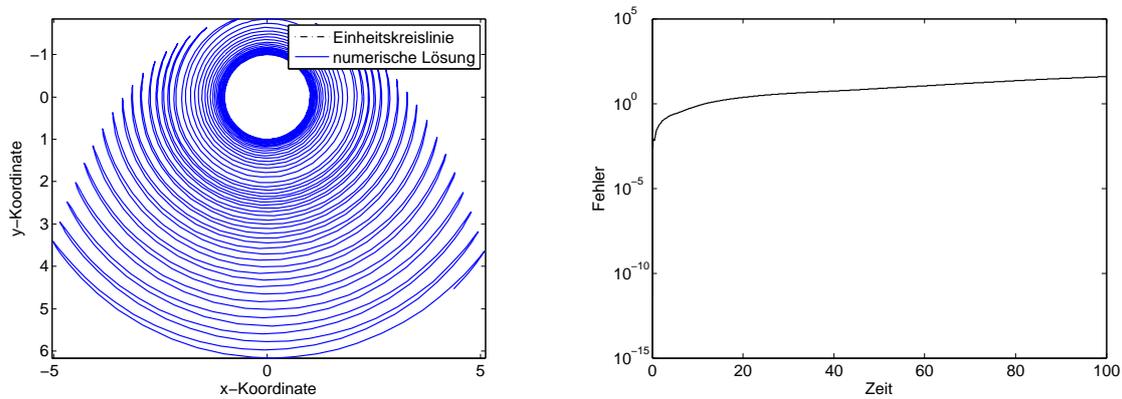
Dieses Gleichungssystem kann mit bestimmten ode-Solvern von MATLAB gelöst werden. Es wurde der `ode15s` verwendet, da sich dieser für DAEs mit Index 1 gut eignet.

Es wurden die in Tabelle 5.4 aufgelisteten Werte für die Simulationen verwendet.

Anfangswerte					
	x	y	v_x	v_y	F
Szenario 1	1	0	0	0	0
Szenario 2	1	0	0	10	100

Tabelle 5.4: Anfangswerte für die Simulation der Bewegungsgleichungen des Pendels mit der Indexreduktionsmethode Differentiation der Zwangsbedingung

In Abbildung 5.11 ist das Ergebnis der Simulation von Szenario 2 mit Endzeit 100 dargestellt.



(a) Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) (b) Fehler der numerischen Lösung zur Einheitskreislinie

Abbildung 5.11: Szenario 2 mit Differentiation der Zwangsbedingung, ode15s und Endzeit 100

Es ist zu erkennen, dass die numerische Lösung von der Einheitskreislinie wegdriftet.

5.3.4.2 Baumgarte–Methode

Die Baumgarte–Methode liefert die Gleichungen

$$\begin{aligned} \dot{x} &= v_x \\ \dot{y} &= v_y \\ \dot{v}_x &= -\frac{2(v_x^2 + v_y^2 + \mathbf{g}y) + 4\alpha(xv_x + yv_y) + \beta^2(x^2 + y^2 - 1)}{2(x^2 + y^2)}x \\ \dot{v}_y &= \mathbf{g} - \frac{2(v_x^2 + v_y^2 + \mathbf{g}y) + 4\alpha(xv_x + yv_y) + \beta^2(x^2 + y^2 - 1)}{2(x^2 + y^2)}y. \end{aligned} \quad (5.91)$$

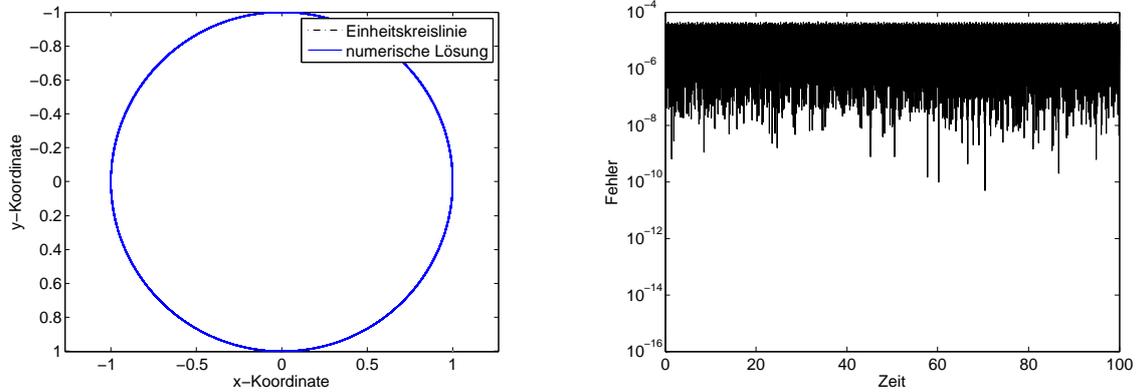
Es wurden die in Tabelle 5.5 aufgelisteten Werte für die Simulationen verwendet.

Anfangswerte						
	x	y	v_x	v_y	α	β
Szenario 1	1	0	0	0	10	10 bzw. 100
Szenario 2	1	0	0	10	10	10 bzw. 100

Tabelle 5.5: Anfangswerte für die Simulation der Bewegungsgleichungen des Pendels mit der Baumgarte–Methode

Für jedes Szenario werden zwei verschiedene Konstellationen für die Parameter α und β getestet.

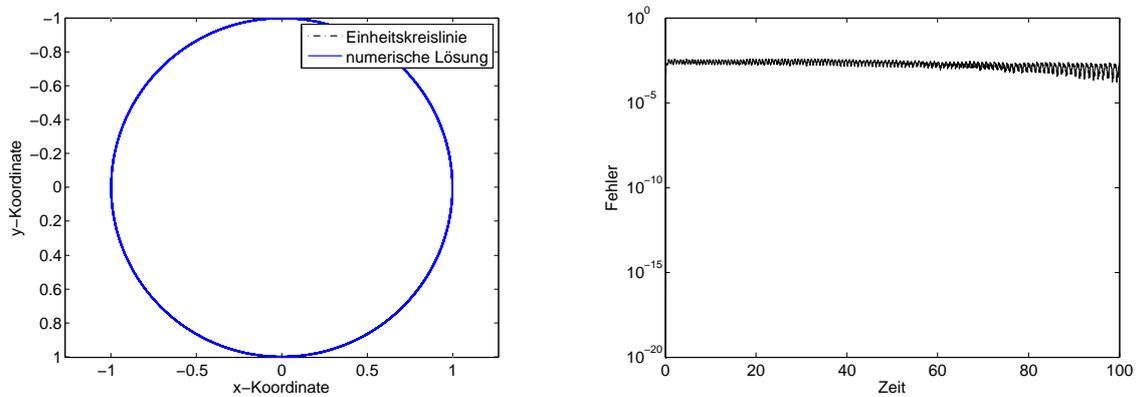
In Abbildung 5.12 ist das Ergebnis der Simulation mit der Baumgarte-Methode mit $\alpha = 10$, $\beta = 100$ und dem ode-Solver `ode45` dargestellt.



(a) Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) (b) Fehler der numerischen Lösung zur Einheitskreislinie

Abbildung 5.12: Szenario 2 mit der Baumgarte-Methode mit $\alpha \neq \beta$ und Endzeit 100

In Abbildung 5.13 ist das Ergebnis der Simulation mit der Baumgarte-Methode mit $\alpha = \beta = 10$ und dem ode-Solver `ode45` dargestellt.



(a) Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) (b) Fehler der numerischen Lösung zur Einheitskreislinie

Abbildung 5.13: Szenario 2 mit der Baumgarte-Methode mit $\alpha = \beta$ und Endzeit 100

Der Fehler für $\alpha \neq \beta$ ist für beide Szenarien kleiner als der Fehler für $\alpha = \beta$.

5.3.4.3 Pantelides–Algorithmus

Der Pantelides–Algorithmus führt auf vier Gleichungssysteme, wobei diese Gleichungssysteme in entsprechenden Bereichen (siehe Abbildung 5.14) verwendet werden. Diese Gleichungssysteme sind in Tabelle 5.6 dargestellt.

<p><u>Bereich 1</u></p> $\dot{x} = v_x$ $y = \sqrt{1 - x^2}$ $dy = -\frac{x}{y}\dot{x}$ $dv_y = d2y$ $F = -\frac{dv_y - \mathbf{g}}{y}$ $\dot{v}_x = -Fx$ $d2x = \dot{v}_x$ $d2y = -\frac{(dy)^2 + (\dot{x})^2 + xd2x}{y}$ $v_y = dy$	<p><u>Bereich 2</u></p> $\dot{x} = v_x$ $y = -\sqrt{1 - x^2}$ $dy = -\frac{x}{y}\dot{x}$ $dv_y = d2y$ $F = -\frac{dv_y - \mathbf{g}}{y}$ $\dot{v}_x = -Fx$ $d2x = \dot{v}_x$ $d2y = -\frac{(dy)^2 + (\dot{x})^2 + xd2x}{y}$ $v_y = dy$
<p><u>Bereich 3</u></p> $\dot{y} = v_y$ $x = \sqrt{1 - y^2}$ $dx = -\frac{y}{x}\dot{y}$ $dv_x = d2x$ $F = -\frac{dv_x}{x}$ $\dot{v}_y = \mathbf{g} - Fy$ $d2y = \dot{v}_y$ $d2x = -\frac{(dx)^2 + (\dot{y})^2 + yd2y}{x}$ $v_x = dx$	<p><u>Bereich 4</u></p> $\dot{y} = v_y$ $x = -\sqrt{1 - y^2}$ $dx = -\frac{y}{x}\dot{y}$ $dv_x = d2x$ $F = -\frac{dv_x}{x}$ $\dot{v}_y = \mathbf{g} - Fy$ $d2y = \dot{v}_y$ $d2x = -\frac{(dx)^2 + (\dot{y})^2 + yd2y}{x}$ $v_x = dx$

Tabelle 5.6: Die vier erhaltenen Gleichungssysteme bei Verwendung des Pantelides–Algorithmus

Die in Tabelle 5.6 aufgelisteten Gleichungssysteme müssen für die Implementierung in MATLAB erweitert werden, da der Zustandsraum des Gesamtsystems nicht neundimensional, sondern elfdimensional ist. Die beiden Zustandsgrößen dy und dv_y , die in den Bereichen 1 und 2 auftreten, sind in den Bereichen 3 und 4 nicht vorhanden. Genauso verhält es sich für die Zustandsgrößen dx und dv_x , die in den Bereichen 3 und 4 auftreten, jedoch nicht in Bereich 1 und 2. Deshalb wird der jeweils neundimensionale Zustandsraum um die beiden fehlenden Zustandsgrößen erweitert. Aufgrund der Erweiterung des

Zustandsraumes müssen auch für jeden Bereich zwei neue Gleichungen eingeführt werden. Für den Bereich 1 und 2 lauten diese Gleichungen $v_x = dx$ und $dv_x = d2x$ und für die Bereiche 3 und 4 $v_y = dy$ und $dv_y = d2y$.

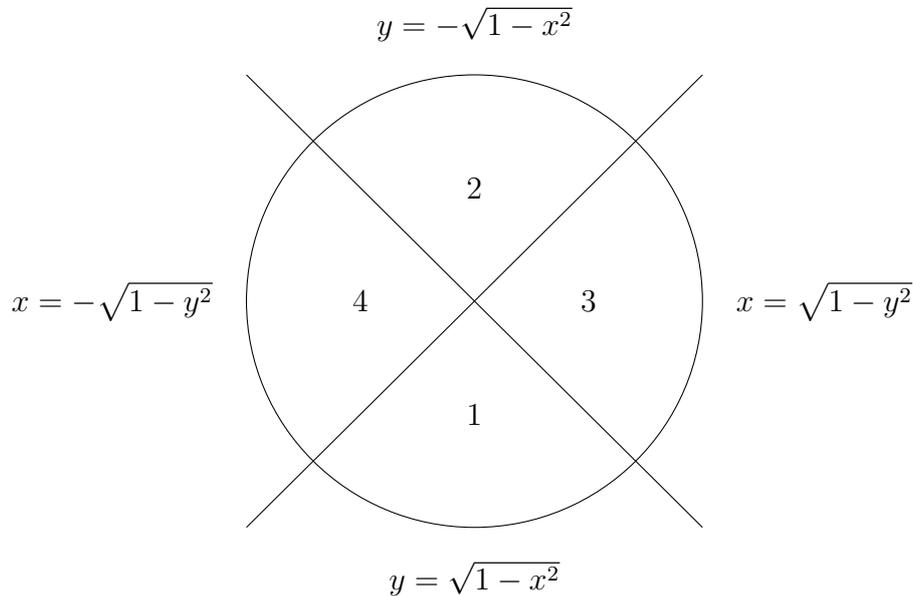


Abbildung 5.14: Einheitskreislinie mit den Bereichen für die entsprechenden Gleichungssysteme des Pantelides-Algorithmus

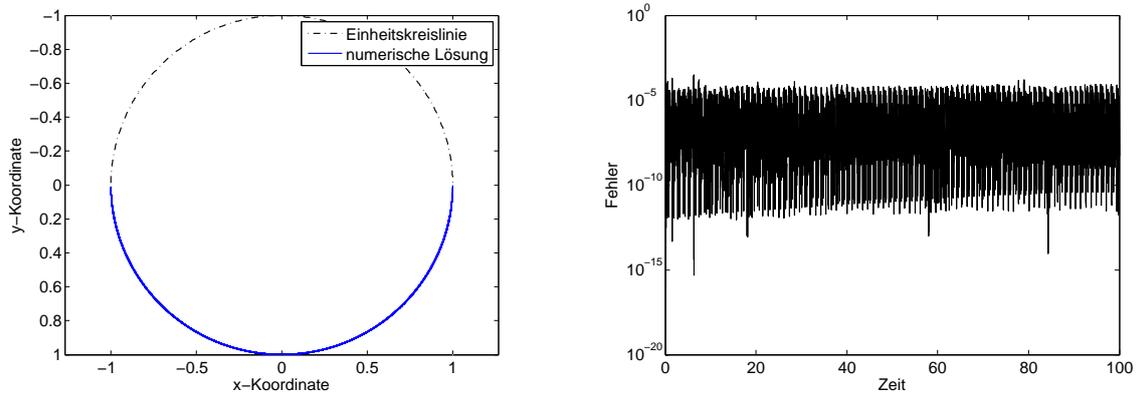
Die numerische Umsetzung des Pantelides-Algorithmus wird mithilfe eines Events in MATLAB durchgeführt. Dieses Event detektiert das Ereignis $|y| - |x| = 0$ und im Fall des Eintretens des Ereignisses wird die Integration gestoppt. Anschließend werden die `if`-Abfragen $|x| > |y|$ und $|y| > |x|$ verwendet und innerhalb dieser wird getrennt abgeprüft, ob x und y positiv oder negativ sind. Durch diese `if`-Abfragen wird das jeweils zu verwendende Gleichungssystem bestimmt und dieses wird dann mit dem ode-Solver `ode15i` gelöst, wobei die maximale Ordnung des Verfahrens auf 2 gesetzt wurde, das heißt `MaxOrder` wurde auf 2 gesetzt.

Es wurden die in Tabelle 5.7 aufgelisteten Werte für die Simulationen verwendet.

Anfangswerte											
	x	$d2x$	y	dy	$d2y$	v_x	v_y	dv_y	F	dx	dv_x
Szenario 1	1	0	0	0	g	0	0	g	0	0	0
Szenario 2	1	-100	0	10	g	0	10	g	100	0	-100

Tabelle 5.7: Anfangswerte für die Simulation der Bewegungsgleichungen des Pendels mit dem Pantelides-Algorithmus

In Abbildung 5.15 ist das Ergebnis der Simulation mit dem Pantelides–Algorithmus und Endzeit 100 dargestellt.

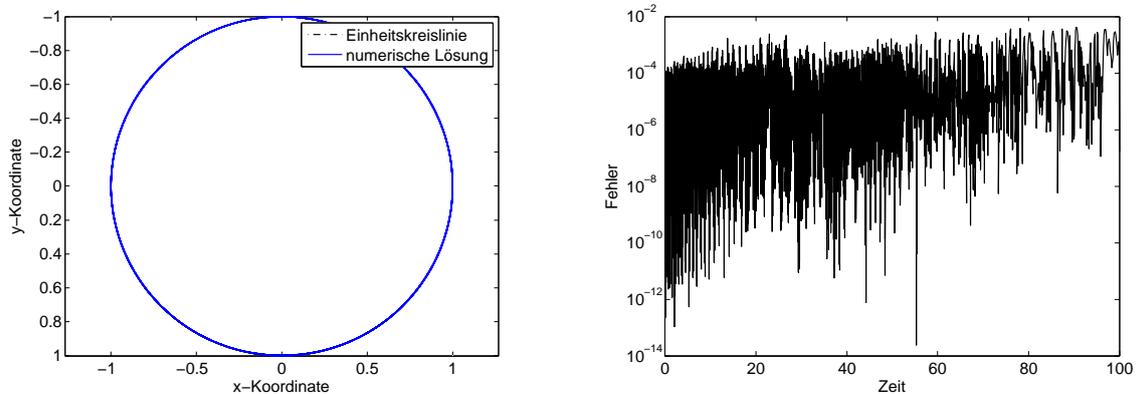


(a) Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) (b) Fehler der numerischen Lösung zur Einheitskreislinie

Abbildung 5.15: Szenario 1 mit dem Pantelides–Algorithmus und Endzeit 100

Für diese Simulation wurde die absolute Toleranz auf 10^{-2} gesetzt.

In Abbildung 5.16 ist das Ergebnis der Simulation mit dem Pantelides–Algorithmus und Endzeit 100 dargestellt.



(a) Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) (b) Fehler der numerischen Lösung zur Einheitskreislinie

Abbildung 5.16: Szenario 2 mit dem Pantelides–Algorithmus und Endzeit 100

Bei dieser Simulation wurde für die absolute Toleranz 10^{-1} gewählt. Wenn die absolute Toleranz auf 10^{-2} gesetzt wird, bricht der ode–Solver vor Erreichen der Endzeit der Simulation ab.

5.3.4.4 Standardprojektionsmethode

Die verwendete orthogonale Projektion $p : \mathbb{R}^4 \rightarrow M$ wird definiert als

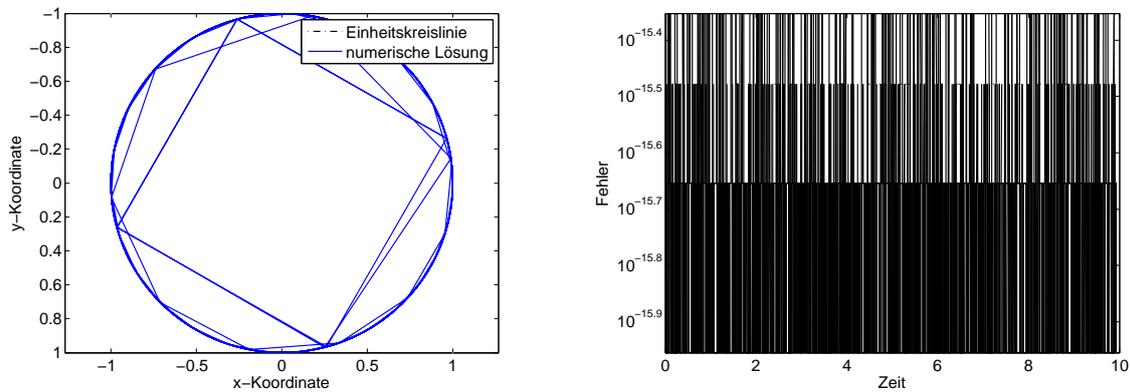
$$p \begin{pmatrix} x \\ y \\ v_x \\ v_y \end{pmatrix} = \begin{pmatrix} \frac{x}{\sqrt{x^2 + y^2}} \\ \frac{y}{\sqrt{x^2 + y^2}} \\ (-\pi_2 \circ p_1((x, y)^T)v_x + \pi_1 \circ p_1((x, y)^T)v_y)(-\pi_2 \circ p_1((x, y)^T)) \\ (-\pi_2 \circ p_1((x, y)^T)v_x + \pi_1 \circ p_1((x, y)^T)v_y)\pi_1 \circ p_1((x, y)^T) \end{pmatrix}. \quad (5.92)$$

Es wurden die in Tabelle 5.8 aufgelisteten Werte für die Simulationen verwendet.

Anfangswerte				
	x	y	v_x	v_y
Szenario 1	1	0	0	0
Szenario 2	1	0	0	10

Tabelle 5.8: Anfangswerte für die Simulation der Bewegungsgleichungen des Pendels mit der orthogonalen Projektionsmethode

In Abbildung 5.17 ist das Ergebnis der Simulation mit der orthogonalen Projektionsmethode und dem expliziten Euler-Verfahren dargestellt.



(a) Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) (b) Fehler der numerischen Lösung zur Einheitskreislinie

Abbildung 5.17: Szenario 2 mit der orthogonalen Projektionsmethode und Endzeit 100

Wie an der Abbildung des Fehlers zu erkennen ist, bleibt die numerische Lösung auf der Lösungsmannigfaltigkeit, allerdings ist an der Trajektorie zu sehen, dass die einzelnen Punkte der numerischen Lösung immer weiter voneinander entfernt sind, sodass teilweise sogar ein Viertelkreis von einem Punkt zum nächsten übersprungen wird. Um diesem

Phänomen auf den Grund zu gehen, wird in Abbildung 5.18 die Norm der Geschwindigkeit $\|(v_x, v_y)^T\|_2$ betrachtet.

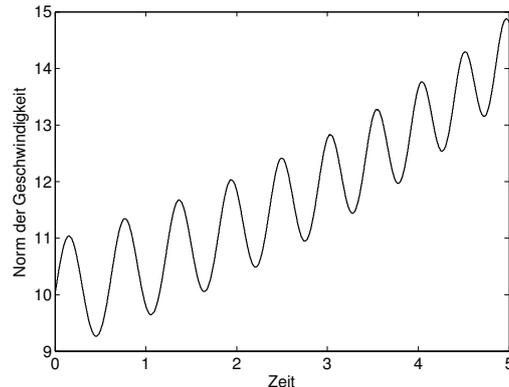
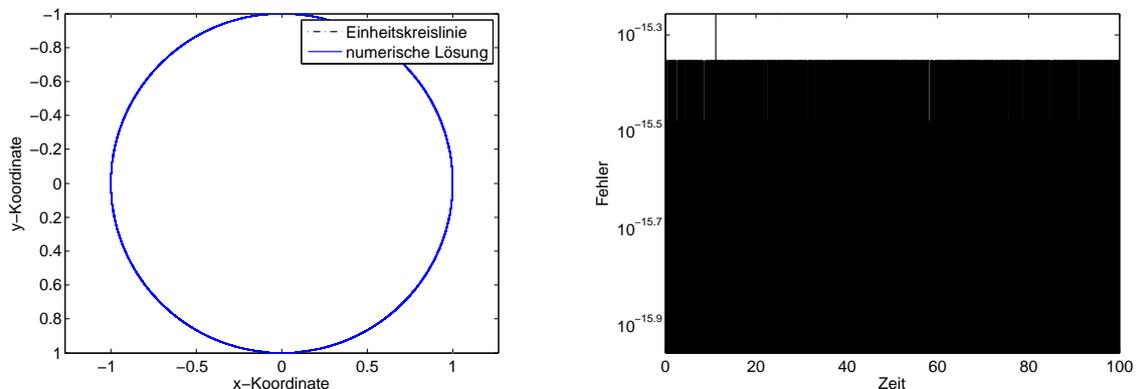


Abbildung 5.18: Norm der Geschwindigkeit des Pendels von Szenario 2 bei der orthogonalen Projektionsmethode mit Endzeit 5

Wie zu erkennen ist, steigt die Norm der Geschwindigkeit in den ersten 5 Sekunden schon stark an. Der Anstieg ist so enorm, dass die numerische Lösung bei 10 Sekunden nur noch aus NaN-Werten besteht.

In Abbildung 5.19 ist das Ergebnis der Simulation mit der orthogonalen Projektionsmethode und dem expliziten Euler-Verfahren dargestellt.



(a) Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) (b) Fehler der numerischen Lösung zur Einheitskreislinie

Abbildung 5.19: Szenario 1 mit der orthogonalen Projektionsmethode und Endzeit 100

Die in Abbildung 5.19 (a) erkennbare Position des Pendels liegt zwar auf der Kreisbahn, allerdings nicht an der richtigen Position. Der Grund dafür ist die immer größer werdende Norm der Geschwindigkeit, die in Abbildung 5.20 dargestellt ist.

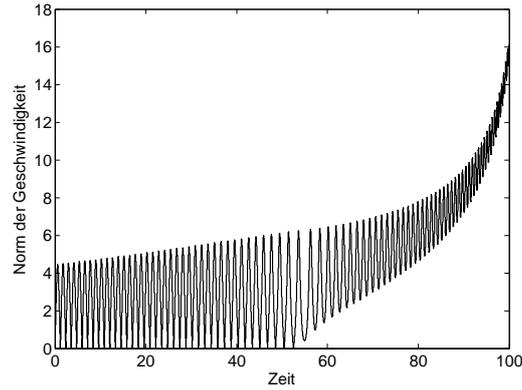


Abbildung 5.20: Norm der Geschwindigkeit des Pendels von Szenario 1 bei der orthogonalen Projektionsmethode mit Endzeit 100

5.3.4.5 Symmetrische Projektionsmethode

Für das zu lösende Gleichungssystem (4.9) ergibt sich

$$\begin{pmatrix} 1 & 0 & 0 & 0 & -4x_{n+1} & -2v_{x_{n+1}} \\ 0 & 1 & 0 & 0 & -4y_{n+1} & -2v_{y_{n+1}} \\ 0 & 0 & 1 & 0 & 0 & -2x_{n+1} \\ 0 & 0 & 0 & 1 & 0 & -2y_{n+1} \\ 2x_{n+1} & 2y_{n+1} & 0 & 0 & 0 & 0 \\ v_{x_{n+1}} & v_{y_{n+1}} & x_{n+1} & y_{n+1} & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta x_{n+1} \\ \Delta y_{n+1} \\ \Delta v_{x_{n+1}} \\ \Delta v_{y_{n+1}} \\ \Delta \mu_1 \\ \Delta \mu_2 \end{pmatrix} = \begin{pmatrix} \hat{x}_n + h\Psi(h, \hat{x}_n, \hat{x}_{n+1}) - \hat{x}_{n+1} \\ \hat{y}_n + h\Psi(h, \hat{y}_n, \hat{y}_{n+1}) - \hat{y}_{n+1} \\ \hat{v}_{x_n} + h\Psi(h, \hat{v}_{x_n}, \hat{v}_{x_{n+1}}) - \hat{v}_{x_{n+1}} \\ \hat{v}_{y_n} + h\Psi(h, \hat{v}_{y_n}, \hat{v}_{y_{n+1}}) - \hat{v}_{y_{n+1}} \\ x_{n+1}^2 + y_{n+1}^2 - 1 \\ x_{n+1}v_{x_{n+1}} + y_{n+1}v_{y_{n+1}} \end{pmatrix}, \quad (5.93)$$

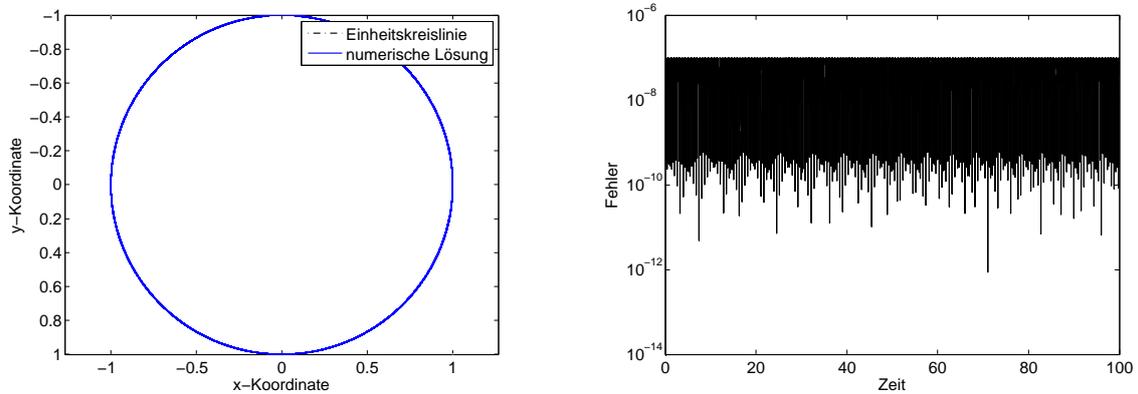
wobei Ψ die Inkrementfunktion der Trapezregel ist.

Es wurden die in Tabelle 5.9 aufgelisteten Werte für die Simulationen verwendet.

Anfangswerte				
	x	y	v_x	v_y
Szenario 1	1	0	0	0
Szenario 2	1	0	0	10

Tabelle 5.9: Anfangswerte für die Simulation der Bewegungsgleichungen des Pendels mit der symmetrischen Projektionsmethode

In Abbildung 5.21 ist das Ergebnis der Simulation mit der symmetrischen Projektionsmethode und der Trapezregel dargestellt.



(a) Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) (b) Fehler der numerischen Lösung zur Einheitskreislinie

Abbildung 5.21: Szenario 2 mit der symmetrischen Projektionsmethode und Endzeit 100

In Abbildung 5.22 ist die Norm der Geschwindigkeit des Pendels dargestellt. Im Vergleich zur Norm der Geschwindigkeit bei der orthogonalen Projektionsmethode (Abbildung 5.20) ist hier die Norm der Geschwindigkeit beschränkt.

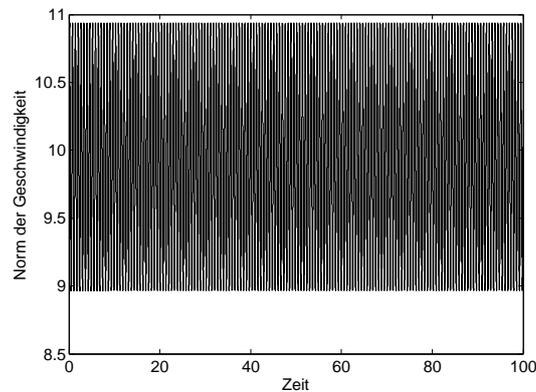


Abbildung 5.22: Norm der Geschwindigkeit des Pendels von Szenario 2 bei der symmetrischen Projektionsmethode mit Endzeit 100

5.3.4.6 Methoden basierend auf lokaler Zustandsraumtransformation

Diese Methode führt auf das folgende Differentialgleichungssystem 1. Ordnung

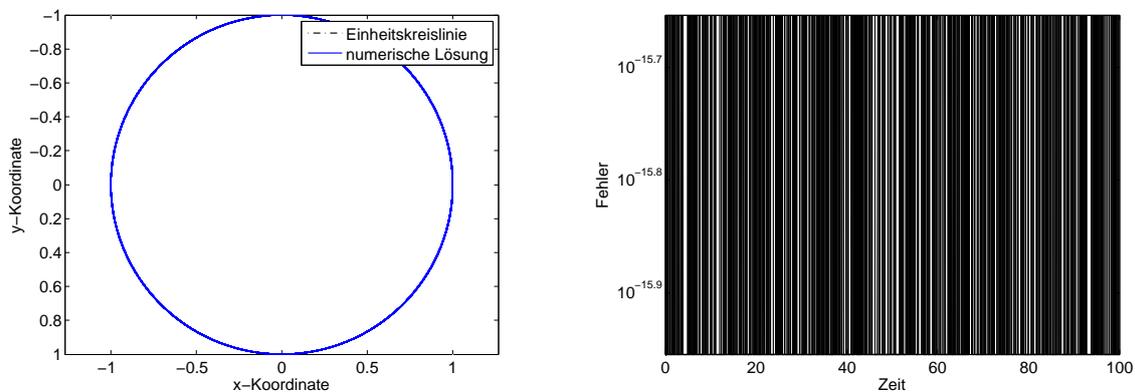
$$\begin{aligned} \dot{\varphi} &= \eta \\ \dot{\eta} &= \mathbf{g} \cos \varphi. \end{aligned} \tag{5.94}$$

Es wurden die in Tabelle 5.10 aufgelisteten Werte für die Simulationen verwendet.

Anfangswerte		
	φ	$\dot{\varphi}$
Szenario 1	0	0
Szenario 2	0	10

Tabelle 5.10: Anfangswerte für die Simulation der Bewegungsgleichungen des Pendels mit Zustandsraumtransformation

In Abbildung 5.23 ist das Ergebnis der Simulation mit Zustandsraumtransformation und dem ode-Solver `ode45` dargestellt.



(a) Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) (b) Fehler der numerischen Lösung zur Einheitskreislinie

Abbildung 5.23: Szenario 2 mit der Zustandsraumtransformation und Endzeit 100

Diese Methode ist sehr einfach mittels des ode-Solvers `ode45` zu implementieren. Wie in Abbildung 5.23 zu erkennen ist, ist die Abweichung zur Einheitskreislinie gering. Daher ist diese Methode gut geeignet um die gegebene DAE zu lösen.

5.3.5 Ergebnisse

Im Folgenden werden für die verschiedenen Formulierungen des Systems (Index 0 bzw. Index 1) und die Indexreduktionsmethoden Ergebnisse des maximalen Fehlers und der Laufzeit in Tabellen aufgelistet. Dabei bezeichnet ein Eintrag mit *, dass der im Programm verwendete ode-Solver nicht bis zur verlangten Endzeit der Simulation lauffähig war.

In Tabelle 5.11 sind der maximale Fehler e und die Laufzeit in Sekunden der Simulationen bis $T = 10$ der Index-0-Formulierung aufgelistet.

Lösungsmethode	Szenario 1		Szenario 2	
	Max. Fehler	Zeit(s)	Max. Fehler	Zeit(s)
ode45	0.5135	0.490	*	*
ode15s	*	*	180.3508	0.951
ode23t	*	*	*	*
explizites Euler–Verfahren	0.2390	2.155	$8.0874 \cdot 10^4$	2.016
implizites Euler–Verfahren	0.3044	8.622	9.4571	8.484
Trapezregel	$9.8303 \cdot 10^{-6}$	9.489	0.0021	10.243

Tabelle 5.11: Maximaler Fehler e und Laufzeit in Sekunden der Index–0–Formulierung mit Endzeit 10

Wie in Tabelle 5.11 zu erkennen ist, führen alle Verfahren außer die Trapezregel zu einem numerischen „Drift–off“. Der ode–Solver `ode23t` bricht die Integration vor der Endzeit der Simulation ab. Die beste Performance hinsichtlich der Abweichung zur Kreisbahn liefert die Trapezregel. Die höheren Laufzeiten des expliziten Euler–Verfahrens, impliziten Euler–Verfahrens und der Trapezregel sind darauf zurückzuführen, dass sie selbst implementiert sind und die beiden impliziten Solver auch noch ein selbst implementiertes Newton–Verfahren verwenden.

In Tabelle 5.12 sind der maximale Fehler e und die Laufzeit in Sekunden der Simulationen bis $T = 100$ der Index–0–Formulierung aufgelistet.

Lösungsmethode	Szenario 1		Szenario 2	
	Max. Fehler	Zeit(s)	Max. Fehler	Zeit(s)
ode45	$1.3554 \cdot 10^3$	0.522	*	*
ode15s	*	*	$2.9945 \cdot 10^4$	0.983
ode23t	*	*	*	*
explizites Euler–Verfahren	1.6493	95.434	Inf	94.540
implizites Euler–Verfahren	$2.0944 \cdot 10^6$	480.712	796.6821	313.994
Trapezregel	$1.2466 \cdot 10^{-5}$	315.038	4.8617	314.108

Tabelle 5.12: Maximaler Fehler e und Laufzeit in Sekunden der Index–0–Formulierung mit Endzeit 100

Für die Ergebnisse in Tabelle 5.12 gilt grundsätzlich dasselbe wie für die Ergebnisse von Tabelle 5.11. Es ist ersichtlich, dass bei längerer Simulation die Trapezregel für Szenario 2 auch versagt.

In Tabelle 5.13 sind der maximale Fehler e und die Laufzeit in Sekunden der Simulationen bis $T = 10$ der Index–1–Formulierung aufgelistet.

Lösungsmethode	Szenario 1		Szenario 2	
	Max. Fehler	Zeit(s)	Max. Fehler	Zeit(s)
ode45	0.4502	0.489	0.9111	0.504
ode15s	0.3083	0.690	0.5043	0.855
ode23t	0.2616	0.754	0.3626	1.271
explizites Euler-Verfahren	0.1736	1.330	1.6893	1.979
implizites Euler-Verfahren	0.1836	5.549	0.6390	6.259
Trapezregel	$9.8100 \cdot 10^{-6}$	6.028	$4.9976 \cdot 10^{-4}$	6.945

Tabelle 5.13: Maximaler Fehler e und Laufzeit in Sekunden der Index-1-Formulierung mit Endzeit 10

In Tabelle 5.13 ist zu erkennen, dass alle Verfahren außer die Trapezregel zum numerischen „Drift-off“ führen.

In Tabelle 5.14 sind der maximale Fehler e und die Laufzeit in Sekunden der Simulationen bis $T = 100$ der Index-1-Formulierung aufgelistet.

Lösungsmethode	Szenario 1		Szenario 2	
	Max. Fehler	Zeit(s)	Max. Fehler	Zeit(s)
ode45	60.3818	0.533	155.5349	0.537
ode15s	20.4483	1.047	46.7038	1.430
ode23t	5.9062	1.619	0.4785	5.045
explizites Euler-Verfahren	1.6214	75.895	$2.6402 \cdot 10^4$	80.111
implizites Euler-Verfahren	60.8834	244.095	12.6511	246.310
Trapezregel	$9.8100 \cdot 10^{-6}$	257.664	0.0049	255.583

Tabelle 5.14: Maximaler Fehler e und Laufzeit in Sekunden der Index-1-Formulierung mit Endzeit 100

Wie in Tabelle 5.14 abzulesen ist, ist bei Verwendung der Trapezregel auch bei längerer Simulation der Abstand zur Kreisbahn gering.

Bei den Simulationen der Index-0-Formulierung und der Index-1-Formulierung tritt bei allen Methoden (außer der Trapezregel) der numerische „Drift-off“ auf. Bei der Trapezregel hingegen bleibt die numerische Lösung in einigen Fällen auf der Kreisbahn. Allerdings tritt bei der Index-0-Formulierung von Szenario 2 ebenfalls der numerische „Drift-off“ auf.

In Tabelle 5.15 sind der maximale Fehler e und die Laufzeit in Sekunden der Simulationen bis $T = 10$ der Indexreduktionsmethoden aufgelistet.

Lösungsmethode	Szenario 1		Szenario 2	
	Max. Fehler	Zeit(s)	Max. Fehler	Zeit(s)
Differentiation Zwangsbedingung	0.3051	0.614	0.7244	2.447
Baumgarte-Methode $\alpha \neq \beta$	$1.8949 \cdot 10^{-5}$	0.557	$4.4292 \cdot 10^{-5}$	1.869
Baumgarte-Methode $\alpha = \beta$	$2.4634 \cdot 10^{-4}$	0.452	0.0038	0.554
Pantelides	$3.3449 \cdot 10^{-4}$	1.396	$5.1756 \cdot 10^{-4}$	4.195
orth. Projektionsmethode	$4.4409 \cdot 10^{-16}$	1.659	$4.4409 \cdot 10^{-16}$	1.700
symm. Projektionsmethode	$2.7006 \cdot 10^{-8}$	7.920	$1.0115 \cdot 10^{-7}$	7.841
Koordinatentransformation	$2.2204 \cdot 10^{-16}$	0.448	$2.2204 \cdot 10^{-16}$	0.631

Tabelle 5.15: Maximaler Fehler e und Laufzeit in Sekunden der Indexreduktionsmethoden mit Endzeit 10

In Tabelle 5.15 ist zu erkennen, dass die Differentiation der Zwangsbedingung alleine nicht ausreicht, um den numerischen „Drift-off“ zu verhindern. Die orthogonale Projektionsmethode liefert leider keine richtigen Ergebnisse in dem Sinn, dass die Position des Pendels zwar auf der Kreisbahn bleibt, aber aufgrund der fälschlicherweise ansteigenden Norm der Geschwindigkeit dennoch nicht korrekt ist. Die anderen Methoden liefern durchwegs geringe Abweichungen zur Kreisbahn. Bei der Baumgarte-Methode führt die Wahl $\alpha \neq \beta$ zu besseren Ergebnissen als $\alpha = \beta$. Den geringsten Fehler liefert die Koordinatentransformation.

In Tabelle 5.16 sind der maximale Fehler e und die Laufzeit in Sekunden der Simulationen bis $T = 100$ der Indexreduktionsmethoden aufgelistet.

Lösungsmethode	Szenario 1		Szenario 2	
	Max. Fehler	Zeit(s)	Max. Fehler	Zeit(s)
Differentiation Zwangsbedingung	24.7310	1.095	38.5859	2.848
Baumgarte-Methode $\alpha \neq \beta$	$1.9089 \cdot 10^{-5}$	1.696	$4.5584 \cdot 10^{-5}$	3.006
Baumgarte-Methode $\alpha = \beta$	$2.4634 \cdot 10^{-4}$	0.723	0.0038	0.890
Pantelides	$3.3449 \cdot 10^{-4}$	8.539	0.0042	5.916
orth. Projektionsmethode	$5.5511 \cdot 10^{-16}$	76.175	$4.4409 \cdot 10^{-16}$	76.268
symm. Projektionsmethode	$2.7006 \cdot 10^{-8}$	71.729	$1.0117 \cdot 10^{-7}$	71.057
Koordinatentransformation	$2.2204 \cdot 10^{-16}$	0.550	$2.2204 \cdot 10^{-16}$	1.770

Tabelle 5.16: Maximaler Fehler e und Laufzeit in Sekunden der Indexreduktionsmethoden mit Endzeit 100

Bei Simulation bis $T = 100$ bleiben die oben erwähnten Resultate gleich. Die größeren Laufzeiten der orthogonalen Projektionsmethode und der symmetrischen Projektionsme-

thode sind darauf zurückzuführen, dass bei der einen ein selbstimplementiertes explizites Euler–Verfahren verwendet wird und bei der anderen die numerische Lösung iterativ berechnet wird. Der Pantelides–Algorithmus besitzt aufgrund der Fallunterscheidungen eine längere Laufzeit als die Baumgarte–Methode.

Zusammenfassung der Ergebnisse:

Das Lösen der Index–0–Formulierung bzw. der Index–1–Formulierung liefert nur bei Verwendung der Trapezregel ein brauchbares Ergebnis, alle anderen Methoden führen zum „Drift–off“.

Die Differentiation der Zwangsbedingung führt zum „Drift–off“ und ist daher nicht geeignet für das Lösen der DAE. Die Baumgarte–Methode liefert für geeignetes α und β Resultate, die nahe an der Kreisbahn bleiben und es sind auch Unterschiede der Lösungen für verschiedene Werte der Parameter α und β zu beobachten. Die Implementierung des Pantelides–Algorithmus ist durch die Fallunterscheidungen etwas umständlich und der `ode15i` hat auch bei einigen Anfangswerten Probleme die Gleichungen zu lösen.

Die orthogonale Projektionsmethode liefert Ergebnisse, die ebenfalls nahe an der Kreisbahn sind, dies ist auch nicht überraschend, da die Methode genau so konstruiert ist. Allerdings ist die Position des Pendels nicht korrekt, was auf die steigende Norm der Geschwindigkeit zurückgeführt werden kann. Im Gegensatz dazu liefert die symmetrische Projektionsmethode die „richtige“ Position und Geschwindigkeit.

Die Transformation des Zustandsraumes führt auf das einfachste zu lösende System und liefert ein Ergebnis, das nahe an der Kreisbahn bleibt.

Nach Durchführen dieser Fallstudie ist daher die zu empfehlende Methode die Transformation auf Polarkoordinaten.

Methodenvergleich – Fallstudie Doppelpendel

In diesem Kapitel werden die verschiedenen Methoden aus Kapitel 4 anhand eines zweiten Beispiels miteinander verglichen. Als zweites Beispiel wird die Bewegung eines Doppelpendels in kartesischen Koordinaten verwendet. Diese ist in expliziter Form gegeben durch die folgenden Gleichungen und in Abbildung 6.1 schematisch illustriert. Der Einfachheit halber werden die Masse und die Länge des Seils beider Pendel mit eins angenommen. Da gewöhnlich sowohl die die Zwangsbedingung definierende Funktion als auch die Erdbeschleunigung mit g bezeichnet werden, wird hier die die Zwangsbedingung definierende Funktion mit g und die Erdbeschleunigung mit \mathbf{g} (fett) bezeichnet.

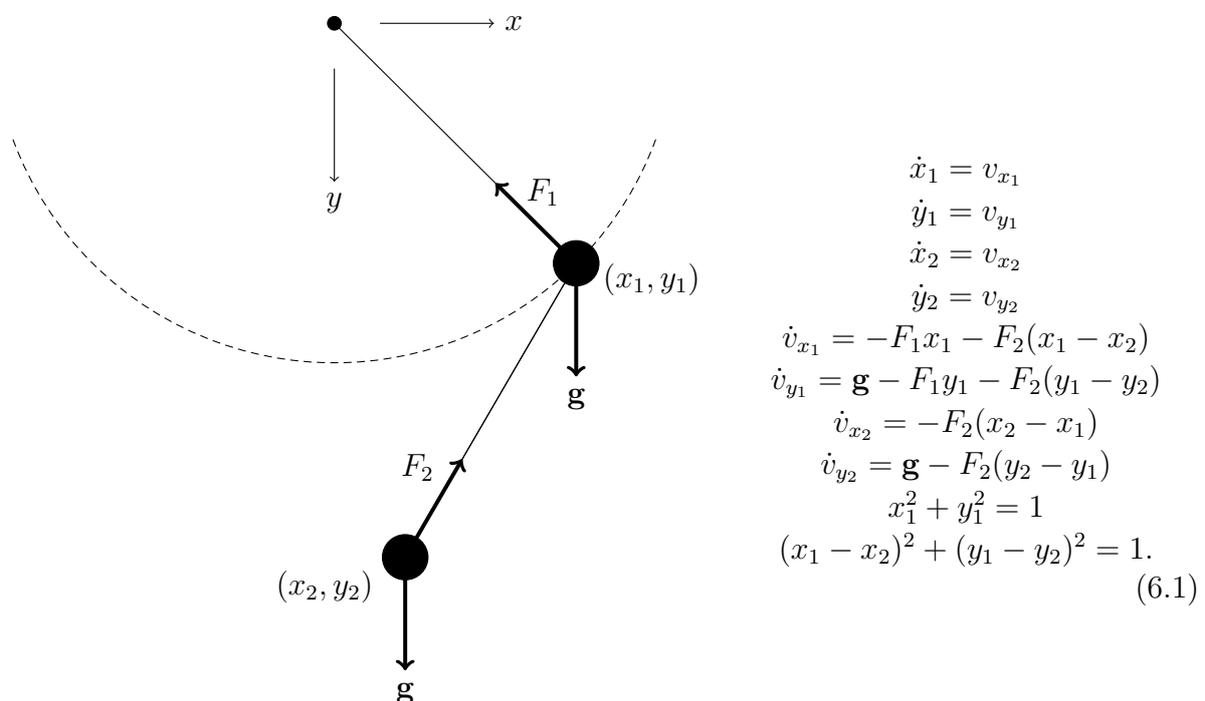


Abbildung 6.1: Schaubild und Gleichungen des Doppelpendels

Dabei bezeichnen F_1 und F_2 Kräfte und g die Erdbeschleunigung. Mathematisch gesehen können F_1 und F_2 als Lagrangesche Multiplikatoren interpretiert werden. Die Zwangsbedingung $g(x_1, y_1, x_2, y_2) = 0$ ist durch die Funktion $g: \mathbb{R}^4 \rightarrow \mathbb{R}^2$ folgendermaßen

$$g(x_1, y_1, x_2, y_2) = \begin{pmatrix} x_1^2 + y_1^2 - 1 \\ (x_1 - x_2)^2 + (y_1 - y_2)^2 - 1 \end{pmatrix} \quad (6.2)$$

gegeben. Im Folgenden werden die Bezeichnungen $g_1 = x_1^2 + y_1^2 - 1$ und $g_2 = (x_1 - x_2)^2 + (y_1 - y_2)^2 - 1$ verwendet. Die acht Differentialgleichungen erster Ordnung können umgeschrieben werden in vier Differentialgleichungen zweiter Ordnung:

$$\begin{aligned} \ddot{x}_1 &= -F_1 x_1 - F_2(x_1 - x_2) \\ \ddot{y}_1 &= \mathbf{g} - F_1 y_1 - F_2(y_1 - y_2) \\ \ddot{x}_2 &= -F_2(x_2 - x_1) \\ \ddot{y}_2 &= \mathbf{g} - F_2(y_2 - y_1) \\ x_1^2 + y_1^2 &= 1 \\ (x_1 - x_2)^2 + (y_1 - y_2)^2 &= 1. \end{aligned} \quad (6.3)$$

6.1 Analyse der DAE

Im Folgenden werden die beiden Varianten aus Abschnitt 3.4 diskutiert. Dazu werden zuerst die kinetische Energie E_{kin} und die potentielle Energie E_{pot} berechnet als

$$E_{kin} = \frac{\dot{x}_1^2 + \dot{y}_1^2}{2} + \frac{\dot{x}_2^2 + \dot{y}_2^2}{2} \quad \text{und} \quad E_{pot} = -\mathbf{g}y_1 - \mathbf{g}y_2. \quad (6.4)$$

- Variationsrechnung:

Die Funktion $L: \mathbb{R}^8 \rightarrow \mathbb{R}$ ist definiert als

$$L(x_1, y_1, \dot{x}_1, \dot{y}_1, x_2, y_2, \dot{x}_2, \dot{y}_2) = E_{kin} - E_{pot} = \frac{\dot{x}_1^2 + \dot{y}_1^2}{2} + \frac{\dot{x}_2^2 + \dot{y}_2^2}{2} + \mathbf{g}y_1 + \mathbf{g}y_2. \quad (6.5)$$

Die Euler–Lagrange Gleichungen lauten

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}_1} \right) - \frac{\partial (L + \lambda_1 g_1 + \lambda_2 g_2)}{\partial x_1} &= 0 \\ \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{y}_1} \right) - \frac{\partial (L + \lambda_1 g_1 + \lambda_2 g_2)}{\partial y_1} &= 0 \\ \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}_2} \right) - \frac{\partial (L + \lambda_1 g_1 + \lambda_2 g_2)}{\partial x_2} &= 0 \\ \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{y}_2} \right) - \frac{\partial (L + \lambda_1 g_1 + \lambda_2 g_2)}{\partial y_2} &= 0. \end{aligned} \quad (6.6)$$

Daraus ergeben sich die Gleichungen

$$\begin{aligned}
 \ddot{x}_1 &= 2\lambda_1 x_1 + 2\lambda_2(x_1 - x_2) \\
 \ddot{y}_1 &= \mathbf{g} + 2\lambda_1 y_1 + 2\lambda_2(y_1 - y_2) \\
 \ddot{x}_2 &= 2\lambda_2(x_2 - x_1) \\
 \ddot{y}_2 &= \mathbf{g} + 2\lambda_2(y_2 - y_1) \\
 x_1^2 + y_1^2 &= 1 \\
 (x_1 - x_2)^2 + (y_1 - y_2)^2 &= 1.
 \end{aligned} \tag{6.7}$$

Diese vier Differentialgleichungen zweiter Ordnung können durch Einführung von vier Variablen auf acht Differentialgleichungen erster Ordnung transformiert werden. Durch $F_1 = -2\lambda_1$ und $F_2 = -2\lambda_2$ resultiert das am Beginn des Abschnitts genannte System.

- Lagrange-Formalismus der klassischen Mechanik:

Die Funktion $L: \mathbb{R}^{12} \rightarrow \mathbb{R}$ ist definiert als

$$\begin{aligned}
 L(x_1, y_1, \lambda_1, \dot{x}_1, \dot{y}_1, \dot{\lambda}_1, x_2, y_2, \lambda_2, \dot{x}_2, \dot{y}_2, \dot{\lambda}_2) &= E_{kin} - E_{pot} - \lambda_1 g_1 - \lambda_2 g_2 = \\
 &= \frac{\dot{x}_1^2 + \dot{y}_1^2}{2} + \frac{\dot{x}_2^2 + \dot{y}_2^2}{2} + \mathbf{g}y_1 + \mathbf{g}y_2 - \\
 &\quad - \lambda_1(x_1^2 + y_1^2 - 1) - \lambda_2((x_1 - x_2)^2 + (y_1 - y_2)^2 - 1)
 \end{aligned} \tag{6.8}$$

und daraus resultieren mit $\frac{d}{dt} \frac{\partial L}{\partial \dot{x}_1} - \frac{\partial L}{\partial x_1} = 0$, $\frac{d}{dt} \frac{\partial L}{\partial \dot{x}_2} - \frac{\partial L}{\partial x_2} = 0$, $\frac{d}{dt} \frac{\partial L}{\partial \dot{y}_1} - \frac{\partial L}{\partial y_1} = 0$, $\frac{d}{dt} \frac{\partial L}{\partial \dot{y}_2} - \frac{\partial L}{\partial y_2} = 0$, $\frac{d}{dt} \frac{\partial L}{\partial \dot{\lambda}_1} - \frac{\partial L}{\partial \lambda_1} = 0$ und $\frac{d}{dt} \frac{\partial L}{\partial \dot{\lambda}_2} - \frac{\partial L}{\partial \lambda_2} = 0$ die Gleichungen

$$\begin{aligned}
 \ddot{x}_1 &= -2\lambda_1 x_1 - 2\lambda_2(x_1 - x_2) \\
 \ddot{y}_1 &= \mathbf{g} - 2\lambda_1 y_1 - 2\lambda_2(y_1 - y_2) \\
 \ddot{x}_2 &= -2\lambda_2(x_2 - x_1) \\
 \ddot{y}_2 &= \mathbf{g} - 2\lambda_2(y_2 - y_1) \\
 x_1^2 + y_1^2 &= 1 \\
 (x_1 - x_2)^2 + (y_1 - y_2)^2 &= 1.
 \end{aligned} \tag{6.9}$$

Diese vier Differentialgleichungen zweiter Ordnung können durch Einführung von vier Variablen auf acht Differentialgleichungen erster Ordnung transformiert werden. Durch $F_1 = 2\lambda_1$ und $F_2 = 2\lambda_2$ resultiert das am Beginn des Abschnitts genannte System.

Auf diese Art und Weise können die Bewegungsgleichungen des Doppelpendels in kartesischen Koordinaten hergeleitet werden.

Die DAE, die die Bewegung des Doppelpendels beschreibt, kann mit der Funktion $f: \mathbb{R}_0^+ \times \mathbb{R}^{10} \times \mathbb{R}^{10} \rightarrow \mathbb{R}^{10}$ und $u = (x_1, y_1, x_2, y_2, v_{x_1}, v_{y_1}, v_{x_2}, v_{y_2}, F_1, F_2)$ in impliziter Form geschrieben werden als

$$0 = f(t, u, \dot{u}) = \begin{pmatrix} \dot{x}_1 - v_{x_1} \\ \dot{y}_1 - v_{y_1} \\ \dot{x}_2 - v_{x_2} \\ \dot{y}_2 - v_{y_2} \\ \dot{v}_{x_1} + F_1 x_1 + F_2(x_1 - x_2) \\ \dot{v}_{y_1} - \mathbf{g} + F_1 y_1 + F_2(y_1 - y_2) \\ \dot{v}_{x_2} + F_2(x_2 - x_1) \\ \dot{v}_{y_2} - \mathbf{g} + F_2(y_2 - y_1) \\ x_1^2 + y_1^2 - 1 \\ (x_1 - x_2)^2 + (y_1 - y_2)^2 - 1 \end{pmatrix}. \quad (6.10)$$

Die Funktion f partiell nach \dot{u} abgeleitet ergibt

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (6.11)$$

Diese Matrix ist singulär und daher kann laut dem Hauptsatz über implizite Funktionen 2.4 das differential–algebraische Gleichungssystem nicht direkt in ein gewöhnliches Differentialgleichungssystem aufgelöst werden.

Im Folgenden werden die ersten zwei Ableitungen nach t der Funktion g berechnet, dazu wird

- $X = x_1 - x_2$
- $Y = y_1 - y_2$

gesetzt. Damit wird $\dot{x}_1 - \dot{x}_2$ zu \dot{X} . Die erste Ableitung von g errechnet sich zu

$$\dot{g}(x_1, y_1, x_2, y_2) = 2 \begin{pmatrix} x_1 \dot{x}_1 + y_1 \dot{y}_1 \\ X \dot{X} + Y \dot{Y} \end{pmatrix}. \quad (6.12)$$

Die zweite Ableitung von g berechnet sich daher zu

$$\ddot{g}(x_1, y_1, x_2, y_2) = 2 \begin{pmatrix} x_1 \ddot{x}_1 + \dot{x}_1^2 + y_1 \ddot{y}_1 + \dot{y}_1^2 \\ X \ddot{X} + \dot{X}^2 + Y \ddot{Y} + \dot{Y}^2 \end{pmatrix}. \quad (6.13)$$

In diese Gleichung können für die zweiten Ableitungen die rechten Seiten der differentiellen Gleichungen der DAE eingesetzt werden. Das ergibt

$$\ddot{g}(x_1, y_1, x_2, y_2) = 2 \begin{pmatrix} \dot{x}_1^2 + \dot{y}_1^2 + \mathbf{g}y_1 - F_1(x_1^2 + y_1^2) - F_2(x_1X + y_1Y) \\ \dot{X}^2 + \dot{Y}^2 - F_1(x_1X + y_1Y) - 2F_2(X^2 + Y^2) \end{pmatrix}. \quad (6.14)$$

Aus der zweiten Ableitung der Zwangsbedingung $g(x_1, y_1, x_2, y_2) = 0$ können die linear auftretenden Kräfte F_1 und F_2 berechnet werden. Dabei kann aus der ersten Gleichung von $g(x_1, y_1, x_2, y_2) = 0$ F_1 in Abhängigkeit von F_2 ausgedrückt werden mit

$$F_1 = \frac{\dot{x}_1^2 + \dot{y}_1^2 + \mathbf{g}y_1 - (x_1X + y_1Y)F_2}{x_1^2 + y_1^2} \quad (6.15)$$

und mithilfe dieser Gleichung kann aus der zweiten Gleichung von $g(x_1, y_1, x_2, y_2) = 0$ F_2 ausgedrückt werden durch

$$F_2 = \frac{\dot{X}^2 + \dot{Y}^2 - (\dot{x}_1^2 + \dot{y}_1^2 + \mathbf{g}y_1) \left(\frac{x_1X + y_1Y}{x_1^2 + y_1^2} \right)}{2(X^2 + Y^2) - \frac{(x_1X + y_1Y)^2}{x_1^2 + y_1^2}}. \quad (6.16)$$

Mit den ausgedrückten Kräften F_1 und F_2 kann die Index-1-Formulierung aufgestellt werden

$$\begin{aligned} \dot{x}_1 &= v_{x_1} \\ \dot{y}_1 &= v_{y_1} \\ \dot{x}_2 &= v_{x_2} \\ \dot{y}_2 &= v_{y_2} \\ \dot{v}_{x_1} &= -F_1x_1 - F_2(x_1 - x_2) \\ \dot{v}_{y_1} &= \mathbf{g} - F_1y_1 - F_2(y_1 - y_2) \\ \dot{v}_{x_2} &= -F_2(x_2 - x_1) \\ \dot{v}_{y_2} &= \mathbf{g} - F_2(y_2 - y_1), \end{aligned} \quad (6.17)$$

wobei

$$\begin{aligned} F_1 &= \frac{\dot{x}_1^2 + \dot{y}_1^2 + \mathbf{g}y_1 - (x_1X + y_1Y)F_2}{x_1^2 + y_1^2} \\ F_2 &= \frac{\dot{X}^2 + \dot{Y}^2 - (\dot{x}_1^2 + \dot{y}_1^2 + \mathbf{g}y_1) \left(\frac{x_1X + y_1Y}{x_1^2 + y_1^2} \right)}{2(X^2 + Y^2) - \frac{(x_1X + y_1Y)^2}{x_1^2 + y_1^2}}. \end{aligned} \quad (6.18)$$

Die Index-1-Formulierung ist daher nicht mehr von F_1 und F_2 abhängig und der Zustandsraum besteht aus den Variablen $x_1, y_1, x_2, y_2, v_{x_1}, v_{y_1}, v_{x_2}$ und v_{y_2} . Aufgrund der Tatsache, dass F_1 und F_2 in der zweiten Ableitung von $g(x_1, y_1, x_2, y_2)$ auftreten, treten in der dritten Ableitung von $g(x_1, y_1, x_2, y_2)$ \dot{F}_1 und \dot{F}_2 auf und somit kann aus $g^{(3)}(x_1, y_1, x_2, y_2) = 0$ durch Umformungen je eine Differentialgleichung für \dot{F}_1 und \dot{F}_2 erzeugt werden. Daher besitzt die DAE Index 3. Auf die Berechnung der dritten Ableitung von $g(x_1, y_1, x_2, y_2)$ sowie von \dot{F}_1 und \dot{F}_2 wird aus Gründen der Übersichtlichkeit verzichtet.

6.2 Indexreduktionsmethoden

In diesem Abschnitt werden die verschiedenen Indexreduktionsmethoden aus Kapitel 4 für das Doppelpendel durchgerechnet.

6.2.1 Indexreduktion durch Differentiation

In diesem Abschnitt werden die drei Methoden mithilfe von Differentiation betrachtet.

6.2.1.1 Differentiation der Zwangsbedingung

Die zweimal abgeleitete Zwangsbedingung

$$2 \begin{pmatrix} \dot{x}_1^2 + \dot{y}_1^2 + \mathbf{g}y_1 - F_1(x_1^2 + y_1^2) - F_2(x_1X + y_1Y) \\ \dot{X}^2 + \dot{Y}^2 - F_1(x_1X + y_1Y) - 2F_2(X^2 + Y^2) \end{pmatrix}. \quad (6.19)$$

ersetzt die Zwangsbedingung. Somit ergibt dies ein Index-1-System. Dieses kann, wie beim einfachen Pendel in Kapitel 5, mit bestimmten ode-Solvern von MATLAB gelöst werden.

6.2.1.2 Baumgarte-Methode

Die Baumgarte-Methode angewendet auf die Zwangsbedingung liefert

$$2 \begin{pmatrix} \dot{x}_1^2 + \dot{y}_1^2 + \mathbf{g}y_1 - F_1(x_1^2 + y_1^2) - F_2(x_1X + y_1Y) \\ \dot{X}^2 + \dot{Y}^2 - F_1(x_1X + y_1Y) - 2F_2(X^2 + Y^2) \end{pmatrix} + 4\alpha \begin{pmatrix} x_1\dot{x}_1 + y_1\dot{y}_1 \\ X\dot{X} + Y\dot{Y} \end{pmatrix} + \beta^2 \begin{pmatrix} x_1^2 + y_1^2 - 1 \\ X^2 + Y^2 - 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad (6.20)$$

wobei $\alpha > 0$. Diese Gleichungen können nach F_1 bzw. F_2 aufgelöst werden, was

$$F_1 = \frac{2(\dot{x}_1^2 + \dot{y}_1^2 + \mathbf{g}y_1 - (x_1X + y_1Y)F_2) + 4\alpha(x_1\dot{x}_1 + y_1\dot{y}_1) + \beta^2(x_1^2 + y_1^2 - 1)}{2(x_1^2 + y_1^2)} \quad (6.21)$$

und

$$F_2 = \frac{2(\dot{X}^2 + \dot{Y}^2) + 4\alpha(X\dot{X} + Y\dot{Y}) + \beta^2(X^2 + Y^2 - 1)}{4(X^2 + Y^2) - \frac{(x_1X + y_1Y)^2}{x_1^2 + y_1^2}} - \frac{\left(\frac{x_1X + y_1Y}{x_1^2 + y_1^2}\right) (2(\dot{x}_1^2 + \dot{y}_1^2 + \mathbf{g}y_1) + 4\alpha(x_1\dot{x}_1 + y_1\dot{y}_1) + \beta^2(x_1^2 + y_1^2 - 1))}{4(X^2 + Y^2) - \frac{(x_1X + y_1Y)^2}{x_1^2 + y_1^2}} \quad (6.22)$$

ergibt. Wird $\alpha = \beta = 0$ gesetzt, so ergeben sich für F_1 und F_2 genau die Gleichungen, die in der Index-1-Formulierung auftreten. Bei Anwendung der Baumgarte-Methode

werden die Gleichungen

$$\begin{aligned}
 \dot{x}_1 &= v_{x_1} \\
 \dot{y}_1 &= v_{y_1} \\
 \dot{x}_2 &= v_{x_2} \\
 \dot{y}_2 &= v_{y_2} \\
 \dot{v}_{x_1} &= -F_1 x_1 - F_2 (x_1 - x_2) \\
 \dot{v}_{y_1} &= \mathbf{g} - F_1 y_1 - F_2 (y_1 - y_2) \\
 \dot{v}_{x_2} &= -F_2 (x_2 - x_1) \\
 \dot{v}_{y_2} &= \mathbf{g} - F_2 (y_2 - y_1),
 \end{aligned} \tag{6.23}$$

wobei

$$\begin{aligned}
 F_1 &= \frac{2(\dot{x}_1^2 + \dot{y}_1^2 + \mathbf{g}y_1 - (x_1 X + y_1 Y)F_2) + 4\alpha(x_1 \dot{x}_1 + y_1 \dot{y}_1) + \beta^2(x_1^2 + y_1^2 - 1)}{2(x_1^2 + y_1^2)} \\
 F_2 &= \frac{2(\dot{X}^2 + \dot{Y}^2) + 4\alpha(X\dot{X} + Y\dot{Y}) + \beta^2(X^2 + Y^2 - 1)}{4(X^2 + Y^2) - \frac{(x_1 X + y_1 Y)^2}{x_1^2 + y_1^2}} - \\
 &\quad \frac{\left(\frac{x_1 X + y_1 Y}{x_1^2 + y_1^2}\right) (2(\dot{x}_1^2 + \dot{y}_1^2 + \mathbf{g}y_1) + 4\alpha(x_1 \dot{x}_1 + y_1 \dot{y}_1) + \beta^2(x_1^2 + y_1^2 - 1))}{4(X^2 + Y^2) - \frac{(x_1 X + y_1 Y)^2}{x_1^2 + y_1^2}},
 \end{aligned} \tag{6.24}$$

erhalten.

6.2.1.3 Pantelides–Algorithmus

Die Vorgehensweise beim Pantelides–Algorithmus für das Doppelpendel verläuft analog zum einfachen Pendel in Abschnitt 5.2.1.3. Der Ausgangspunkt für den Pantelides–Algorithmus ist das DAE–System (6.1), welches aus zehn Gleichungen und zehn Unbekannten $\dot{x}_1, \dot{y}_1, \dot{x}_2, \dot{y}_2, \dot{v}_{x_1}, \dot{v}_{y_1}, \dot{v}_{x_2}, \dot{v}_{y_2}, F_1$ und F_2 besteht. Anwenden des Pantelides–Algorithmus führt bei Einführen der Variablen dx_1 und dx_2 auf 18 Gleichungen und 18 Unbekannte ($x_1, x_2, dx_1, dx_2, d2x_1, d2x_2, d2y_1, d2y_2, \dot{y}_1, \dot{y}_2, v_{x_1}, v_{x_2}, dv_{x_1}, dv_{x_2}, \dot{v}_{y_1}, \dot{v}_{y_2}, F_1$ und F_2). Diese Gleichungen lauten

$$dx_1 = v_{x_1} \tag{6.25a}$$

$$d2x_1 = dv_{x_1} \tag{6.25b}$$

$$\dot{y}_1 = v_{y_1} \tag{6.25c}$$

$$d2y_1 = \dot{v}_{y_1} \tag{6.25d}$$

$$dv_{x_1} = -F_1 x_1 - F_2 (x_1 - x_2) \tag{6.25e}$$

$$\dot{v}_{y_1} = \mathbf{g} - F_1 y_1 - F_2 (y_1 - y_2) \tag{6.25f}$$

$$x_1^2 + y_1^2 = 1 \tag{6.25g}$$

$$x_1 dx_1 + y_1 \dot{y}_1 = 0 \quad (6.25h)$$

$$(dx_1)^2 + x_1 d^2x_1 + (\dot{y}_1)^2 + y_1 d^2y_1 = 0 \quad (6.25i)$$

$$dx_2 = v_{x_2} \quad (6.25j)$$

$$d^2x_2 = dv_{x_2} \quad (6.25k)$$

$$\dot{y}_2 = v_{y_2} \quad (6.25l)$$

$$d^2y_2 = \dot{v}_{y_2} \quad (6.25m)$$

$$dv_{x_2} = -F_2(x_2 - x_1) \quad (6.25n)$$

$$\dot{v}_{y_2} = \mathbf{g} - F_2(y_2 - y_1) \quad (6.25o)$$

$$(x_1 - x_2)^2 + (y_1 - y_2)^2 = 1 \quad (6.25p)$$

$$(x_1 - x_2)(dx_1 - dx_2) + (y_1 - y_2)(\dot{y}_1 - \dot{y}_2) = 0 \quad (6.25q)$$

$$(dx_1 - dx_2)^2 + (x_1 - x_2)(d^2x_1 - d^2x_2) + (\dot{y}_1 - \dot{y}_2)^2 + (y_1 - y_2)(d^2y_1 - d^2y_2) = 0. \quad (6.25r)$$

Diese Gleichungen können umgeformt werden zu

$$\dot{y}_1 = v_{y_1} \quad (6.26a)$$

$$x_1 = \pm \sqrt{1 - y_1^2} \quad (6.26b)$$

$$dx_1 = -\frac{y_1}{x_1} \dot{y}_1 \quad (6.26c)$$

$$dv_{x_1} = d^2x_1 \quad (6.26d)$$

$$F_1 = -\frac{dv_{x_1} + F_2(x_1 - x_2)}{x_1} \quad (6.26e)$$

$$\dot{v}_{y_1} = \mathbf{g} - F_1(y_1) - F_2(y_1 - y_2) \quad (6.26f)$$

$$d^2y_1 = \dot{v}_{y_1} \quad (6.26g)$$

$$d^2x_1 = -\frac{(dx_1)^2 + (\dot{y}_1)^2 + y_1 d^2y_1}{x_1} \quad (6.26h)$$

$$v_{x_1} = dx_1 \quad (6.26i)$$

$$\dot{y}_2 = v_{y_2} \quad (6.26j)$$

$$x_2 = x_1 \mp \sqrt{1 - (y_1 - y_2)^2} \quad (6.26k)$$

$$dx_2 = dx_1 + \frac{(y_1 - y_2)}{x_1 - x_2} (\dot{y}_1 - \dot{y}_2) \quad (6.26l)$$

$$dv_{x_2} = d^2x_2 \quad (6.26m)$$

$$F_2 = \frac{dv_{x_2}}{x_1 - x_2} \quad (6.26n)$$

$$\dot{v}_{y_2} = \mathbf{g} - F_2(y_2 - y_1) \quad (6.26o)$$

$$d2y_2 = \dot{v}_{y_2} \quad (6.26p)$$

$$d2x_2 = \frac{(dx_1 - dx_2)^2 + (\dot{y}_1 - \dot{y}_2)^2 + (y_1 - y_2)(d2y_1 - d2y_2) + (x_1 - x_2)d2x_1}{x_1 - x_2} \quad (6.26q)$$

$$v_{x_2} = dx_2. \quad (6.26r)$$

Analog zum einfachen Pendel besitzt jede der zwei Massen vier verschiedene Gleichungssysteme, die entstehen je nachdem welche Variable aufgelöst wird und welches Vorzeichen verwendet wird. Daraus ergeben sich für das Gesamtsystem 16 verschiedene Gleichungssysteme, die zur Beschreibung notwendig sind. Mittels Fallunterscheidungen muss zwischen den verschiedenen Gleichungssystemen umgeschaltet werden.

6.2.2 Stabilisierung durch Projektion

Bei diesen Methoden wird die Index-1-Formulierung, siehe Abschnitt 6.1, mithilfe von Einschrittverfahren gelöst. Durch die Verwendung einer Projektion wird sichergestellt, dass die numerische Lösung auf der Mannigfaltigkeit liegt. Diese Mannigfaltigkeit ist gegeben durch

$$M = \{z \in \mathbb{R}^8 : h(z) = 0\}, \quad (6.27)$$

wobei $z = (x_1, y_1, v_{x_1}, v_{y_1}, x_2, y_2, v_{x_2}, v_{y_2})$ und $h: \mathbb{R}^8 \rightarrow \mathbb{R}^4$ definiert ist als

$$h(z) = \begin{pmatrix} x_1^2 + y_1^2 - 1 \\ (x_1 - x_2)^2 + (y_1 - y_2)^2 - 1 \\ x_1 v_{x_1} + y_1 v_{y_1} \\ (x_1 - x_2)(v_{x_1} - v_{x_2}) + (y_1 - y_2)(v_{y_1} - v_{y_2}) \end{pmatrix}. \quad (6.28)$$

Die Jacobimatrix $\frac{\partial h}{\partial z}$ lautet

$$\begin{pmatrix} 2x_1 & 2y_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2X & 2Y & 0 & 0 & -2X & -2Y & 0 & 0 \\ v_{x_1} & v_{y_1} & x_1 & y_1 & 0 & 0 & 0 & 0 \\ \dot{X} & \dot{Y} & X & Y & -\dot{X} & -\dot{Y} & -X & -Y \end{pmatrix}. \quad (6.29)$$

Diese hat für alle $z \in \mathbb{R}^8 \setminus \{(0, 0, a, b, r, t, c, d), (r, r, a, b, t, t, c, d)\}$ mit $r, t, a, b, c, d \in \mathbb{R}$ ($(0, 0, a, b, r, t, c, d)$ und (r, r, a, b, t, t, c, d) kann nicht auftreten, da immer $x_1^2 + y_1^2 = 1$ und $(x_1 - x_2)^2 + (y_1 - y_2)^2 = 1$ gelten muss) Rang 4 und ist daher nach Lemma 2.6 eine implizit definierte Mannigfaltigkeit. M ist eine 4-dimensionale Mannigfaltigkeit im \mathbb{R}^8 . Der Tangentialraum von M errechnet sich mit $w = (x_1, y_1, x_2, y_2)$ in diesem Fall als

$$T_z M = \left\{ v \in \mathbb{R}^8 : \frac{\partial g}{\partial w}(z) \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \dot{w} \otimes \frac{\partial^2 g}{\partial w^2}(z) \otimes \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} + \frac{\partial g}{\partial w}(z) \frac{\partial \dot{w}}{\partial \dot{w}} \begin{pmatrix} v_5 \\ v_6 \\ v_7 \\ v_8 \end{pmatrix} = 0 \right\}. \quad (6.30)$$

Umformungen ergeben

$$T_z M = \left\{ v \in \mathbb{R}^8 : \begin{pmatrix} x_1 v_1 + y_1 v_2 \\ X(v_1 - v_3) + Y(v_2 - v_4) \\ v_{x_1} v_1 + v_{y_1} v_2 + x_1 v_5 + y_1 v_6 \\ \dot{X}(v_1 - v_3) + \dot{Y}(v_2 - v_4) + X(v_5 - v_7) + Y(v_6 - v_8) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right\}. \quad (6.31)$$

Eine Basis des 4-dimensionalen Tangentialraumes $T_z M$ ist daher gegeben durch

$$\begin{pmatrix} -y_1 & 0 & 0 & 0 \\ x_1 & 0 & 0 & 0 \\ -y_2 & 0 & -Y & 0 \\ x_2 & 0 & X & 0 \\ -v_{y_1} & -y_1 & 0 & 0 \\ v_{x_1} & x_1 & 0 & 0 \\ -v_{y_2} & -y_2 & -\dot{Y} & -Y \\ v_{x_2} & x_2 & \dot{X} & X \end{pmatrix}. \quad (6.32)$$

6.2.2.1 Standardprojektionsmethode

Wie in Abschnitt 5.2.2.1 werden auch bei diesem Beispiel zuerst die Positionskomponenten und anschließend die Geschwindigkeitskomponenten projiziert. Die orthogonale Projektion wird mithilfe der Abbildungen $p_1: \mathbb{R}^4 \rightarrow M_1$ (Projektion der Positionskomponenten) und $p_2: \mathbb{R}^4 \rightarrow M_2$ (Projektion der Geschwindigkeitskomponenten) durchgeführt. Dabei sind die Mengen M_1 und M_2 folgendermaßen definiert

$$M_1 = \left\{ (x_1, y_1, x_2, y_2) \in \mathbb{R}^4 : h_1(x_1, y_1, x_2, y_2) = \begin{pmatrix} x_1^2 + y_1^2 - 1 \\ (x_1 - x_2)^2 + (y_1 - y_2)^2 - 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\} \quad (6.33)$$

und

$$M_2 = \left\{ (v_{x_1}, v_{y_1}, v_{x_2}, v_{y_2}) \in \mathbb{R}^4 : h_2(v_{x_1}, v_{y_1}, v_{x_2}, v_{y_2}) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\}, \quad (6.34)$$

wobei

$$h_2 \begin{pmatrix} v_{x_1} \\ v_{y_1} \\ v_{x_2} \\ v_{y_2} \end{pmatrix} = \begin{pmatrix} \pi_1 \circ p_1(w)v_{x_1} + \pi_2 \circ p_1(w)v_{y_1} \\ (\pi_1 \circ p_1(w) - \pi_3 \circ p_1(w))\dot{X} + (\pi_2 \circ p_1(w) - \pi_4 \circ p_1(w))\dot{Y} \end{pmatrix}. \quad (6.35)$$

Die Mengen M_1 und M_2 sind laut Lemma 2.6 implizit definierte Mannigfaltigkeiten, da

$$\frac{\partial h_1}{\partial(x_1, y_1, x_2, y_2)} = \begin{pmatrix} 2x_1 & 2y_1 & 0 & 0 \\ 2(x_1 - x_2) & 2(y_1 - y_2) & -2(x_1 - x_2) & -2(y_1 - y_2) \end{pmatrix} \quad (6.36)$$

Rang 2 besitzt für alle $(x_1, y_1, x_2, y_2) \in \mathbb{R}^4 \setminus \{(0, 0, a, b), (a, b, a, b)\}$ mit $a, b \in \mathbb{R}$ und

$$\frac{\partial h_2}{\partial V} = \begin{pmatrix} \pi_1 \circ p_1(w) & \pi_2 \circ p_1(w) & 0 & 0 \\ p_{1,X} & p_{1,Y} & -p_{1,X} & -p_{1,Y} \end{pmatrix} \quad (6.37)$$

mit $V = (v_{x_1}, v_{y_1}, v_{x_2}, v_{y_2})$, $p_{1,X} = \pi_1 \circ p_1(w) - \pi_3 \circ p_1(w)$ und $p_{1,Y} = \pi_2 \circ p_1(w) - \pi_4 \circ p_1(w)$ ebenfalls Rang 2 besitzt für alle $(v_{x_1}, v_{y_1}, v_{x_2}, v_{y_2}) \in \mathbb{R}^4$.

Die Projektionen p_1 und p_2 sind gegeben durch

$$p_1(x_1, y_1, x_2, y_2) = \begin{pmatrix} \frac{x_1}{\sqrt{x_1^2 + y_1^2}} \\ \frac{y_1}{\sqrt{x_1^2 + y_1^2}} \\ \frac{x_1}{\sqrt{x_1^2 + y_1^2}} + \frac{1}{d} \left(x_2 - \frac{x_1}{\sqrt{x_1^2 + y_1^2}} \right) \\ \frac{y_1}{\sqrt{x_1^2 + y_1^2}} + \frac{1}{d} \left(y_2 - \frac{y_1}{\sqrt{x_1^2 + y_1^2}} \right) \end{pmatrix}, \quad (6.38)$$

wobei $d = \sqrt{\left(x_2 - \frac{x_1}{\sqrt{x_1^2 + y_1^2}} \right)^2 + \left(y_2 - \frac{y_1}{\sqrt{x_1^2 + y_1^2}} \right)^2}$, und

$$p_2(V) = \begin{pmatrix} (-\pi_2 \circ p_1(w)v_{x_1} + \pi_1 \circ p_1(w)v_y)(-\pi_2 \circ p_1(w)) \\ (-\pi_2 \circ p_1(w)v_{x_1} + \pi_1 \circ p_1(w)v_y)\pi_1 \circ p_1(w) \\ (v_{x_2}p_{1,Y} - v_{y_2}p_{1,X})(p_{1,Y}) \\ (v_{x_2}p_{1,Y} - v_{y_2}p_{1,X})(-p_{1,X}) \end{pmatrix}. \quad (6.39)$$

Damit ergibt sich die orthogonale Projektion $p: \mathbb{R}^8 \rightarrow M \subset \mathbb{R}^8$ als

$$p \begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ v_{x_1} \\ v_{y_1} \\ v_{x_2} \\ v_{y_2} \end{pmatrix} = \begin{pmatrix} p_1 \begin{pmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \end{pmatrix} \\ p_2 \begin{pmatrix} v_{x_1} \\ v_{y_1} \\ v_{x_2} \\ v_{y_2} \end{pmatrix} \end{pmatrix}. \quad (6.40)$$

6.2.2.2 Symmetrische Projektionsmethode

Für das zu lösende Gleichungssystem (4.9) ergibt sich

$$\begin{pmatrix} I_{8 \times 8} & -2G^T \\ G & 0_{4 \times 4} \end{pmatrix} \begin{pmatrix} \Delta x_{1,n+1} \\ \Delta y_{1,n+1} \\ \Delta x_{2,n+1} \\ \Delta y_{2,n+1} \\ \Delta v_{x_{1,n+1}} \\ \Delta v_{y_{1,n+1}} \\ \Delta v_{x_{2,n+1}} \\ \Delta v_{y_{2,n+1}} \\ \Delta \mu_1 \\ \Delta \mu_2 \\ \Delta \mu_3 \\ \Delta \mu_4 \end{pmatrix} = - \begin{pmatrix} \hat{x}_{1,n} + h\Psi(h, \hat{x}_{1,n}, \hat{x}_{1,n+1}) - \hat{x}_{1,n+1} \\ \hat{y}_{1,n} + h\Psi(h, \hat{y}_{1,n}, \hat{y}_{1,n+1}) - \hat{y}_{1,n+1} \\ \hat{x}_{2,n} + h\Psi(h, \hat{x}_{2,n}, \hat{x}_{2,n+1}) - \hat{x}_{2,n+1} \\ \hat{y}_{2,n} + h\Psi(h, \hat{y}_{2,n}, \hat{y}_{2,n+1}) - \hat{y}_{2,n+1} \\ \hat{v}_{x_{1,n}} + h\Psi(h, \hat{v}_{x_{1,n}}, \hat{v}_{x_{1,n+1}}) - \hat{v}_{x_{1,n+1}} \\ \hat{v}_{y_{1,n}} + h\Psi(h, \hat{v}_{y_{1,n}}, \hat{v}_{y_{1,n+1}}) - \hat{v}_{y_{1,n+1}} \\ \hat{v}_{x_{2,n}} + h\Psi(h, \hat{v}_{x_{2,n}}, \hat{v}_{x_{2,n+1}}) - \hat{v}_{x_{2,n+1}} \\ \hat{v}_{y_{2,n}} + h\Psi(h, \hat{v}_{y_{2,n}}, \hat{v}_{y_{2,n+1}}) - \hat{v}_{y_{2,n+1}} \\ x_{1,n+1}^2 + y_{1,n+1}^2 - 1 \\ X_{n+1}^2 + Y_{n+1}^2 - 1 \\ x_{1,n+1}v_{x_{1,n+1}} + y_{1,n+1}v_{y_{1,n+1}} \\ X_{n+1}\dot{X}_{n+1} + Y_{n+1}\dot{Y}_{n+1} \end{pmatrix}, \quad (6.41)$$

wobei $I_{8 \times 8}$ die 8×8 -Einheitsmatrix, $0_{4 \times 4}$ die 4×4 -Nullmatrix, Ψ die Inkrementfunktion der Trapezregel und

$$G = \begin{pmatrix} 2x_1 & 2y_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2X & 2Y & -2X & -2Y & 0 & 0 & 0 & 0 \\ v_{x_1} & v_{y_1} & 0 & 0 & x_1 & y_1 & 0 & 0 \\ \dot{X} & \dot{Y} & -\dot{X} & -\dot{Y} & X & Y & -X & -Y \end{pmatrix} \quad (6.42)$$

ist.

6.2.3 Methoden basierend auf lokaler Zustandsraumtransformation

Ausgangspunkt ist die Index-1-Formulierung aus der F_1 und F_2 berechnet werden können. Das Lösen der DAE mithilfe einer Lösungsmannigfaltigkeit und lokaler Zustandsraumtransformation wird durch eine globale Transformation umgesetzt. Die Mannigfaltigkeit ist gegeben durch

$$M = \{z \in \mathbb{R}^8 : h(z) = 0\} \quad (6.43)$$

mit $z = (x_1, y_1, v_{x_1}, v_{y_1}, x_2, y_2, v_{x_2}, v_{y_2})$ und $h: \mathbb{R}^8 \rightarrow \mathbb{R}^4$ ist definiert als

$$h(z) = \begin{pmatrix} x_1^2 + y_1^2 - 1 \\ (x_1 - x_2)^2 + (y_1 - y_2)^2 - 1 \\ x_1 v_{x_1} + y_1 v_{y_1} \\ (x_1 - x_2)(v_{x_1} - v_{x_2}) + (y_1 - y_2)(v_{y_1} - v_{y_2}) \end{pmatrix}. \quad (6.44)$$

Die Transformation $(x_1, y_1, v_{x_1}, v_{y_1}, x_2, y_2, v_{x_2}, v_{y_2}) = \psi(\varphi_1, \eta_1, \varphi_2, \eta_2)$ ist gegeben durch

$$\begin{aligned}
 x_1 &= \cos \varphi_1 \\
 y_1 &= \sin \varphi_1 \\
 v_{x_1} &= -\eta_1 \sin \varphi_1 \\
 v_{y_1} &= \eta_1 \cos \varphi_1 \\
 x_2 &= \cos \varphi_1 + \cos \varphi_2 \\
 y_2 &= \sin \varphi_1 + \sin \varphi_2 \\
 v_{x_2} &= -\eta_1 \sin \varphi_1 - \eta_2 \sin \varphi_2 \\
 v_{y_2} &= \eta_1 \cos \varphi_1 + \eta_2 \cos \varphi_2.
 \end{aligned} \tag{6.45}$$

Mithilfe der Pseudoinversen A^+ von $A := \frac{\partial \psi(\varphi_1, \eta_1, \varphi_2, \eta_2)}{\partial (\varphi_1, \eta_1, \varphi_2, \eta_2)}$ wird das Differentialgleichungssystem

$$\begin{pmatrix} \dot{\varphi}_1 \\ \dot{\eta}_1 \\ \dot{\varphi}_2 \\ \dot{\eta}_2 \end{pmatrix} = A^+ f(\psi(\varphi, \eta)) \tag{6.46}$$

erhalten. Da die Berechnung der Pseudoinversen von A und des entstehenden gewöhnlichen Differentialgleichungssystems sehr komplex ist, wurden die Berechnung mithilfe der Symbolic Math Toolbox von MATLAB durchgeführt. Das Differentialgleichungssystem errechnet sich somit zu

$$\begin{aligned}
 \dot{\varphi}_1 &= \eta_1 \\
 \dot{\eta}_1 &= \mathbf{g} \cos(\varphi_1) + \left(\frac{\dot{\varphi}_2^2 + (\sin(\varphi_1) \sin(\varphi_2) + \cos(\varphi_1) \cos(\varphi_2))(\dot{\varphi}_1^2 + \mathbf{g} \sin(\varphi_1))}{2 - (\sin(\varphi_1) \sin(\varphi_2) + \cos(\varphi_1) \cos(\varphi_2))^2} \right) \\
 &\quad \cdot (-\sin(\varphi_1) \cos(\varphi_2) + \cos(\varphi_1) \sin(\varphi_2)) \\
 \dot{\varphi}_2 &= \eta_2 \\
 \dot{\eta}_2 &= \mathbf{g} \cos(\varphi_2) - \dot{\varphi}_1^2 (\cos(\varphi_1) \sin(\varphi_2) - \sin(\varphi_1) \cos(\varphi_2)) - \\
 &\quad - \ddot{\varphi}_1 (\sin(\varphi_1) \sin(\varphi_2) + \cos(\varphi_1) \cos(\varphi_2)).
 \end{aligned} \tag{6.47}$$

Die Zwangsbedingung und deren Ableitung sind durch die gewählte Koordinatentransformation erfüllt.

Als alternative Möglichkeit kann, wie beim einfachen Pendel, eine Transformation in die gegebenen Differentialgleichungen zweiter Ordnung eingesetzt werden. Diese Transformation lautet

$$\begin{aligned}
 x_1 &= \cos(\varphi_1) \\
 y_1 &= \sin(\varphi_1) \\
 x_2 &= \cos(\varphi_1) + \cos(\varphi_2) \\
 y_2 &= \sin(\varphi_1) + \sin(\varphi_2).
 \end{aligned} \tag{6.48}$$

Für die Kraft F_2 ergibt sich die Gleichung

$$F_2 = \frac{\dot{\varphi}_2^2 + (\sin(\varphi_1) \sin(\varphi_2) + \cos(\varphi_1) \cos(\varphi_2))(\dot{\varphi}_1^2 + \mathbf{g} \sin(\varphi_1))}{2 - (\sin(\varphi_1) \sin(\varphi_2) + \cos(\varphi_1) \cos(\varphi_2))^2}. \tag{6.49}$$

Die Kraft F_1 kommt in den Differentialgleichungen nicht mehr vor. Die vier gegebenen Differentialgleichungen zweiter Ordnung lauten nach der Transformation

$$\begin{aligned}\ddot{\varphi}_1 &= \mathbf{g} \cos(\varphi_1) + F_2(-\sin(\varphi_1) \cos(\varphi_2) + \cos(\varphi_1) \sin(\varphi_2)) \\ \ddot{\varphi}_2 &= \mathbf{g} \cos(\varphi_2) - \dot{\varphi}_1^2(\cos(\varphi_1) \sin(\varphi_2) - \sin(\varphi_1) \cos(\varphi_2)) - \\ &\quad - \ddot{\varphi}_1(\sin(\varphi_1) \sin(\varphi_2) + \cos(\varphi_1) \cos(\varphi_2)).\end{aligned}\tag{6.50}$$

Diese beiden Differentialgleichungen erster Ordnung können so umgeschrieben werden, dass ein Differentialgleichungssystem erster Ordnung entsteht.

Unabhängig von der gewählten Vorgehensweise werden dieselben zwei Differentialgleichungen zweiter Ordnung bzw. vier Differentialgleichungen erster Ordnung erhalten.

6.3 Simulationen

Alle folgenden Simulationen wurden in MATLAB durchgeführt. Es wurden verschiedene numerische Verfahren bzw. ode-Solver zum Lösen der DAE verwendet. Alle Laufzeiten wurden mithilfe der MATLAB-Befehle `tic` und `toc`, über 10 Durchläufe gemittelt, berechnet und in Sekunden (auf drei Nachkommastellen gerundet) angegeben. Bei diesem Beispiel wurde der Abstand von (x_1, y_1) zur Einheitskreislinie gemessen, das heißt

$$e_{1_i} = x_{1_i}^2 + y_{1_i}^2 - 1,\tag{6.51}$$

wobei x_{1_i} und y_{1_i} die numerisch berechneten Werte für x_1 und y_1 sind. Außerdem wurde der Abstand von $(x_1 - x_2, y_1 - y_2) = (X, Y)$ zur Einheitskreislinie gemessen, das heißt

$$e_{2_i} = (x_{1_i} - x_{2_i})^2 + (y_{1_i} - y_{2_i})^2 - 1 = X_i^2 + Y_i^2 - 1,\tag{6.52}$$

wobei x_{1_i} , y_{1_i} , x_{2_i} und y_{2_i} die numerisch berechneten Werte für x_1 , y_1 , x_2 und y_2 sind und es gilt $X_i := x_{1_i} - x_{2_i}$, $Y_i := y_{1_i} - y_{2_i}$. Es wurden zu jeder Simulation die maximalen Fehler e_1 und e_2

$$e_1 = \max_i(x_{1_i}^2 + y_{1_i}^2 - 1) \quad \text{und} \quad e_2 = \max_i(X_i^2 + Y_i^2 - 1)\tag{6.53}$$

berechnet.

Die verwendeten Werte für die Parameter und die Simulationszeit wurden für die folgenden Simulationen folgendermaßen gewählt:

- Erdbeschleunigung: $\mathbf{g} = 9.81$
- Endzeit der Simulation: 10 bzw. 100.

Die Berechnungen wurden auf einem Laptop mit Intel Core i5 1.6GHz Prozessor und 4GB RAM durchgeführt. Es wurde die MATLAB-Version R2012b verwendet.

Bei den Simulationen der Index-0-Formulierung und Index-1-Formulierung wurden ausschließlich die ode-Solver `ode45`, `ode23t` und `ode15s` verwendet.

6.3.1 Simulationen der Index–0–Formulierung

Für die Simulationen des Gleichungssystems der Index–0–Formulierung wurden die in Tabelle 6.1 aufgelisteten Werte verwendet.

Anfangswerte									
x_1	y_1	v_{x_1}	v_{y_1}	x_2	y_2	v_{x_2}	v_{y_2}	F_1	F_2
1	0	0	0	2	0	0	0	0	0

Tabelle 6.1: Anfangswerte für die Simulation der Bewegungsgleichungen des Doppelpendels mit Index 0

Die Anfangswerte für F_1 und F_2 wurden so gewählt, dass sie mit den Werten von F_1 und F_2 in den Simulationen der Index–1–Formulierung zusammenpassen. Ansonsten haben die gewählten Anfangswerte keine spezielle Bedeutung.

Die Simulationen der Index–0–Formulierung brechen bei einem Wert von etwa 5 Sekunden ab. In Abbildung 6.2 sind die Position des Pendels mit den Koordinaten (x_1, y_1) bei Verwendung des ode–Solvers `ode45` und die Positionen (X, Y) bei Verwendung des ode–Solvers `ode15s` dargestellt.

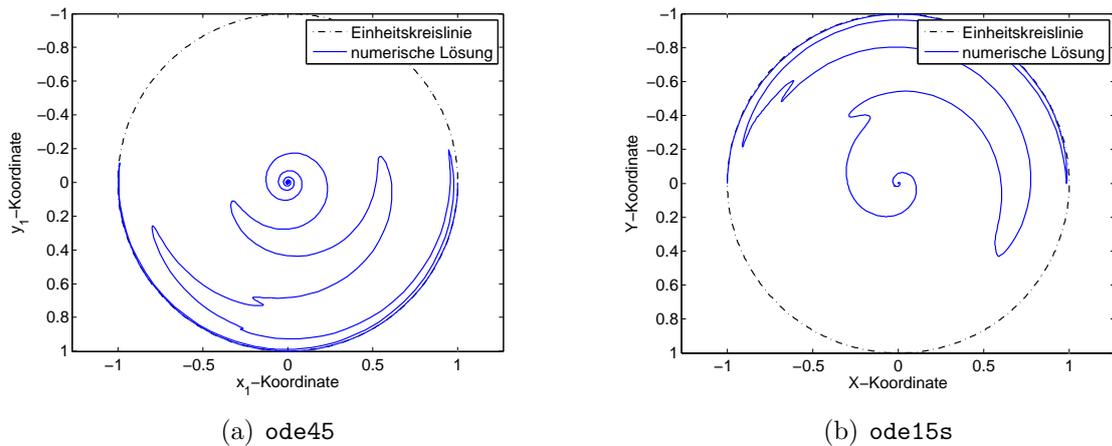


Abbildung 6.2: Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) der Index–0–Formulierung der Bewegungsgleichungen des Doppelpendels bis zum Abbruch des Programmes

Wie in Abbildung 6.2 zu erkennen ist, driftet die numerische Lösung von der Lösungsmannigfaltigkeit weg. Bei Verwendung des ode–Solver `ode45` passiert der Abbruch in dem Moment, wo die Position des Pendels mit den Koordinaten (x_1, y_1) im Punkt $(0, 0)$ ist. Bei Verwendung der ode–Solver `ode23t` und `ode15s` passiert dasselbe mit den Positionen (X, Y) .

Da die Simulationen nur bis etwa 5 Sekunden lauffähig sind, wird auf die Darstellung der anderen Plots verzichtet.

6.3.2 Simulationen der Index–1–Formulierung

Für die Simulationen des Gleichungssystems der Index–1–Formulierung, die in Abschnitt 6.1 aufgestellt wurde, wurden die in Tabelle 6.2 aufgelisteten Werte verwendet.

Anfangswerte							
x_1	y_1	v_{x_1}	v_{y_1}	x_2	y_2	v_{x_2}	v_{y_2}
1	0	0	0	2	0	0	0

Tabelle 6.2: Anfangswerte für die Simulation der Bewegungsgleichungen des Doppelpendels mit Index 1

Hier ist die Simulation bei Verwendung der ode–Solver `ode45` und `ode15s` bis 100 Sekunden möglich. Der ode–Solver `ode23t` bricht nach ungefähr 9 Sekunden ab.

In den Abbildung 6.3, 6.4 und 6.5 ist das Ergebnis der Simulation mit Endzeit 10 und dem ode–Solver `ode45` dargestellt. In Abbildung 6.3 sind die numerischen Lösungen im Vergleich zur Einheitskreislinie dargestellt.

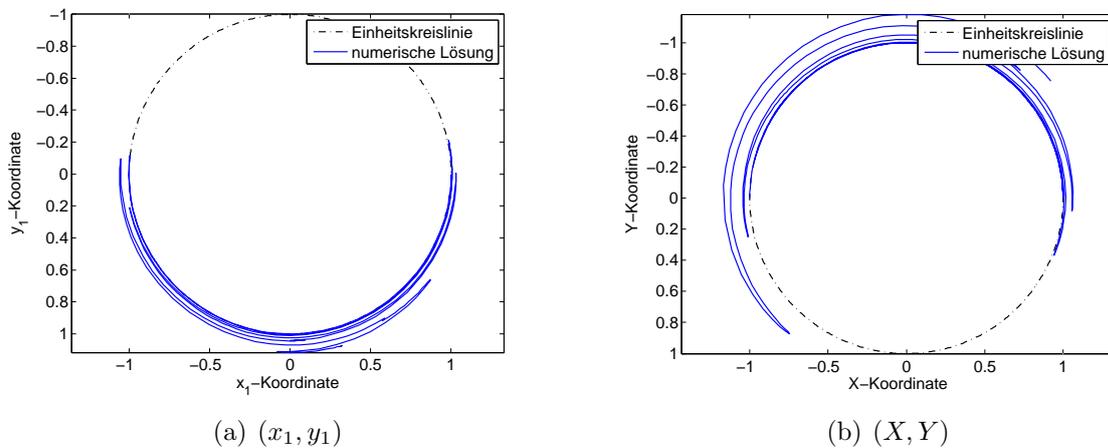


Abbildung 6.3: Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) der Index–1–Formulierung der Bewegungsgleichungen des Doppelpendels mit `ode45` und Endzeit 10

Wie in der obigen Abbildung zu erkennen ist, „driftet“ die numerische Lösung von der Lösungsmannigfaltigkeit weg. Bei Simulation bis 100 Sekunden entfernt sie sich immer weiter.

In Abbildung 6.4 sind die Fehler zur Einheitskreislinie dargestellt.

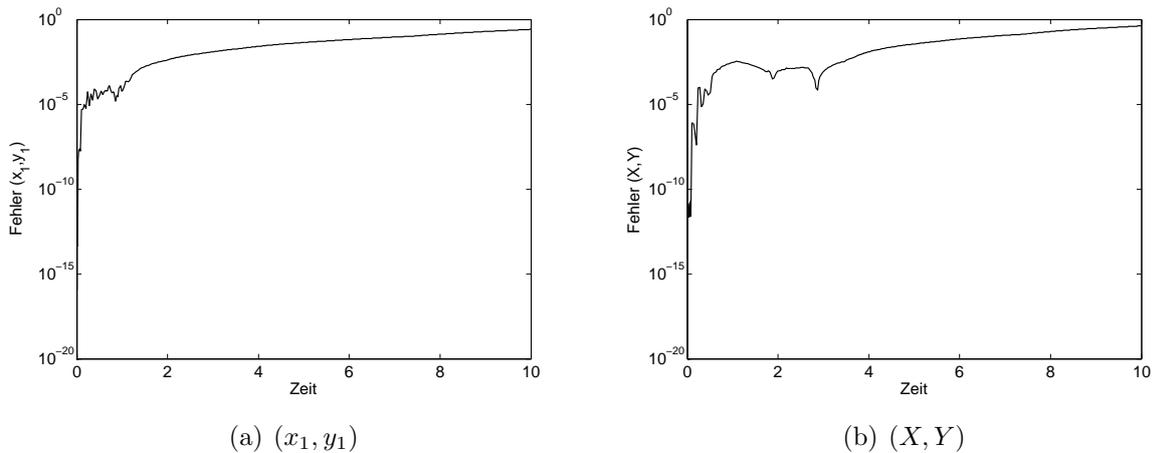


Abbildung 6.4: Fehler der numerischen Lösung zur Einheitskreislinie der Index-1-Formulierung der Bewegungsgleichungen des Doppelpendels mit `ode45` und Endzeit 10

In Abbildung 6.5 wird die numerische Lösung der Position des Pendels (x_2, y_2) dargestellt.

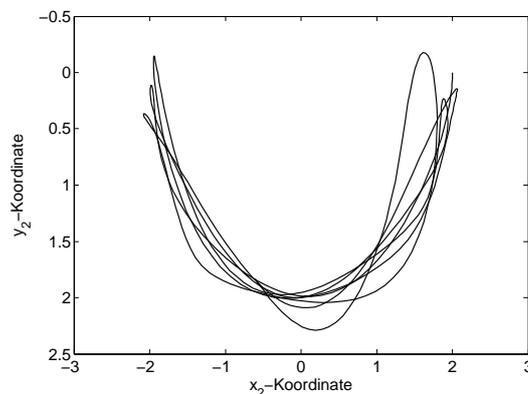


Abbildung 6.5: Numerische Lösung für (x_2, y_2) der Index-1-Formulierung der Bewegungsgleichungen des Doppelpendels mit `ode45` und Endzeit 10

Bei Verwendung des ode-Solvers `ode15s` entfernt sich die numerische Lösung ebenfalls, wobei die numerische Lösung für die Koordinaten (x_1, y_1) zuerst Richtung $(0, 0)$ läuft und dann nach außen. Aufgrund der starken Abweichung der Ergebnisse zur Lösungsmannigfaltigkeit unter Verwendung der beiden ode-Solver ist diese Vorgehensweise also nicht geeignet, um die gegebene DAE zu lösen.

6.3.3 Simulationen der Indexreduktionsmethoden

Im Folgenden werden die Lösungen der Simulationen mit den verschiedenen Indexreduktionsmethoden präsentiert.

6.3.3.1 Differentiation der Zwangsbedingung

Es wurden die in Tabelle 6.3 aufgelisteten Werte für die Simulation des Systems aus Abschnitt 6.2.1.1, das durch Differentiation der Zwangsbedingung erhalten wurde, verwendet.

Anfangswerte									
x_1	y_1	v_{x_1}	v_{y_1}	x_2	y_2	v_{x_2}	v_{y_2}	F_1	F_2
1	0	0	0	2	0	0	0	0	0

Tabelle 6.3: Anfangswerte für die Simulation der Bewegungsgleichungen des Doppelpendels mit der Indexreduktionsmethode Differentiation der Zwangsbedingung

Die Simulation wurde mit dem ode-Solver `ode15s` durchgeführt, da dieser für das Lösen von DAEs mit Index 1 geeignet ist.

In Abbildung 6.6 sind die numerischen Lösungen im Vergleich zur Einheitskreislinie dargestellt.

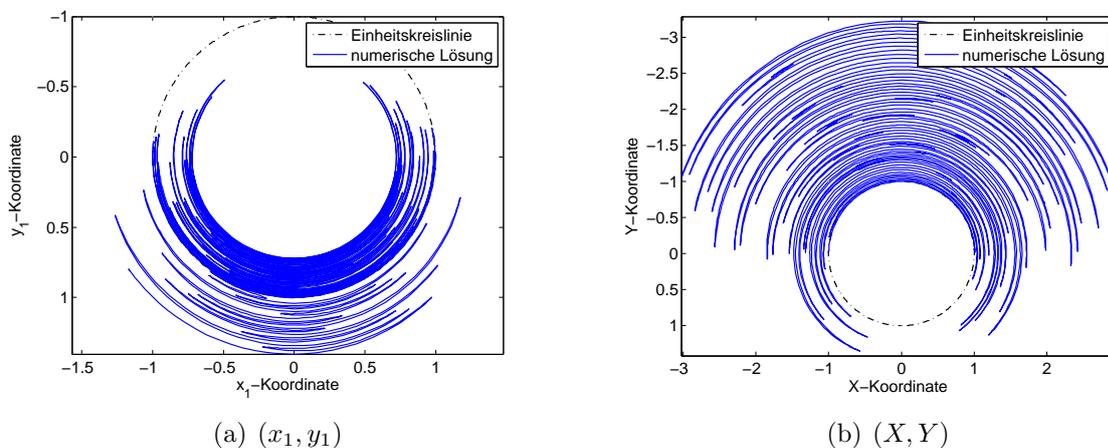


Abbildung 6.6: Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) mit Differentiation der Zwangsbedingung und Endzeit 100

Wie in Abbildung 6.6 zu erkennen ist, weicht die numerische Lösung von der Lösungsmannigfaltigkeit ab, wobei sie für die Koordinaten (x_1, y_1) zuerst Richtung $(0, 0)$ läuft und dann nach außen.

In Abbildung 6.7 werden die erhaltenen Fehler dargestellt.

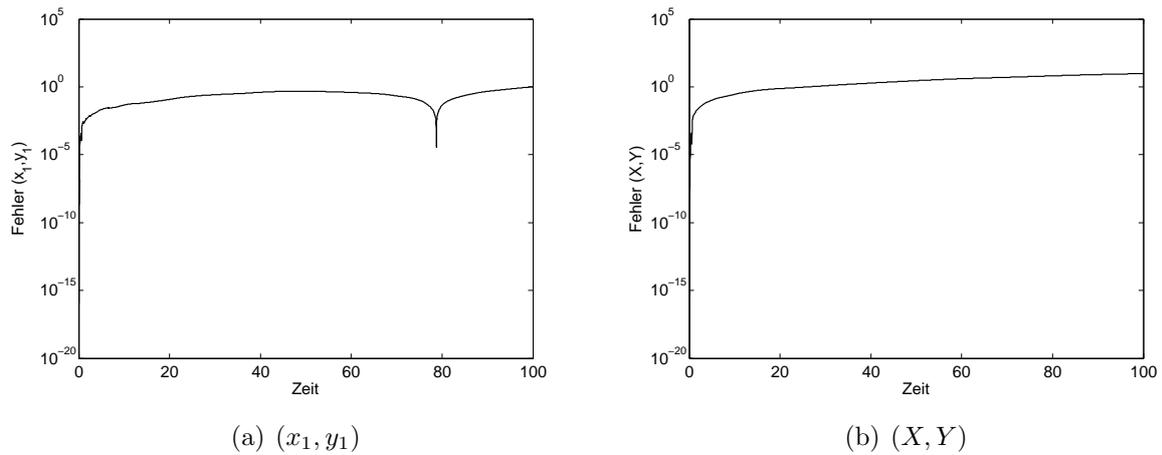


Abbildung 6.7: Fehler der numerischen Lösung zur Einheitskreislinie mit Differentiation der Zwangsbedingung und Endzeit 100

In Abbildung 6.8 wird die numerische Lösung der Position des Pendels (x_2, y_2) dargestellt.

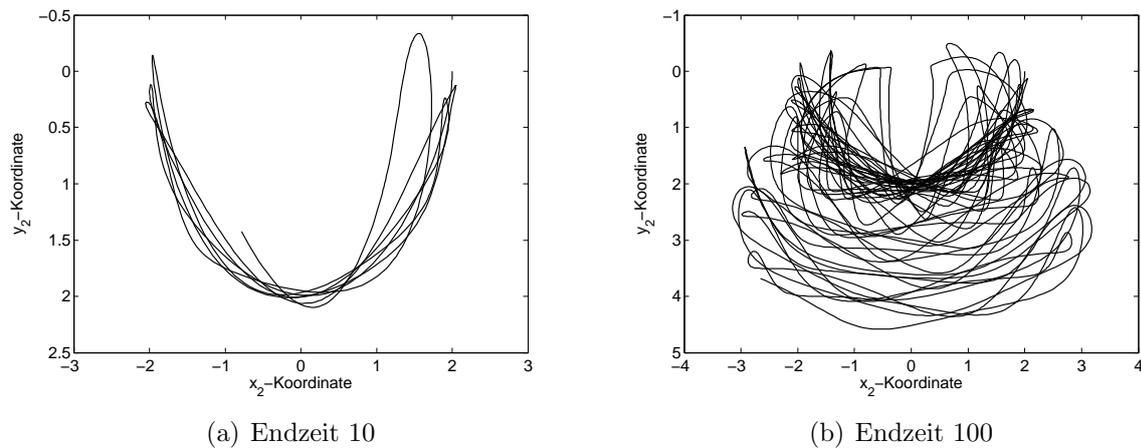


Abbildung 6.8: Numerische Lösung für (x_2, y_2) mit Differentiation der Zwangsbedingung

Aufgrund der Entfernung der numerischen Lösung zur Einheitskreislinie ist diese Methode nicht geeignet für die Simulation der gegebenen DAE.

6.3.3.2 Baumgarte–Methode

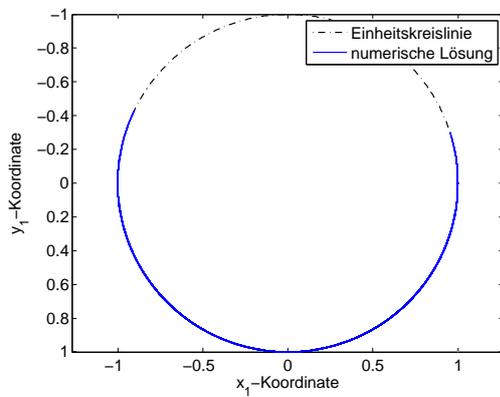
Es wurden die in Tabelle 6.4 aufgelisteten Werte für die Simulationen des mithilfe der Baumgarte–Methode erhaltenen Systems aus Abschnitt 6.2.1.2 verwendet.

Anfangswerte										
	x_1	y_1	v_{x_1}	v_{y_1}	x_2	y_2	v_{x_2}	v_{y_2}	α	β
$\alpha \neq \beta$	1	0	0	0	2	0	0	0	10 bzw. 100	$10 \cdot \alpha$
$\alpha = \beta$	1	0	0	0	2	0	0	0	10 bzw. 100	10 bzw. 100

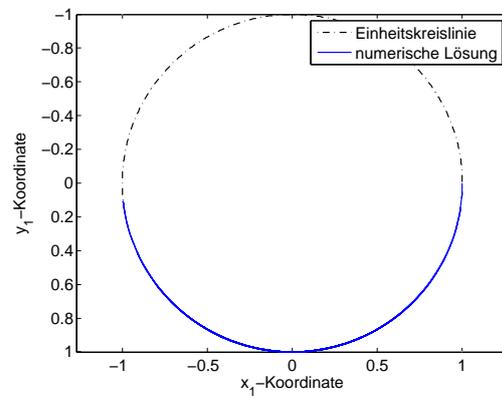
Tabelle 6.4: Anfangswerte für die Simulation der Bewegungsgleichungen des Doppelpendels mit der Baumgarte-Methode

Die Simulationen wurden mit dem ode-Solver `ode45` durchgeführt.

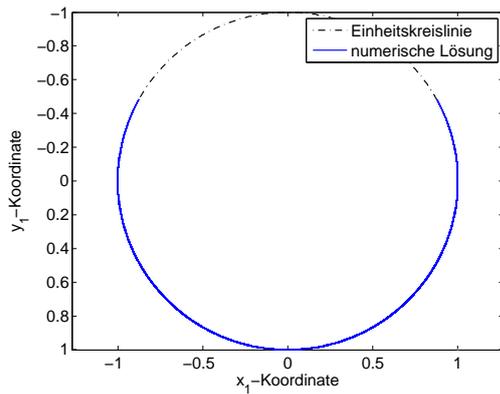
Im Folgenden werden die Ergebnisse, die mit der Baumgarte-Methode und verschiedenen Werten für α und β erhalten werden, dargestellt. In Abbildung 6.9 wird die numerische Lösung der Position des Pendels (x_1, y_1) dargestellt.



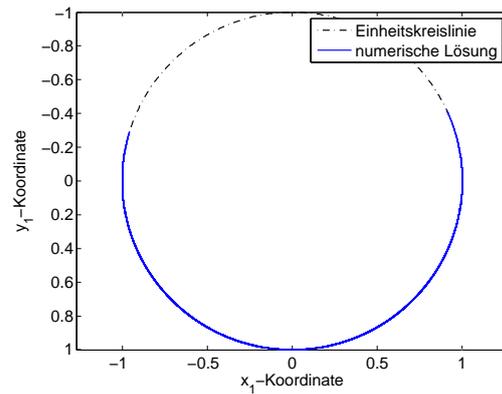
(a) $\alpha = 10$ und $\beta = 100$



(b) $\alpha = \beta = 10$



(c) $\alpha = 100$ und $\beta = 1000$



(d) $\alpha = \beta = 100$

Abbildung 6.9: Numerische Lösung für (x_1, y_1) mit der Baumgarte-Methode und Endzeit 100

Wie in Abbildung 6.9 zu erkennen ist, unterscheiden sich die Ergebnisse für (x_1, y_1) : Wenn $\alpha = 100$ und $\beta = 1000$ ist, schwingt das Pendel (x_1, y_1) am weitesten nach oben, für $\alpha = \beta = 10$ bleibt die numerische Lösung auf der unteren Hälfte der Einheitskreislinie.

In Abbildung 6.10 wird die numerische Lösung der Position des Pendels (x_2, y_2) bis 10 Sekunden in rot und von 10 bis 100 Sekunden in schwarz dargestellt.

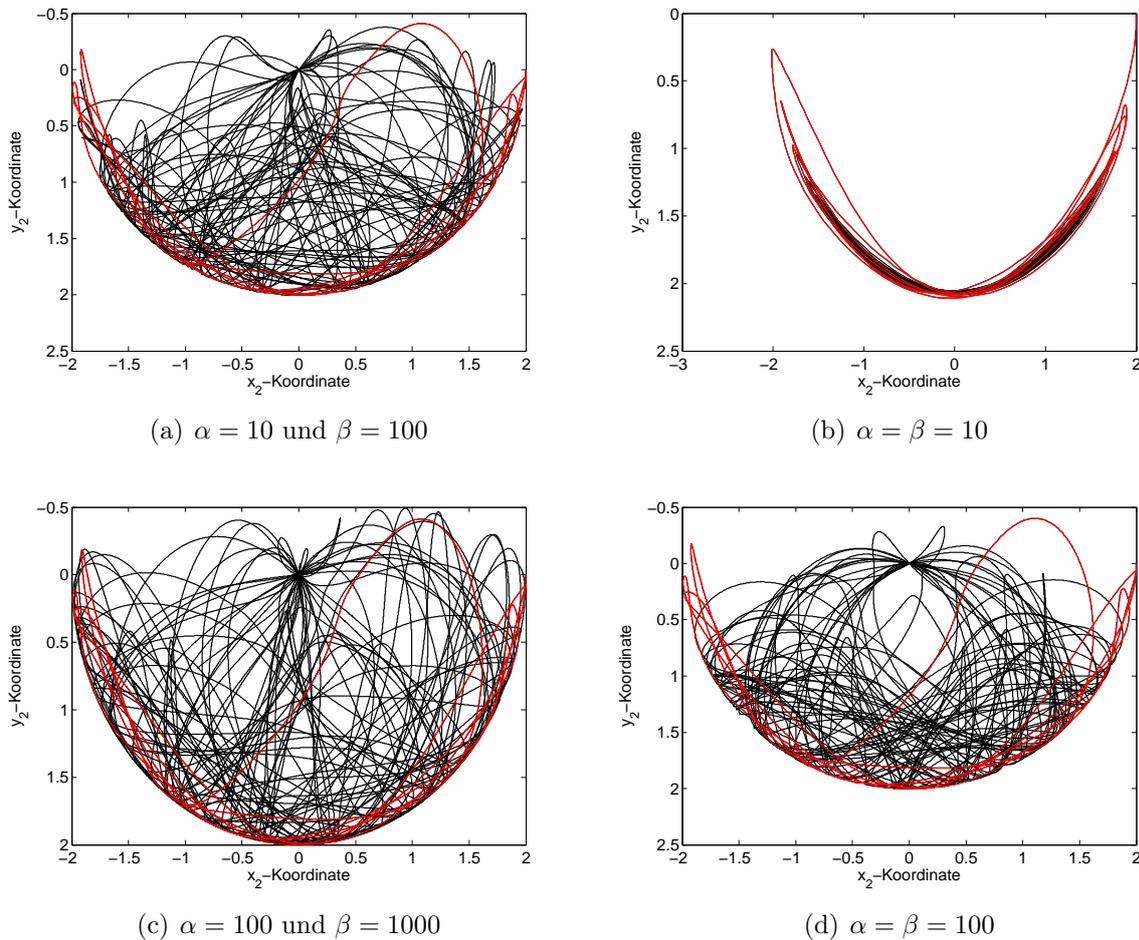
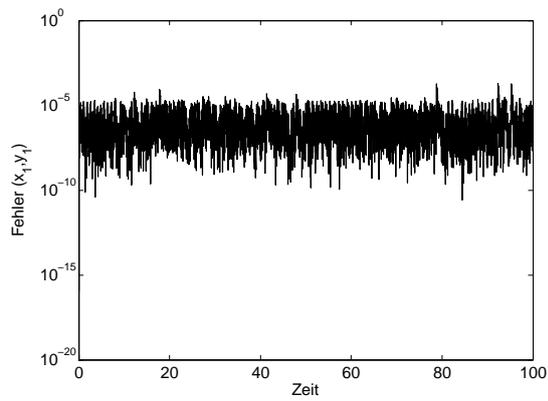


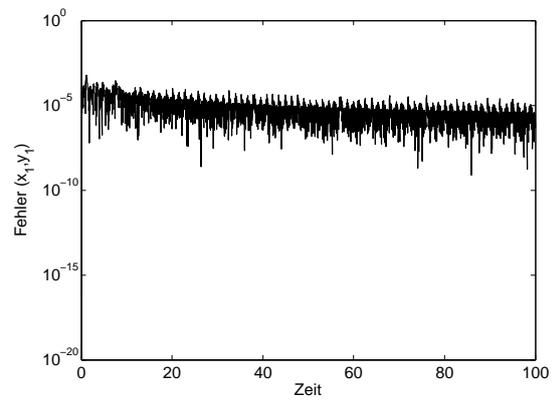
Abbildung 6.10: Numerische Lösung für (x_2, y_2) mit der Baumgarte-Methode und Endzeit 10 (rot) bzw. 100 (schwarz)

Wie in obiger Abbildung zu erkennen ist, unterscheiden sich die Positionen stark je nach Wahl der Parameter α und β . Für $\alpha = \beta = 10$ erscheint die numerische Lösung nicht plausibel.

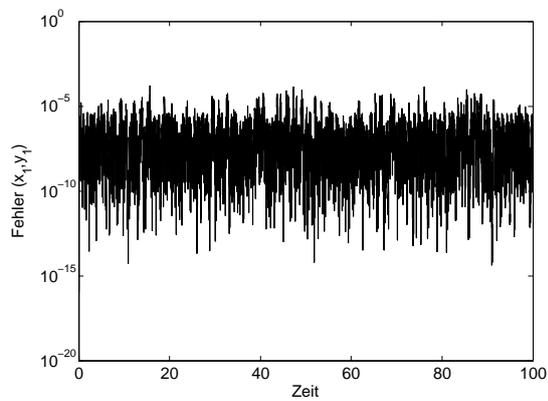
In Abbildung 6.11 werden die mit der Baumgarte-Methode und dem ode-Solver ode45 erhaltenen Fehler von (x_1, y_1) dargestellt.



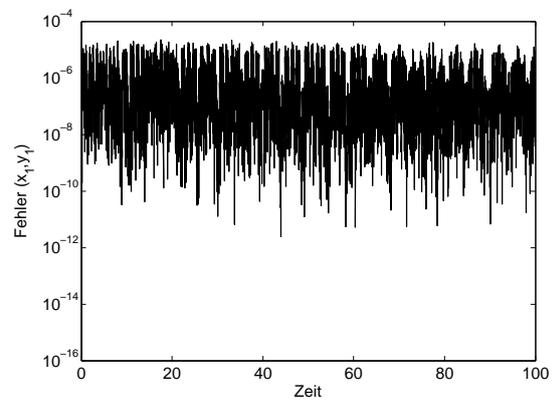
(a) $\alpha = 10$ und $\beta = 100$



(b) $\alpha = \beta = 10$



(c) $\alpha = 100$ und $\beta = 1000$



(d) $\alpha = \beta = 100$

Abbildung 6.11: Fehler der numerischen Lösung von (x_1, y_1) zur Einheitskreislinie mit der Baumgarte-Methode und Endzeit 100

In Abbildung 6.12 werden die mit der Baumgarte-Methode und dem ode-Solver ode45 erhaltenen Fehler von (X, Y) dargestellt.

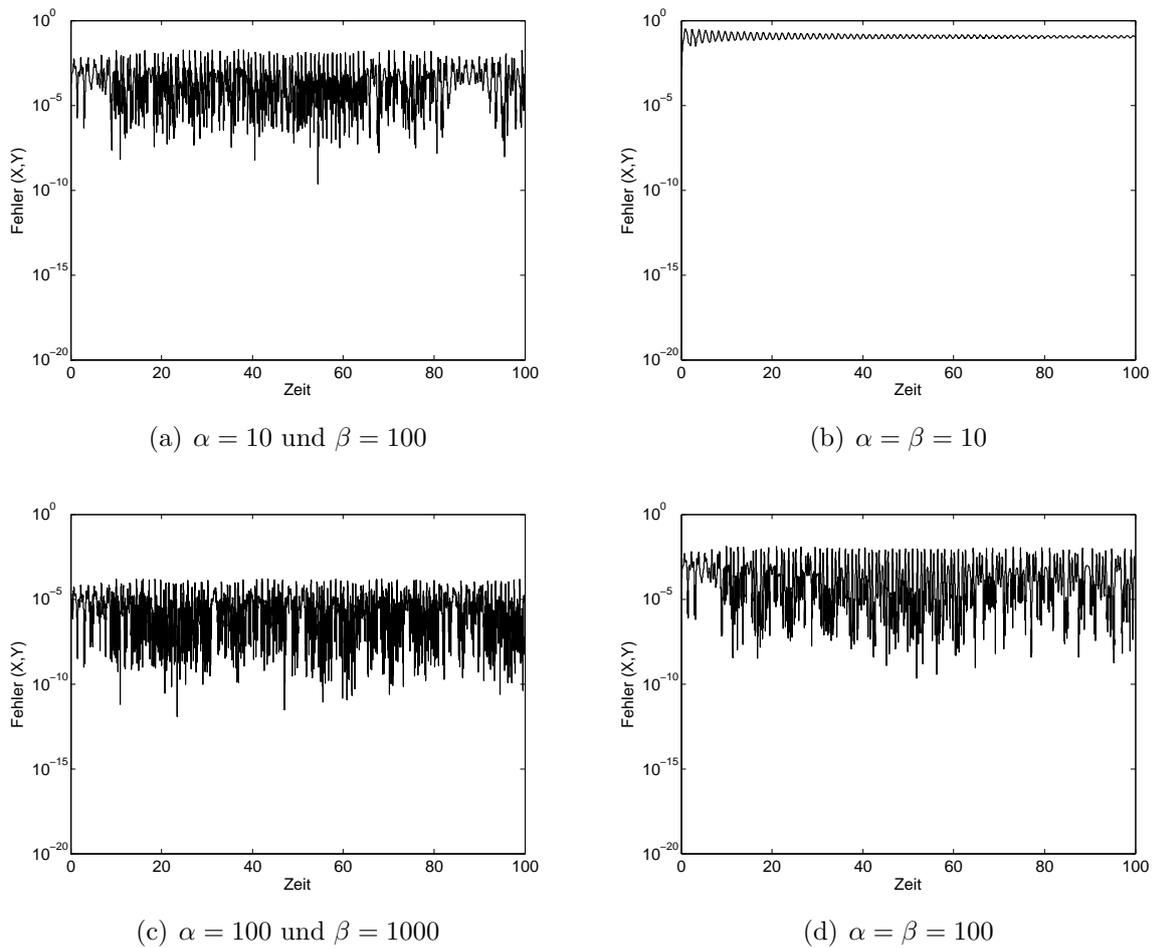


Abbildung 6.12: Fehler der numerischen Lösung von (X, Y) zur Einheitskreislinie mit der Baumgarte-Methode und Endzeit 100

Die Wahl der Parameter α und β ist empfindlich gegenüber Änderungen, das heißt bei unterschiedlicher Wahl der beiden Parameter variieren die Ergebnisse stark. Bei gleicher Wahl von α und β sind die Unterschiede zwischen $\alpha = \beta = 10$ und $\alpha = \beta = 100$ sehr groß, was beispielsweise bei der Darstellung von (x_2, y_2) ersichtlich ist. Wenn $\alpha \neq \beta$ ist, werden nicht so stark unterschiedliche Ergebnisse erhalten, allerdings ist auch hier erkennbar, dass (x_1, y_1) bei größeren Werten von α und β sich im oberen Teil der Einheitskreislinie weiter nach oben bewegt. Da die Ergebnisse für verschiedene α und β sehr unterschiedlich sind und keine analytische Lösung für die Bewegung des Doppelpendels vorliegt, ist es nicht möglich zu bestimmen, welche Lösung die „richtige“ bzw. welche Wahl der Parameter die „beste“ ist.

6.3.3.3 Pantelides–Algorithmus

Es wurden die in Tabelle 6.5 aufgelisteten Werte für die Simulationen der nach Anwendung des Pantelides–Algorithmus erhaltenen Gleichungen aus Abschnitt 6.2.1.3 verwendet.

Anfangswerte										
x_1	$d2x_1$	y_1	dy_1	$d2y_1$	v_{x_1}	v_{y_1}	dv_{y_1}	F_1	dx_1	dv_{x_1}
1	0	0	0	g	0	0	g	0	0	0
x_2	$d2x_2$	y_2	dy_2	$d2y_2$	v_{x_2}	v_{y_2}	dv_{y_2}	F_2	dx_2	dv_{x_2}
2	0	0	0	g	0	0	g	0	0	0

Tabelle 6.5: Anfangswerte für die Simulation der Bewegungsgleichungen des Doppelpendels mit dem Pantelides–Algorithmus

Die Simulation wurde mit dem ode–Solver `ode15i` durchgeführt, wobei `MaxOrder` auf 2 und die absolute Toleranz auf $10^{-1.89}$ gesetzt wurde. Die Wahl dieser beiden Größen hat sehr großen Einfluss auf die numerische Lösung.

Ähnlich wie beim einfachen Pendel mussten die quadratischen Gleichungen aus der Zwangsbedingung händisch aufgelöst werden, wodurch für jede Masse vier Fallunterscheidungen nötig sind. Bei der Implementierung in MATLAB mussten daher $4 \cdot 4 = 16$ Fallunterscheidungen durchgeführt werden, siehe Abschnitt 6.2.1.3. Analog zum einfachen Pendel müssen hier vier Gleichungen und Unbekannte zu den 18 hinzugefügt werden. Somit besteht jedes der 16 Gleichungssysteme aus 22 Gleichungen und Unbekannten.

Um zwischen den Fällen umzuschalten, wurde ein Event (gegeben durch die Aufteilung der beiden Einheitskreise in Quadranten) verwendet, bei dem der ode–Solver abgebrochen wird und dann mit dem neuen Gleichungssystem wieder gestartet wird. An dem Zeitpunkt des Wechsels wurden auch die Anfangswerte neu berechnet. Außerdem wurde der ode–Solver mit neuen Anfangswerten gestartet, sobald er aufgrund der Toleranz eine zu kleine Schrittweite hätte wählen müssen. Die neuen Anfangswerte wurden dabei mit dem Befehl `decic` von MATLAB, bei dem iterativ konsistente Anfangswerte gesucht werden, berechnet. Als Startwert für die Iteration wurde dabei die numerische Lösung zum Abbruchzeitpunkt verwendet.

Durch die vielen Fälle, die bei Anwendung des Pantelides–Algorithmus entstehen, wird die Implementierung aufwändig. Da der verwendete ode–Solver Probleme beim Lösen des Systems hat, muss dieser im Lauf einer Simulation immer wieder abgebrochen und (mit leicht modifizierten Werten) neu gestartet werden, was nicht wünschenswert ist.

In Abbildung 6.13 sind die numerischen Lösungen im Vergleich zur Einheitskreislinie dargestellt.

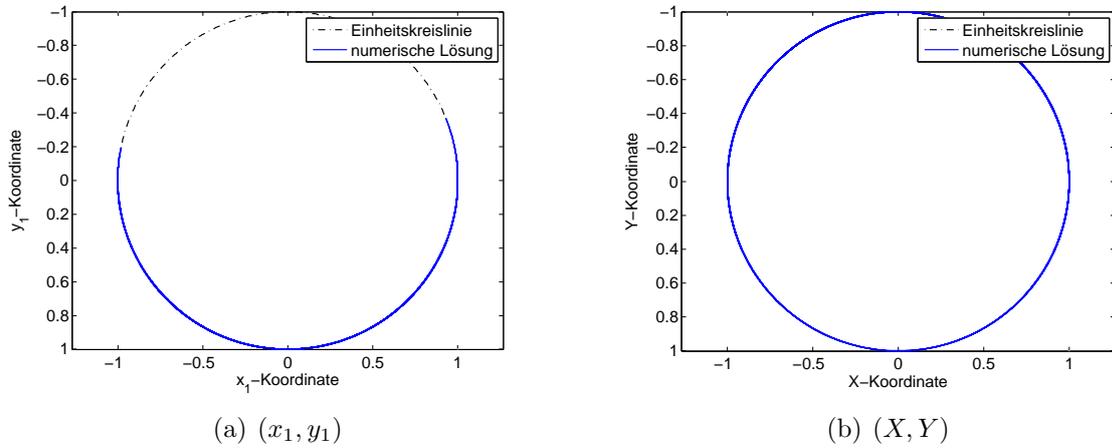


Abbildung 6.13: Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) mit dem Pantelides-Algorithmus und Endzeit 100

In Abbildung 6.14 werden die mit dem Pantelides-Algorithmus erhaltenen Fehler dargestellt.

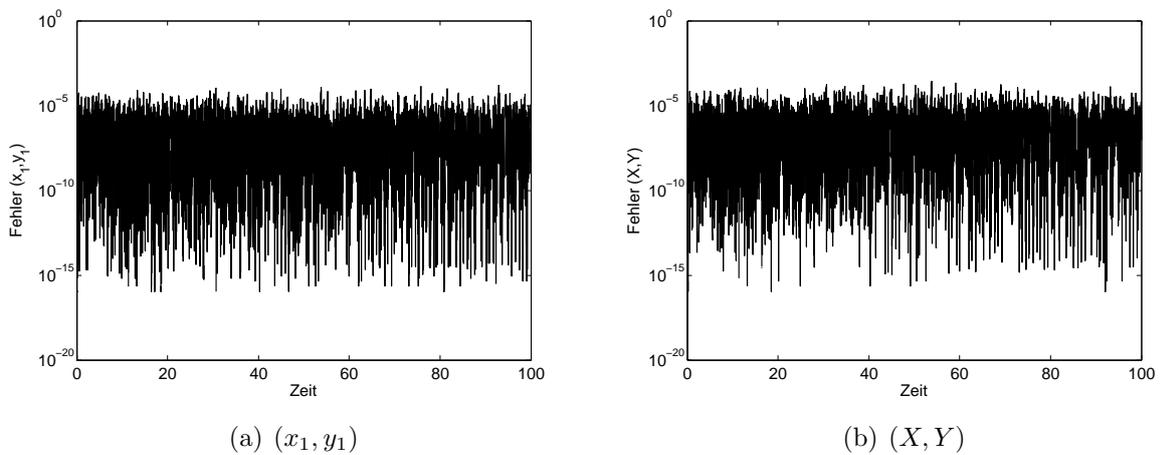


Abbildung 6.14: Fehler der numerischen Lösung zur Einheitskreislinie mit dem Pantelides-Algorithmus und Endzeit 100

In Abbildung 6.15 wird die numerische Lösung der Position des Pendels (x_2, y_2) dargestellt.

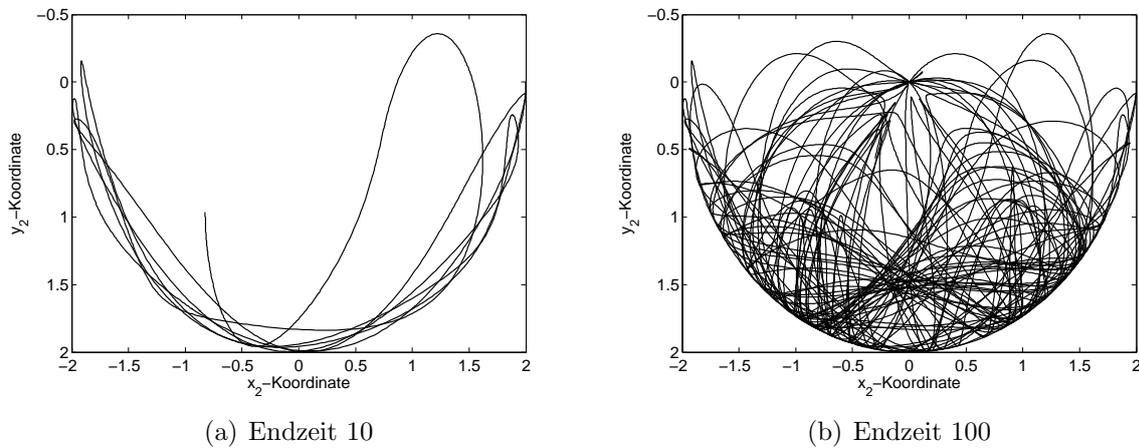


Abbildung 6.15: Numerische Lösung für (x_2, y_2) mit dem Pantelides–Algorithmus

6.3.3.4 Standardprojektionsmethode

Es wurden die in Tabelle 6.6 aufgelisteten Werte für die Simulation unter Verwendung der Standardprojektionsmethode, siehe Abschnitt 6.2.2.1, verwendet.

Anfangswerte							
x_1	y_1	v_{x_1}	v_{y_1}	x_2	y_2	v_{x_2}	v_{y_2}
1	0	0	0	2	0	0	0

Tabelle 6.6: Anfangswerte für die Simulation der Bewegungsgleichungen des Doppelpendels mit der orthogonalen Projektionsmethode

Die Simulation wurde mit dem expliziten Euler–Verfahren und Schrittweite 10^{-3} durchgeführt. Nach jedem Lösungsschritt wird kontrolliert ob die numerische Lösung die Zwangsbedingung erfüllt. Falls dies nicht der Fall ist, wird die Lösung mithilfe der in Abschnitt 6.2.2.1 definierten orthogonalen Projektion auf die Lösungsmannigfaltigkeit abgebildet.

Bei Verwendung dieser Methode ist die euklidische Norm der Geschwindigkeit beider Pendel stark ansteigend und daher werden die numerischen Lösungen der Positionen verfälscht.

In Abbildung 6.16 sind die numerischen Lösungen im Vergleich zur Einheitskreislinie dargestellt.

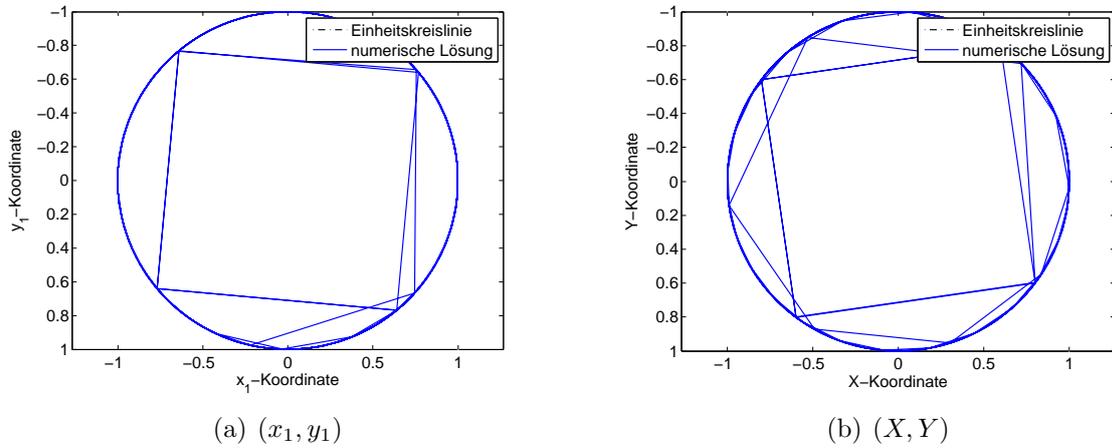


Abbildung 6.16: Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) mit der orthogonalen Projektionsmethode und Endzeit 100

In Abbildung 6.17 werden die mit der orthogonalen Projektionsmethode erhaltenen Fehler dargestellt.

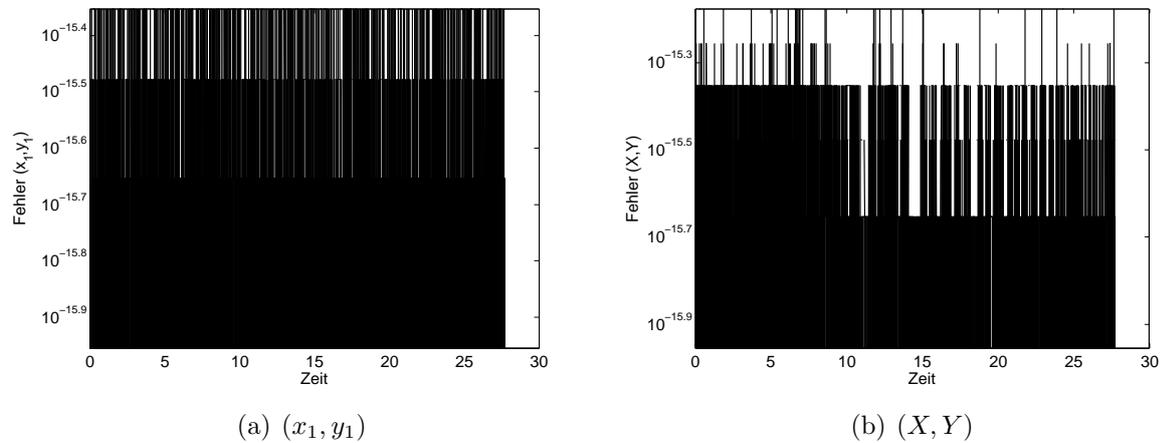


Abbildung 6.17: Fehler der numerischen Lösung zur Einheitskreislinie mit der orthogonalen Projektionsmethode und Endzeit 100

In Abbildung 6.18 werden die mit der orthogonalen Projektionsmethode erhaltenen Normen der Geschwindigkeiten dargestellt.

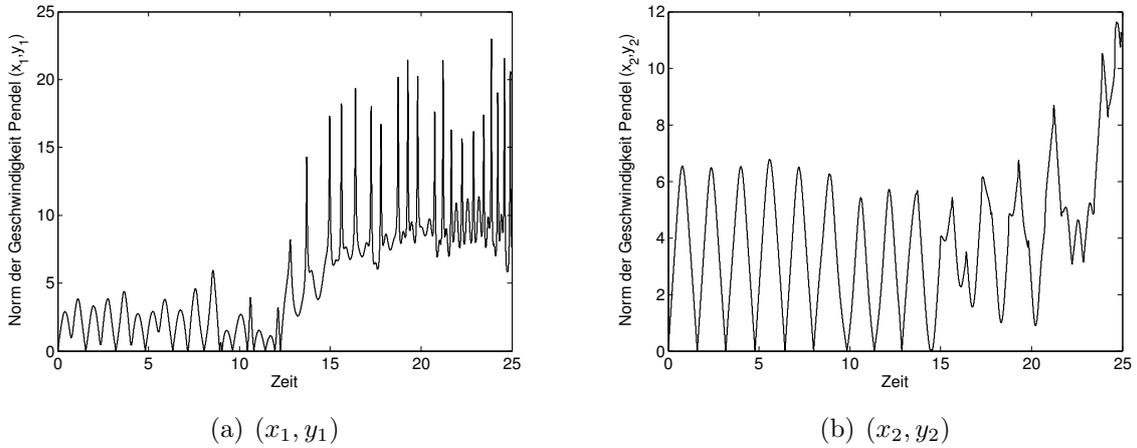


Abbildung 6.18: Euklidische Normen der Geschwindigkeiten mit der orthogonalen Projektionsmethode und Endzeit 25

In Abbildung 6.19 wird die numerische Lösung der Position des Pendels (x_2, y_2) dargestellt.

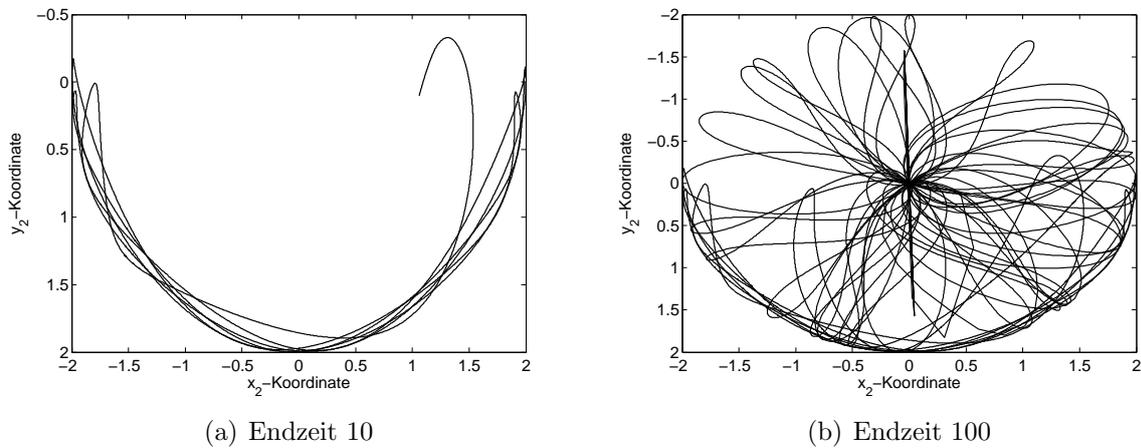


Abbildung 6.19: Numerische Lösung für (x_2, y_2) mit der orthogonalen Projektionsmethode

Die erhaltenen Fehler sind zwar gering, aber die ansteigenden Normen der Geschwindigkeiten deuten darauf hin, dass die Positionen der Pendel nicht korrekt sind. Außerdem liefert das Programm nach etwa 28 Sekunden keine Werte mehr, wie im Fehlerplot ersichtlich ist.

6.3.3.5 Symmetrische Projektionsmethode

Es wurden die in Tabelle 6.7 aufgelisteten Werte für die Simulation unter Zuhilfenahme der symmetrischen Projektionsmethode, siehe Abschnitt 6.2.2.2, verwendet.

Anfangswerte							
x_1	y_1	v_{x_1}	v_{y_1}	x_2	y_2	v_{x_2}	v_{y_2}
1	0	0	0	2	0	0	0

Tabelle 6.7: Anfangswerte für die Simulation der Bewegungsgleichungen des Doppelpendels mit der symmetrischen Projektionsmethode

In Abbildung 6.20 sind die numerischen Lösungen im Vergleich zur Einheitskreislinie dargestellt.

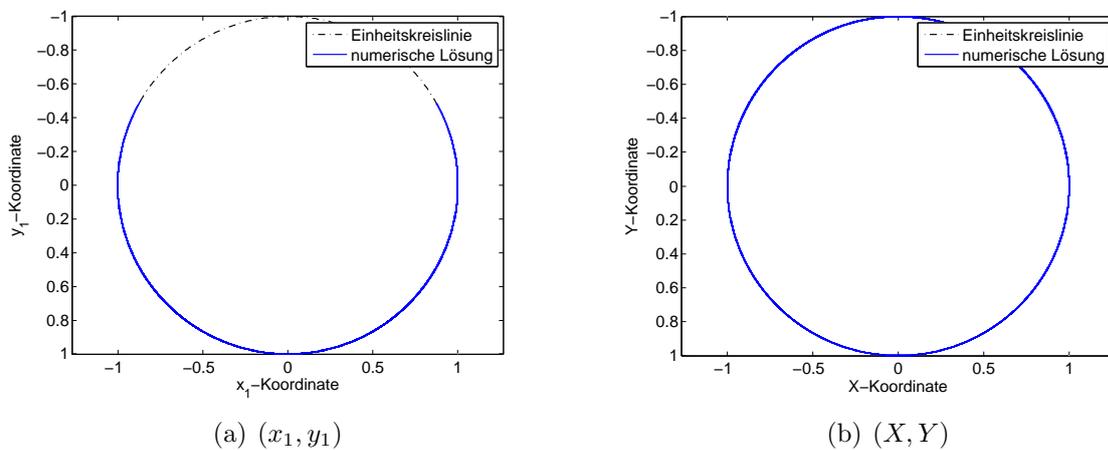


Abbildung 6.20: Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) mit der symmetrischen Projektionsmethode und Endzeit 100

Wie in Abbildung 6.20 zu erkennen ist, unterscheidet sich die Position des Pendels wesentlich von jener bei dem Pantelides–Algorithmus und bei der orthogonalen Projektionsmethode. Die Position des Pendels mit den Koordinaten (x_1, y_1) geht deutlich über den unteren Halbkreis hinaus und steigt höher als bei Verwendung des Pantelides–Algorithmus, allerdings ist kein Überschlag vorhanden, wie es bei der orthogonalen Projektionsmethode der Fall ist.

Daher drängt sich die Frage nach der „richtigen“ Lösung auf. Diese kann nicht beantwortet werden, da keine analytische Lösung vorliegt und die verwendeten Methoden zu unterschiedlichen Ergebnissen führen, die sich auch qualitativ stark unterscheiden.

In Abbildung 6.21 werden die mit der symmetrischen Projektionsmethode erhaltenen Fehler dargestellt.

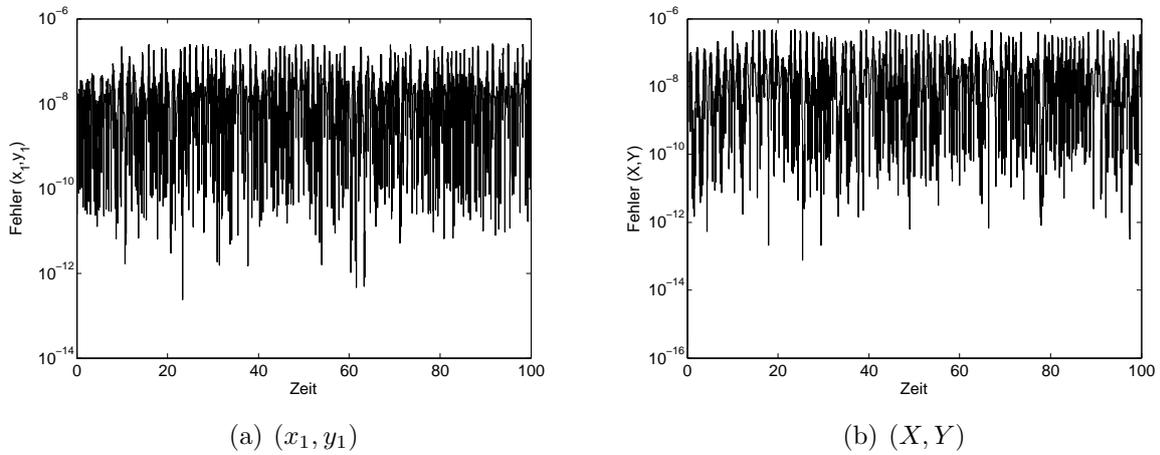


Abbildung 6.21: Fehler der numerischen Lösung zur Einheitskreislinie mit der symmetrischen Projektionsmethode und Endzeit 100

In Abbildung 6.22 werden die mit der symmetrischen Projektionsmethode erhaltenen Normen der Geschwindigkeiten dargestellt.

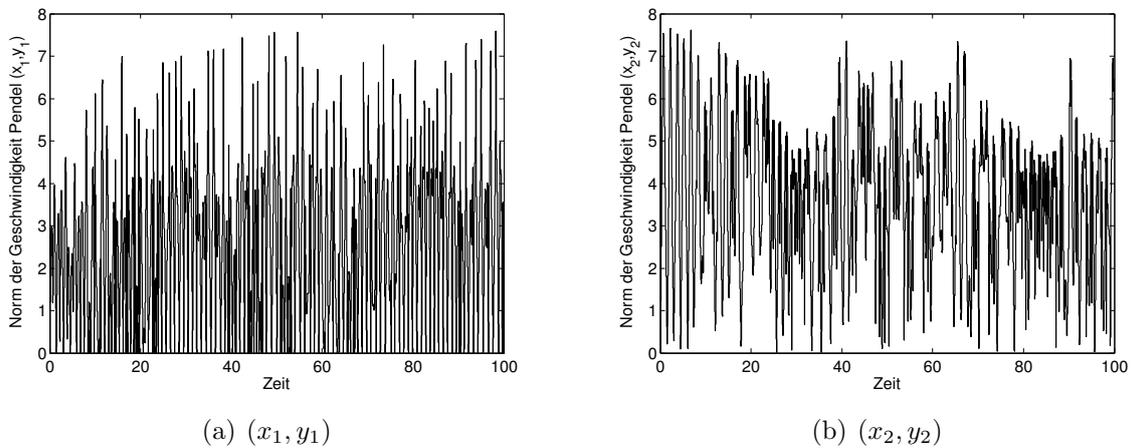


Abbildung 6.22: Euklidische Normen der Geschwindigkeiten mit der symmetrischen Projektionsmethode und Endzeit 100

Wie zu erkennen ist, sind bei dieser Methode die Normen der Geschwindigkeiten im Gegensatz zur orthogonalen Projektionsmethode beschränkt.

In Abbildung 6.23 wird die numerische Lösung der Position des Pendels (x_2, y_2) dargestellt.

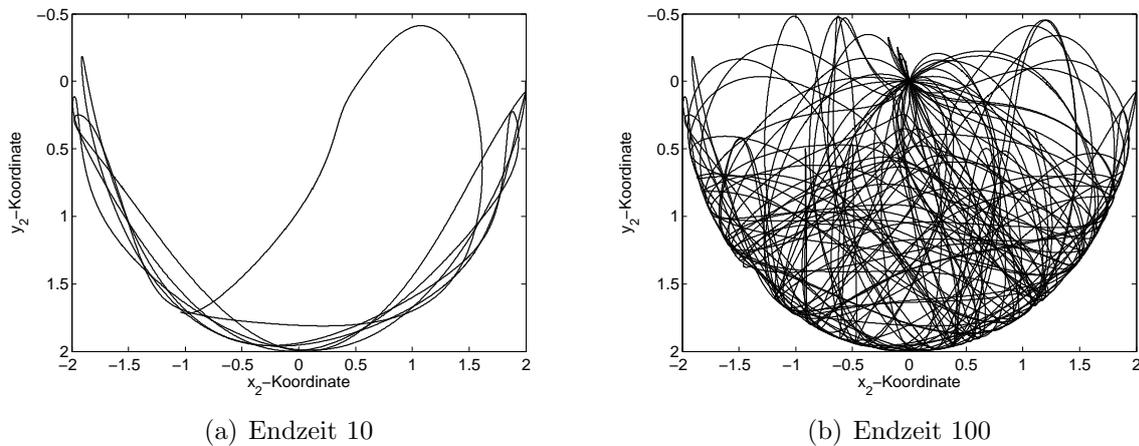


Abbildung 6.23: Numerische Lösung für (x_2, y_2) mit der symmetrischen Projektionsmethode

6.3.3.6 Methoden basierend auf lokaler Zustandsraumtransformation

Es wurden die in Tabelle 6.8 aufgelisteten Werte für die Simulation mit Methoden basierend auf lokaler Zustandsraumtransformation, siehe Abschnitt 6.2.3, verwendet.

Anfangswerte			
φ_1	$\dot{\varphi}_1$	φ_2	$\dot{\varphi}_2$
0	0	0	0

Tabelle 6.8: Anfangswerte für die Simulation der Bewegungsgleichungen des Doppelpendels mit Zustandsraumtransformation

Die Simulation wurde mit den ode-Solvern `ode45`, `23t` und `ode15s` durchgeführt.

Die drei ode-Solver liefern stark unterschiedliche Ergebnisse. Im Gegensatz zu der symmetrischen Projektionsmethode, dem Pantelides-Algorithmus und der Baumgarte-Methode durchläuft bei Verwendung des ode-Solvers `ode45` und Simulation bis 100 Sekunden das Pendel (x_1, y_1) die gesamte Einheitskreislinie. Bei Verwendung der beiden anderen Solver tritt bei Simulation bis 100 Sekunden kein Überschlag für (x_1, y_1) auf. Die numerische Lösung für (x_2, y_2) unterscheidet sich allerdings auch bei diesen beiden Solvern bereits nach 10 Sekunden deutlich.

Im Folgenden werden die Ergebnisse unter Verwendung des `ode23t` dargestellt.

In Abbildung 6.24 sind die numerischen Lösungen im Vergleich zur Einheitskreislinie dargestellt.

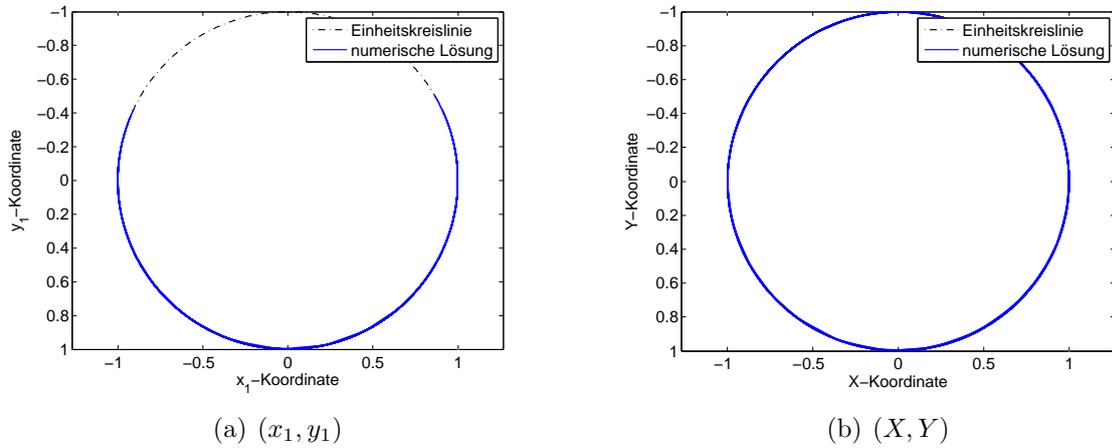


Abbildung 6.24: Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) mit der Zustandsraumtransformation und Endzeit 100

In Abbildung 6.25 werden die mit der Transformation des Zustandsraumes und dem ode-Solver `ode23t` erhaltenen Fehler dargestellt.

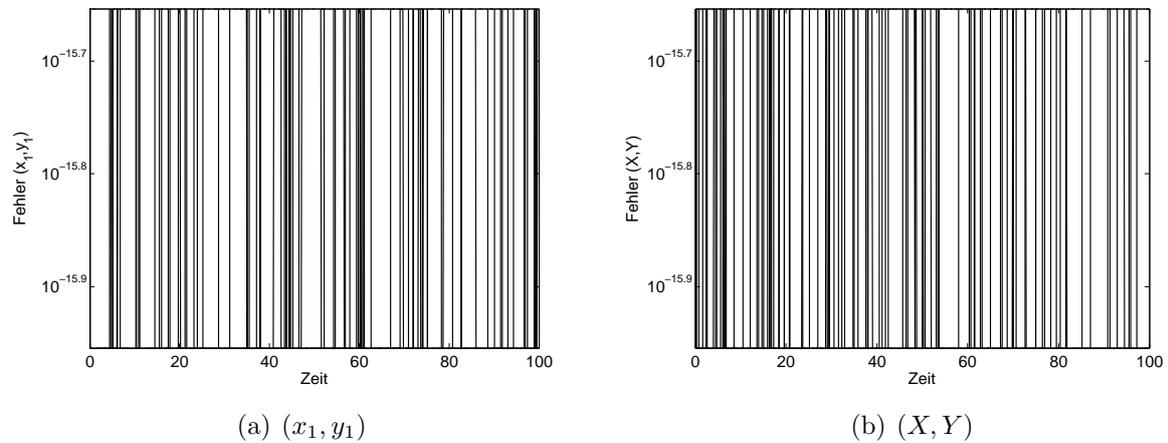


Abbildung 6.25: Fehler der numerischen Lösung zur Einheitskreislinie mit der Zustandsraumtransformation und Endzeit 100

Wie in Abbildung 6.25 zu erkennen ist, sind die Fehler zur Einheitskreislinie gering.

In Abbildung 6.26 wird die numerische Lösung der Position des Pendels (x_2, y_2) dargestellt.

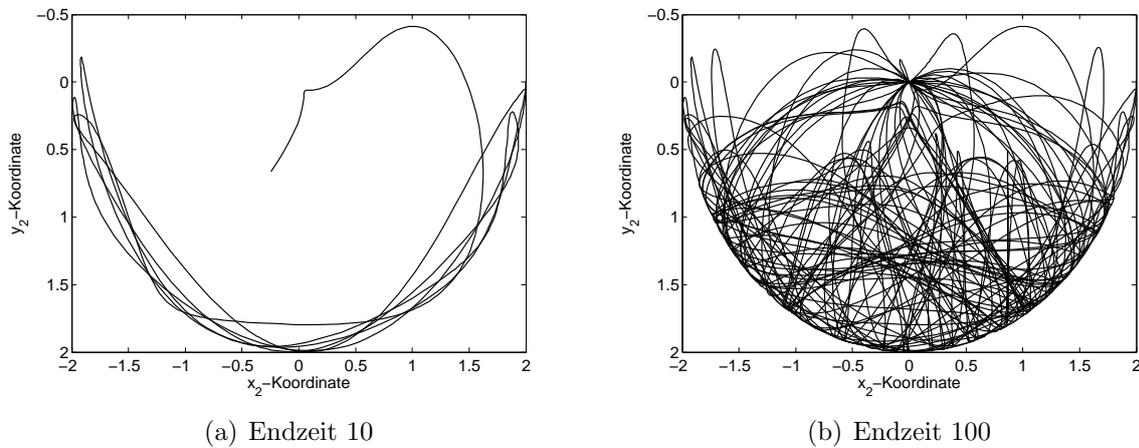


Abbildung 6.26: Numerische Lösung für (x_2, y_2) mit der Zustandsraumtransformation

6.3.4 Ergebnisse

Im Folgenden werden für die Index-1-Formulierung des Systems und die Regularisierungsmethoden Ergebnisse der maximalen Fehler und der Laufzeit in Tabellen aufgelistet. Dabei bezeichnet ein Eintrag mit *, dass der im Programm verwendete ode-Solver nicht bis zur verlangten Endzeit der Simulation lauffähig war.

Da die Simulationen der Index-0-Formulierung bei ungefähr 5 Sekunden abbrechen, werden für diese keine Fehler und Laufzeiten berechnet.

In Tabelle 6.9 sind die maximalen Fehler e_1 und e_2 und die Laufzeit in Sekunden der Simulationen bis $T = 10$ der Index-1-Formulierung aufgelistet.

Lösungsmethode	Max. Fehler e_1	Max. Fehler e_2	Zeit(s)
ode45	0.2603	0.4180	1.588
ode15s	0.1025	0.5489	1.856
ode23t	*	*	*

Tabelle 6.9: Maximale Fehler e_1 und e_2 und Laufzeit in Sekunden der Index-1-Formulierung mit Endzeit 10

Wie Tabelle 6.9 zu entnehmen ist, kommt es bei Verwendung von ode45 und ode15s zu einem numerischen „Drift-off“. Der ode-Solver ode23t bricht bei 9 Sekunden ab.

In Tabelle 6.10 sind die maximalen Fehler e_1 und e_2 und die Laufzeit in Sekunden der Simulationen bis $T = 100$ der Index-1-Formulierung aufgelistet.

Lösungsmethode	Max. Fehler e_1	Max. Fehler e_2	Zeit(s)
ode45	34.1509	70.3141	1.626
ode15s	3.6842	4.5993	3.927
ode23t	*	*	*

Tabelle 6.10: Maximale Fehler e_1 und e_2 und Laufzeit in Sekunden der Index-1-Formulierung mit Endzeit 100

Bei einer Simulation bis 100 Sekunden weicht die numerische Lösung sehr stark von der Lösungsmannigfaltigkeit ab, daher sind die maximalen Fehler groß.

In Tabelle 6.11 sind die maximalen Fehler e_1 und e_2 und die Laufzeit in Sekunden der Simulationen bis $T = 10$ der Indexreduktionsmethoden aufgelistet.

Lösungsmethode	Max. Fehler e_1	Max. Fehler e_2	Zeit(s)
Differentiation Zwangsbedingung	0.0453	0.2792	1.950
Baumgarte-Methode $\alpha = 10, \beta = 100$	$2.5738 \cdot 10^{-5}$	0.0154	1.706
Baumgarte-Methode $\alpha = \beta = 10$	$5.8354 \cdot 10^{-4}$	0.335	1.617
Baumgarte-Methode $\alpha = 100, \beta = 1000$	$2.8281 \cdot 10^{-5}$	$1.3837 \cdot 10^{-4}$	2.550
Baumgarte-Methode $\alpha = \beta = 100$	$2.0022 \cdot 10^{-5}$	0.0130	1.746
Pantelides	$5.5467 \cdot 10^{-5}$	$7.6908 \cdot 10^{-5}$	3.886
orth. Projektionsmethode	$4.4409 \cdot 10^{-16}$	$6.6613 \cdot 10^{-16}$	4.805
symm. Projektionsmethode	$2.1979 \cdot 10^{-7}$	$4.0779 \cdot 10^{-7}$	14.965
Koordinatentransformation ode23t	$2.2204 \cdot 10^{-16}$	$2.2204 \cdot 10^{-16}$	1.707

Tabelle 6.11: Maximale Fehler e_1 und e_2 und Laufzeit in Sekunden der Indexreduktionsmethoden mit Endzeit 10

In Tabelle 6.11 ist zu erkennen, dass die Differentiation der Zwangsbedingung alleine nicht ausreicht, um den numerischen „Drift-off“ zu verhindern. Die orthogonale Projektionsmethode liefert leider keine richtigen Ergebnisse in dem Sinn, dass die Positionen aufgrund der fälschlicherweise ansteigenden Normen der Geschwindigkeiten zwar auf der Kreisbahn bleiben, aber dennoch nicht korrekt sind. Die anderen Methoden liefern durchwegs geringe Abweichungen zur Kreisbahn. Den geringsten Fehler liefert die Koordinatentransformation.

In Tabelle 6.12 sind die maximalen Fehler e_1 und e_2 und die Laufzeit in Sekunden der Simulationen bis $T = 100$ der Indexreduktionsmethoden aufgelistet.

Lösungsmethode	Max. Fehler e_1	Max. Fehler e_2	Zeit(s)
Differentiation Zwangsbedingung	1.0078	9.4948	4.853
Baumgarte-Methode $\alpha = 10, \beta = 100$	$1.9786 \cdot 10^{-4}$	0.0187	3.093
Baumgarte-Methode $\alpha = \beta = 10$	$5.8354 \cdot 10^{-4}$	0.3350	1.761
Baumgarte-Methode $\alpha = 100, \beta = 1000$	$1.6118 \cdot 10^{-4}$	$1.6052 \cdot 10^{-4}$	11.708
Baumgarte-Methode $\alpha = \beta = 100$	$2.2419 \cdot 10^{-5}$	0.0130	3.782
Pantelides	$1.5501 \cdot 10^{-4}$	$2.6766 \cdot 10^{-4}$	21.002
orth. Projektionsmethode	$4.4409 \cdot 10^{-16}$	$6.6613 \cdot 10^{-16}$	166.633
symm. Projektionsmethode	$2.5376 \cdot 10^{-7}$	$4.7636 \cdot 10^{-7}$	129.584
Koordinatentransformation <code>ode23t</code>	$2.2204 \cdot 10^{-16}$	$2.2204 \cdot 10^{-16}$	3.659

Tabelle 6.12: Maximale Fehler e_1 und e_2 und Laufzeit in Sekunden der Indexreduktionsmethoden mit Endzeit 100

Die Simulation bis $T = 100$ liefert analoge Resultate zu den Ergebnissen der Simulation bis 10 Sekunden. Die größeren Laufzeiten der orthogonalen Projektionsmethode und der symmetrischen Projektionsmethode sind darauf zurückzuführen, dass bei der einen ein selbstimplementiertes explizites Euler-Verfahren verwendet wird und bei der anderen die numerische Lösung iterativ berechnet wird. Der Pantelides-Algorithmus besitzt aufgrund der vielen Fallunterscheidungen eine längere Laufzeit als die anderen Programme, die ode-Solver verwenden.

Zusammenfassung der Ergebnisse:

Das numerische Lösen der Index-0-Formulierung bzw. der Index-1-Formulierung liefert keine brauchbaren Ergebnisse. Es führen alle verwendeten ode-Solver zum „Drift-off“ oder die Integration bricht ab.

Die Differentiation der Zwangsbedingung führt zum „Drift-off“ und ist daher nicht geeignet für das Lösen der DAE. Die Baumgarte-Methode liefert für geeignetes α und β Resultate, die geringe Fehler besitzen. Es sind große Unterschiede der Lösungen für verschiedene Werte der Parameter α und β zu beobachten. Die Implementierung des Pantelides-Algorithmus ist durch die Fallunterscheidungen sehr umständlich und der `ode15i` hat Probleme, die Gleichungen zu lösen.

Die orthogonale Projektionsmethode liefert Ergebnisse, die ebenfalls geringe Fehler aufweisen, dies ist auch nicht überraschend, da die Methode genau so konstruiert ist. Allerdings sind die Positionen nicht korrekt, was auf die steigenden Normen der Geschwindigkeiten zurückgeführt werden kann. Im Gegensatz dazu liefert die symmetrische Projektionsmethode keine unbeschränkten Normen der Geschwindigkeiten.

Die Transformation des Zustandsraumes führt auf das einfachste zu lösende System und liefert ein Ergebnis, das nahe an der Lösungsmannigfaltigkeit bleibt.

Die Positionen beider Pendel besitzen für die Baumgarte-Methode, den Pantelides-

Algorithmus, die symmetrische Projektionsmethode und die Transformation auf Polarkoordinaten unterschiedliche Ergebnisse – da keine analytische Lösung bekannt ist, ist nicht klar, welche der Methoden die „richtigen“ Positionen liefert.

Zusammenfassung

In dieser Arbeit wurden verschiedene Regularisierungsmethoden an zwei Fallbeispielen demonstriert. Bei den beiden Beispielen handelt es sich um mechanische Systeme mit differentielltem Index drei: ein einfaches Pendel und ein Doppelpendel. Die Analyse und das Anwenden der Regularisierungsmethoden zeigt, dass die Berechnungen beim einfachen Pendel leicht durchzuführen sind, wohingegen beim Doppelpendel die Gleichungen lange und unübersichtlich werden, was bei „händischem“ Anwenden der Methoden sehr leicht zu Fehlern führen kann. Beim einfachen Pendel wurden qualitativ meist ähnliche Resultate erhalten. Im Gegensatz dazu ist das beim Doppelpendel nicht der Fall, hier unterscheiden sich die Ergebnisse der verschiedenen Methoden sehr stark aufgrund des chaotischen Verhaltens, das das Doppelpendel besitzt.

Nach der Analyse der Methoden anhand der Fallbeispiele in den vorherigen Kapiteln kann zusammenfassend darauf geschlossen werden, dass die Simulationen der Index-0-Formulierung und der Index-1-Formulierung im Allgemeinen zu Ergebnissen führen, die von der Lösungsmannigfaltigkeit abweichen, das heißt zum numerischen „Drift-off“ führen. Daher erweist es sich als sinnvoll, Regularisierungsmethoden bzw. Indexreduktionsmethoden zu betrachten.

Die erste betrachtete Kategorie waren Indexreduktionsmethoden mithilfe von Differentiation. Die zweimalige Differentiation und anschließendes Ersetzen der Zwangsbedingung führt zu keinen brauchbaren Ergebnissen. Diese Methode kann implementiert werden, indem ein ode-Solver verwendet wird, der für differential-algebraische Gleichungssysteme mit differentielltem Index 1 geeignet ist. Im Gegensatz dazu liefert die Baumgarte-Methode für geeignete α und β numerische Lösungen, die keinen numerischen „Drift-off“ aufweisen und daher ist diese Methode zum Lösen von differential-algebraischen Gleichungssystemen geeignet. Bei dieser Methode besteht die Schwierigkeit in der Wahl dieser beiden Parameter. Die Implementierung gestaltet sich sehr einfach, da nur ein gewöhnliches Differentialgleichungssystem gelöst werden muss. Der Pantelides-

Algorithmus bleibt zwar auf der Lösungsmannigfaltigkeit, allerdings werden bei dieser Methode viele Gleichungen und Unbekannte erhalten. Die erhaltenen Gleichungen müssen aufgrund des Auftretens von quadratischen Gleichungen mittels Fallunterscheidungen in verschiedene Gleichungssysteme aufgeteilt werden. Daher ist die Implementierung umständlich. Außerdem werden bei einer anderen Wahl von `MaxOrder` und `AbsTol` bei den Optionen des `ode`-Solvers andere Ergebnisse erhalten. Allerdings ist diese Methode die einzige der hier vorgestellten, bei der der differentielle Index des differentialalgebraischen Gleichungssystems nicht im Vorhinein bekannt sein muss, da der Algorithmus von selbst abbricht, sobald das System Index 1 besitzt.

Die nächste Kategorie an Methoden sind Projektionsmethoden, wobei die orthogonale Projektionsmethode und die symmetrische Projektionsmethode betrachtet wurden. Der Nachteil der Projektionsmethoden ist, dass die Lösungsmannigfaltigkeit erst gefunden werden muss. Bei der orthogonalen Projektionsmethode muss die verwendete Projektion so berechnet werden, dass sie orthogonal auf die Mannigfaltigkeit ist. Die orthogonale Projektionsmethode hat leider für beide Fallbeispiele keine brauchbaren Resultate geliefert. Die Implementierung dieser Methode verwendet ein explizites Euler-Verfahren. Im Gegensatz zur orthogonalen Projektionsmethode ist die Norm der Geschwindigkeit bei der symmetrischen Projektionsmethode beschränkt und die numerische Lösung bleibt auf der Lösungsmannigfaltigkeit. Die Implementierung der symmetrischen Projektionsmethode lässt sich mithilfe der Trapezregel und einer `while`-Schleife umsetzen.

Die letzte Methode ist die (lokale) Zustandsraumtransformation. Bei den beiden Fallbeispielen kann der Zustandsraum sogar global transformiert werden. Eine globale Transformation kann das gegebene Gleichungssystem in ein gewöhnliches Differentialgleichungssystem, das eine wesentlich einfachere Struktur besitzt, umwandeln. Beim einfachen Pendel ist dieser Umstand eindeutig erkennbar, beim Doppelpendel ist trotz der komplexeren Gleichungen das transformierte System dennoch das einfachste. Die Schwierigkeit dieser Methode ist, die Lösungsmannigfaltigkeit und eine geeignete Transformation der Lösungsmannigfaltigkeit, wenn möglich global, zu finden. In MATLAB kann das sehr einfach implementiert werden mittels eines `ode`-Solvers.

Es sind verschiedene Erweiterungen zu dieser Arbeit denkbar:

Eine Verfeinerung einiger verwendeter Algorithmen der verschiedenen Methoden ist möglich. So kann zum Beispiel beim Pantelides-Algorithmus ein eigener Solver implementiert werden, da die Implementierung mit dem ode-Solver `ode15i` in MATLAB sehr umständlich ist und beim Doppelpendel der ode-Solver sogar abgebrochen werden muss. Weiters kann bei der orthogonalen Projektionsmethode ein Solver höherer Ordnung verwendet werden, um zu überprüfen, ob die Norm der Geschwindigkeit ebenfalls ansteigend ist bzw. das qualitative Verhalten der numerischen Lösung gleich bleibt.

Als weitere Aufgabenstellung können die in dieser Arbeit vorgestellten Ansätze an anderen Beispielen getestet werden, um genauer festzustellen, welche Methode für welche Problemstellung geeignet ist. Die Methoden sollten (sofern es möglich ist) auch an Beispielen, die nicht Index drei besitzen, getestet werden. Um die Methode mit lokaler Zustandsraumtransformation besser auf die Probe stellen zu können, ist es denkbar, ein Beispiel zu wählen, bei dem keine globale Transformation möglich ist.

Eine andere interessante Aufgabenstellung ist, die vorgestellten Methoden noch mit weiteren in der Literatur vorkommenden Methoden, die in dieser Arbeit nicht vorkommen, zu vergleichen. Diese Methoden sind oft, ähnlich wie die Baumgarte-Methode, nur für spezielle differential-algebraische Gleichungssysteme geeignet und können daher nicht für jede Problemstellung verwendet werden.

Eine weitere Erweiterung dieser Arbeit wäre, die hier vorgestellten Methoden und deren numerische Lösung mit Simulatoren, die auf das Lösen von differential-algebraischen Gleichungssystemen spezialisiert sind, zu vergleichen.

Abbildungsverzeichnis

4.1	Illustration der Standardprojektionsmethode (Quelle: [12])	31
4.2	Illustration der symmetrischen Projektionsmethode (Quelle: [12])	33
4.3	Illustration der Tangentialraumkoordinatisierung (Quelle: [12])	36
5.1	Schaubild und Gleichungen des Pendels (Quelle: [24])	37
5.2	Einheitskreislinie mit der entsprechenden Aufteilung für das Lösen der DAE mithilfe des Pantelides-Algorithmus	46
5.3	Orthogonale Projektion	51
5.4	Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) für Szenario 1 der Index-0-Formulierung mit <code>ode45</code> und <code>ode15s</code> und Endzeit 10	56
5.5	Fehler der numerischen Lösung zur Einheitskreislinie für Szenario 1 der Index-0-Formulierung mit <code>ode45</code> und <code>ode15s</code> und Endzeit 10	56
5.6	Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) von der Simulation der Index-0-Formulierung mit der Trapezregel und Endzeit 100	57
5.7	Fehler der numerischen Lösung zur Einheitskreislinie von der Simulation der Index-0-Formulierung mit der Trapezregel und Endzeit 100	58
5.8	Szenario 2 der Index-1-Formulierung mit <code>ode23t</code> und Endzeit 500	59
5.9	Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) der Index-1-Formulierung mit der Trapezregel und Endzeit 100	59
5.10	Fehler der numerischen Lösung zur Einheitskreislinie der Index-1-Formulierung mit der Trapezregel und Endzeit 100	60
5.11	Szenario 2 mit Differentiation der Zwangsbedingung, <code>ode15s</code> und Endzeit 100	61
5.12	Szenario 2 mit der Baumgarte-Methode mit $\alpha \neq \beta$ und Endzeit 100	62

5.13	Szenario 2 mit der Baumgarte–Methode mit $\alpha = \beta$ und Endzeit 100 . . .	62
5.14	Einheitskreislinie mit den Bereichen für die entsprechenden Gleichungssysteme des Pantelides–Algorithmus	64
5.15	Szenario 1 mit dem Pantelides–Algorithmus und Endzeit 100	65
5.16	Szenario 2 mit dem Pantelides–Algorithmus und Endzeit 100	65
5.17	Szenario 2 mit der orthogonalen Projektionsmethode und Endzeit 100 . .	66
5.18	Norm der Geschwindigkeit des Pendels von Szenario 2 bei der orthogonalen Projektionsmethode mit Endzeit 5	67
5.19	Szenario 1 mit der orthogonalen Projektionsmethode und Endzeit 100 . .	67
5.20	Norm der Geschwindigkeit des Pendels von Szenario 1 bei der orthogonalen Projektionsmethode mit Endzeit 100	68
5.21	Szenario 2 mit der symmetrischen Projektionsmethode und Endzeit 100 .	69
5.22	Norm der Geschwindigkeit des Pendels von Szenario 2 bei der symmetrischen Projektionsmethode mit Endzeit 100	69
5.23	Szenario 2 mit der Zustandsraumtransformation und Endzeit 100	70
6.1	Schaubild und Gleichungen des Doppelpendels	75
6.2	Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) der Index–0–Formulierung der Bewegungsgleichungen des Doppelpendels bis zum Abbruch des Programmes	89
6.3	Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) der Index–1–Formulierung der Bewegungsgleichungen des Doppelpendels mit <code>ode45</code> und Endzeit 10	90
6.4	Fehler der numerischen Lösung zur Einheitskreislinie der Index–1–Formulierung der Bewegungsgleichungen des Doppelpendels mit <code>ode45</code> und Endzeit 10	91
6.5	Numerische Lösung für (x_2, y_2) der Index–1–Formulierung der Bewegungsgleichungen des Doppelpendels mit <code>ode45</code> und Endzeit 10	91
6.6	Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) mit Differentiation der Zwangsbedingung und Endzeit 100	92
6.7	Fehler der numerischen Lösung zur Einheitskreislinie mit Differentiation der Zwangsbedingung und Endzeit 100	93
6.8	Numerische Lösung für (x_2, y_2) mit Differentiation der Zwangsbedingung	93
6.9	Numerische Lösung für (x_1, y_1) mit der Baumgarte–Methode und Endzeit 100	94
6.10	Numerische Lösung für (x_2, y_2) mit der Baumgarte–Methode und Endzeit 10 (rot) bzw. 100 (schwarz)	95
6.11	Fehler der numerischen Lösung von (x_1, y_1) zur Einheitskreislinie mit der Baumgarte–Methode und Endzeit 100	96

6.12 Fehler der numerischen Lösung von (X, Y) zur Einheitskreislinie mit der Baumgarte–Methode und Endzeit 100	97
6.13 Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) mit dem Pantelides–Algorithmus und Endzeit 100	99
6.14 Fehler der numerischen Lösung zur Einheitskreislinie mit dem Pantelides–Algorithmus und Endzeit 100	99
6.15 Numerische Lösung für (x_2, y_2) mit dem Pantelides–Algorithmus	100
6.16 Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) mit der orthogonalen Projektionsmethode und Endzeit 100	101
6.17 Fehler der numerischen Lösung zur Einheitskreislinie mit der orthogonalen Projektionsmethode und Endzeit 100	101
6.18 Euklidische Normen der Geschwindigkeiten mit der orthogonalen Projektionsmethode und Endzeit 25	102
6.19 Numerische Lösung für (x_2, y_2) mit der orthogonalen Projektionsmethode	102
6.20 Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) mit der symmetrischen Projektionsmethode und Endzeit 100	103
6.21 Fehler der numerischen Lösung zur Einheitskreislinie mit der symmetrischen Projektionsmethode und Endzeit 100	104
6.22 Euklidische Normen der Geschwindigkeiten mit der symmetrischen Projektionsmethode und Endzeit 100	104
6.23 Numerische Lösung für (x_2, y_2) mit der symmetrischen Projektionsmethode	105
6.24 Vergleich der numerischen Lösung (blau) zur Einheitskreislinie (schwarz) mit der Zustandsraumtransformation und Endzeit 100	106
6.25 Fehler der numerischen Lösung zur Einheitskreislinie mit der Zustandsraumtransformation und Endzeit 100	106
6.26 Numerische Lösung für (x_2, y_2) mit der Zustandsraumtransformation . .	107

Tabellenverzeichnis

5.1	Links: Index-0-Formulierung der Bewegung des Pendels; Rechts: Index-1-Formulierung der Bewegung des Pendels	40
5.2	Anfangswerte für die Simulation der Bewegungsgleichungen des Pendels mit Index 0	55
5.3	Anfangswerte für die Simulation der Bewegungsgleichungen des Pendels mit Index 1	58
5.4	Anfangswerte für die Simulation der Bewegungsgleichungen des Pendels mit der Indexreduktionsmethode Differentiation der Zwangsbedingung . .	60
5.5	Anfangswerte für die Simulation der Bewegungsgleichungen des Pendels mit der Baumgarte-Methode	61
5.6	Die vier erhaltenen Gleichungssysteme bei Verwendung des Pantelides-Algorithmus	63
5.7	Anfangswerte für die Simulation der Bewegungsgleichungen des Pendels mit dem Pantelides-Algorithmus	64
5.8	Anfangswerte für die Simulation der Bewegungsgleichungen des Pendels mit der orthogonalen Projektionsmethode	66
5.9	Anfangswerte für die Simulation der Bewegungsgleichungen des Pendels mit der symmetrischen Projektionsmethode	68
5.10	Anfangswerte für die Simulation der Bewegungsgleichungen des Pendels mit Zustandsraumtransformation	70
5.11	Maximaler Fehler e und Laufzeit in Sekunden der Index-0-Formulierung mit Endzeit 10	71
5.12	Maximaler Fehler e und Laufzeit in Sekunden der Index-0-Formulierung mit Endzeit 100	71
5.13	Maximaler Fehler e und Laufzeit in Sekunden der Index-1-Formulierung mit Endzeit 10	72

5.14	Maximaler Fehler e und Laufzeit in Sekunden der Index-1-Formulierung mit Endzeit 100	72
5.15	Maximaler Fehler e und Laufzeit in Sekunden der Indexreduktionsmethoden mit Endzeit 10	73
5.16	Maximaler Fehler e und Laufzeit in Sekunden der Indexreduktionsmethoden mit Endzeit 100	73
6.1	Anfangswerte für die Simulation der Bewegungsgleichungen des Doppelpendels mit Index 0	89
6.2	Anfangswerte für die Simulation der Bewegungsgleichungen des Doppelpendels mit Index 1	90
6.3	Anfangswerte für die Simulation der Bewegungsgleichungen des Doppelpendels mit der Indexreduktionsmethode Differentiation der Zwangsbedingung	92
6.4	Anfangswerte für die Simulation der Bewegungsgleichungen des Doppelpendels mit der Baumgarte-Methode	94
6.5	Anfangswerte für die Simulation der Bewegungsgleichungen des Doppelpendels mit dem Pantelides-Algorithmus	98
6.6	Anfangswerte für die Simulation der Bewegungsgleichungen des Doppelpendels mit der orthogonalen Projektionsmethode	100
6.7	Anfangswerte für die Simulation der Bewegungsgleichungen des Doppelpendels mit der symmetrischen Projektionsmethode	103
6.8	Anfangswerte für die Simulation der Bewegungsgleichungen des Doppelpendels mit Zustandsraumtransformation	105
6.9	Maximale Fehler e_1 und e_2 und Laufzeit in Sekunden der Index-1-Formulierung mit Endzeit 10	107
6.10	Maximale Fehler e_1 und e_2 und Laufzeit in Sekunden der Index-1-Formulierung mit Endzeit 100	108
6.11	Maximale Fehler e_1 und e_2 und Laufzeit in Sekunden der Indexreduktionsmethoden mit Endzeit 10	108
6.12	Maximale Fehler e_1 und e_2 und Laufzeit in Sekunden der Indexreduktionsmethoden mit Endzeit 100	109

Literaturverzeichnis

- [1] Stolze, J.: *Einführung in die nichtlineare Dynamik*. http://t1.physik.tu-dortmund.de/stolze/teaching/TNF/tnf_chaos.pdf, Zugriff: 24.06.2014.
- [2] Kaltenbäck, M.: *Analysis 1*. Skriptum zur Vorlesung, Technische Universität Wien, 2010.
- [3] Kaltenbäck, M.: *Analysis 2*. Skriptum zur Vorlesung, Technische Universität Wien, 2012.
- [4] Kaltenbäck, M.: *Analysis 3*. Skriptum zur Vorlesung, Technische Universität Wien, 2011.
- [5] Melenk, J.M.: *Numerik*. Skriptum zur Vorlesung, Technische Universität Wien, 2008.
- [6] Drmota, M.: *Lineare Algebra 2*. Skriptum zur Vorlesung, Technische Universität Wien, 2006.
- [7] Havlicek, H.: *Lineare Algebra für Technische Mathematiker*. Heldermann Verlag, 2006.
- [8] Ganster, M.: *Definition eines Tensors*. http://www.math.tugraz.at/~ganster/lv_vektoranalysis_ss_10/17_tensor_definition.pdf, Zugriff: 30.04.2014.
- [9] Straumann, U.: *Einführung in die Tensorrechnung*. http://www.physik.uzh.ch/~strauman/physik-a/Ergaenzung_Tensoren.pdf, Zugriff: 24.06.2014.
- [10] Melenk, J.M.: *Numerik von Differentialgleichungen*. Skriptum zur Vorlesung, Technische Universität Wien, 2009.

- [11] Arnold, A.: *Variationsrechnung*. Skriptum zur Vorlesung, Technische Universität Wien, 2013.
- [12] Hairer, E.: *Geometric Integration of Ordinary Differential Equations on Manifolds*. Springer, 2001.
- [13] Kunkel, P. und Mehrmann, V.L.: *Differential–algebraic Equations: Analysis and Numerical Solution*. European Mathematical Society, 2006.
- [14] Hairer, E. und Wanner, G.: *Solving Ordinary Differential Equations II: Stiff and Differential–Algebraic Problems*. Springer, 2002.
- [15] Estévez Schwarz, D.: *Consistent initialization for index–2 differential algebraic equations and its application to circuit simulation*. Dissertation, Humboldt–Universität zu Berlin, 2000.
- [16] Eich, E. und Hanke, M.: *Regularization Methods for Constrained Mechanical Multi-body Systems*. ZAMM - Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik, 1995.
- [17] Kugi, A., Kemmetmüller, W. und Glück, T.: *Fachvertiefung Automatisierungs- und Regelungstechnik*. Skriptum zur Vorlesung, Technische Universität Wien, 2012.
- [18] Nedialkov, N. und Pryce, J.: *DAETS: a Differential–Algebraic Equation Code in C++ for High Index and High Accuracy*. <http://www.mmrc.iss.ac.cn/~dart4/slides/Ned.pdf>, Zugriff: 06.03.2014.
- [19] APMonitor. <http://apmonitor.com>, Zugriff: 06.03.2014.
- [20] Richard, M.J.: *Simulation of Flexible Multibody Systems Using Linear Graph Theory*. InTech, 2012.
- [21] McPhee, J., Schmitke, C. und Morency, K.: *Using graph theory and symbolic computing to generate efficient models for multi-body vehicle dynamics*. Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi–body Dynamics, 2008.
- [22] Ascher, U.M., Chin, H., Petzold, L.R. und Reich, S.: *Stabilization of constrained mechanical systems with DAEs and invariant manifolds*. J. Mech. Struct. Machines, 1994.
- [23] Cellier, F.E.: *The Structural Singularity Removal Algorithm by Pantelides*. http://www.inf.ethz.ch/personal/fcellier/Lect/MMPS/Ppt/mmps_6_engl.ppt, Zugriff: 10.02.2014.

- [24] Cellier, F.E. und Kofman, E.: *Continuous System Simulation*. Springer, 2006.
- [25] Hairer, E.: *Symmetric Projection Methods for Ordinary Differential Equations on Manifolds*. Springer, 2000.
- [26] Frochte, J.: *Evaluation and Adaption of Techniques for Higher Index DAE with Respect to Real-Time Simulation*. SNE Simulation Notes Europe, 2012.
- [27] The MathWorks Inc.: *Simulink Online-Hilfe*. <http://www.mathworks.de/de/help/simulink/ug/choosing-a-solver.html>, Zugriff: 25.06.2014.
- [28] Benker, H.: *Differentialgleichungen mit MATHCAD und MATLAB*. Springer, 2005.
- [29] The MathWorks Inc.: *Matlab Online-Hilfe*. <http://www.mathworks.de/de/help/matlab/math/ordinary-differential-equations.html>, Zugriff: 25.06.2014.