

# Stabilitätsanalyse für nichtlineare Systeme mithilfe der Intervallarithmetik

vom Fachbereich  
Elektrotechnik, Informationstechnik und Medientechnik  
der Bergischen Universität Wuppertal  
genehmigte

Dissertation

zur Erlangung des akademischen Grades eines  
Doktors der Ingenieurwissenschaften

von  
Sascha Alexander Warthenpfehl, M.Sc.

aus  
Wuppertal

Tag der mündlichen Prüfung: 20.04.2012

Referent: Univ.-Prof. Dr.-Ing. Bernd Tibken

Koreferent: Univ.-Prof. Dr.-Ing. habil. Christoph Ament

Diese Dissertation kann wie folgt zitiert werden:

urn:nbn:de:hbz:468-20120516-114847-9

[<http://nbn-resolving.de/urn/resolver.pl?urn=urn:nbn:de:hbz:468-20120516-114847-9>]

# Inhaltsverzeichnis

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Einleitung</b>  | <b>1</b>  |
| 1.1      | Motivation . . . . .                                       | 1         |
| 1.2      | Zielsetzung und Konzept . . . . .                          | 2         |
| <b>2</b> | <b>Problemstellung und Stand der Forschung</b>             | <b>4</b>  |
| 2.1      | Stabilität von dynamischen Systemen . . . . .              | 4         |
| 2.2      | Problemstellung . . . . .                                  | 11        |
| 2.3      | Stand der Forschung für polynomiale Systeme . . . . .      | 14        |
| 2.4      | Stand der Forschung für nichtpolynomiale Systeme . . . . . | 16        |
| <b>3</b> | <b>Intervallarithmetik</b>                                 | <b>19</b> |
| 3.1      | Einführung . . . . .                                       | 19        |
| 3.2      | Notation . . . . .   | 20        |
| 3.3      | Arithmetische Operatoren für Intervalle . . . . .          | 21        |
| 3.4      | Intervallvektoren . . . . .                                | 23        |
| 3.5      | Intervallerweiterung . . . . .                             | 24        |
| 3.6      | Globaler Optimierungsalgorithmus . . . . .                 | 31        |
| 3.7      | Lösen von Intervallgleichungssystemen . . . . .            | 33        |

|                                 |   |               |
|---------------------------------|---|---------------|
| <b>4</b>                        | <b>Berechnung gesicherter Einzugsgebiete für nichtlineare Systeme</b> | <b>37</b>     |
| 4.1                             | Erster algorithmischer Ansatz . . . . .                               | 37            |
| 4.2                             | Erweiterungen der Methode . . . . .                                   | 41            |
| 4.3                             | Anpassungen für Lyapunowfunktionen höherer Ordnung . . . . .          | 44            |
| 4.4                             | Beispiele . . . . .   | 45            |
| <br><b>5</b>                    | <br><b>Parallelisierung</b>   | <br><b>74</b> |
| 5.1                             | Einleitung . . . . .  | 74            |
| 5.2                             | Umsetzung . . . . .   | 75            |
| 5.3                             | Effizienzbetrachtung . . . . .  | 78            |
| <br><b>6</b>                    | <br><b>Zusammenfassung und Ausblick</b>                               | <br><b>82</b> |
| <br><b>A</b>                    | <br><b>Algorithmen</b>  | <br><b>85</b> |
| A.1                             | Stabilitätsanalyse mit quadratischen Ljapunowfunktionen . . . . .     | 85            |
| A.2                             | Stabilitätsanalyse mit Ljapunowfunktionen höherer Ordnung . . . . .   | 86            |
| <br><b>Literaturverzeichnis</b> |   | <br><b>88</b> |

# Abbildungsverzeichnis

|     |  |    |
|-----|--|----|
| 2.1 | Trajektorie eines asymptotisch stabilen Systems gemäß Definition   |    |
|     | 2.1 mit Ruhelage $x^R = 0$ . . . . .   | 5  |
| 2.2 | Pendel nach [61] . . . . .   | 8  |
| 2.3 | Nicht zusammenhängendes, durch $V(x) < 1$ definiertes Gebiet . .   | 10 |
| 2.4 | Gebiete $\Omega_c$ für verschiedene $c$ . . . . .  | 12 |
| 3.1 | Schnitt von zwei Intervallvektoren . . . . .   | 24 |
| 3.2 | Infimum und Supremum von $F^I(x^I)$ auf $x^I = [0, 2]$ verglichen mit dem Funktionsverlauf von $f(x)$ auf $x^I$ (aus Beispiel 3.1) . . . . . | 26 |
| 3.3 | Vergleich natürlicher Intervallerweiterung und Bisektion (aus Beispiel 3.2) . . . . .  | 28 |
| 3.4 | Vergleich natürlicher Intervallerweiterung und Mittelpunktsform (aus Beispiel 3.3) . . . . .   | 29 |
| 3.5 | Vergleich natürlicher Intervallerweiterung und Sekantenform (aus Beispiel 3.4) . . . . .   | 31 |
| 4.1 | Aufbau der Liste $\mathcal{L}$ . . . . .   | 38 |
| 4.2 | Darstellung der Fälle $\bar{V} < \underline{c}$ und $\underline{V} < \bar{c}$ . . . . .  | 39 |
| 4.3 | Darstellung der Fälle $\underline{V} > 0$ und $\bar{V} < 0$ . . . . .  | 40 |
| 4.4 | Aufbau der erweiterten Liste $\mathcal{L}$ . . . . .   | 42 |
| 4.5 | Ermitteltes Einzugsgebiet für Beispiel 4.1 . . . . .   | 47 |

---

|      |   |    |
|------|---|----|
| 4.6  | Ermitteltes Einzugsgebiet für Beispiel 4.2 . . . . .                    | 47 |
| 4.7  | Ermitteltes Einzugsgebiet für Beispiel 4.3 . . . . .                    | 48 |
| 4.8  | Ermitteltes Einzugsgebiet für Beispiel 4.4 . . . . .                    | 48 |
| 4.9  | Ermitteltes Einzugsgebiet für Beispiel 4.5 . . . . .                    | 50 |
| 4.10 | Ermitteltes Einzugsgebiet für Beispiel 4.6 . . . . .                    | 50 |
| 4.11 | Ermitteltes Einzugsgebiet für Beispiel 4.8 . . . . .                    | 53 |
| 4.12 | Ermitteltes Einzugsgebiet für Beispiel 4.9 . . . . .                    | 53 |
| 4.13 | Ermitteltes Einzugsgebiet für Beispiel 4.10 . . . . .                   | 55 |
| 4.14 | Ermitteltes Einzugsgebiet für Beispiel 4.11 . . . . .                   | 55 |
| 4.15 | Ermitteltes Einzugsgebiet für Beispiel 4.12 . . . . .                   | 56 |
| 4.16 | Ermitteltes Einzugsgebiet für Beispiel 4.13 . . . . .                   | 56 |
| 4.17 | Ermitteltes Einzugsgebiet für Beispiel 4.14 . . . . .                   | 59 |
| 4.18 | Ermitteltes Einzugsgebiet für Beispiel 4.15 . . . . .                   | 59 |
| 4.19 | Ermitteltes Einzugsgebiet für Beispiel 4.16 . . . . .                   | 61 |
| 4.20 | Ermitteltes Einzugsgebiet für Beispiel 4.17 . . . . .                   | 61 |
| 4.21 | Ermitteltes Einzugsgebiet für Beispiel 4.18 . . . . .                   | 63 |
| 4.22 | Ermitteltes Einzugsgebiet für Beispiel 4.19 . . . . .                   | 63 |
| 4.23 | Ermitteltes Einzugsgebiet für Beispiel 4.20 . . . . .                   | 64 |
| 4.24 | Ermitteltes Einzugsgebiet für Beispiel 4.21 . . . . .                   | 64 |
| 4.25 | Ermitteltes Einzugsgebiet für Beispiel 4.22 . . . . .                   | 66 |
| 4.26 | Ermitteltes Einzugsgebiet für Beispiel 4.23 . . . . .                   | 66 |
| 4.27 | Ermitteltes Einzugsgebiet für Beispiel 4.24 . . . . .                   | 68 |
| 4.28 | Ermitteltes Einzugsgebiet für Beispiel 4.25 . . . . .                   | 68 |
| 5.1  | Architektur der Parallelisierung auf einer CPU mit $e$ Kernen . . . . . | 75 |
| 5.2  | Verwaltung von parallelen Listen . . . . .                              | 76 |

---

|     |  |    |
|-----|--|----|
| 5.3 | Architektur der Parallelisierung auf einem Mehrkernrechner-Verbund (nach [64]) . . . . . | 77 |
| 5.4 | Zwei-Listen-Konzept . . . . .  | 78 |
| 5.5 | Konfiguration mit 3 Threads pro 4-Kern-Prozessor . . . . .                               | 80 |
| 5.6 | Konfiguration mit 4 Threads pro 4-Kern-Prozessor . . . . .                               | 81 |

# Tabellenverzeichnis

|     |   |    |
|-----|---|----|
| 3.1 | Verkleinerung des Startintervalls mithilfe des Intervall-Newton-Verfahrens im Beispiel 3.5 . . . . .    | 35 |
| 4.1 | Berechnung von Einzugsgebieten für polynomiale Systeme . . . . .  | 70 |
| 4.2 | Berechnung von Einzugsgebieten für polynomiale Systeme (Fortsetzung) . . . . .                          | 71 |
| 4.3 | Berechnung von Einzugsgebieten für polynomiale Systeme mit Ljapunowfunktionen höherer Ordnung . . . . . | 72 |
| 4.4 | Ljapunowfunktionen für die Beispiele aus Tabelle 4.3 . . . . .  | 72 |
| 4.5 | Berechnung von Einzugsgebieten für nichtpolynomiale Systeme . . . . .                                   | 73 |
| 5.1 | Laufzeiten für verschiedene Konfigurationen . . . . .   | 79 |



# Symbolverzeichnis

|                              |  |
|------------------------------|--|
| $\mathbb{R}_+$               | Menge der echt positiven reellen Zahlen  |
| $V(x)$                       | Ljapunowfunktion in den Variablen $x = [x_1, \dots, x_n]^T$                      |
| $\dot{V}(x)$                 | zeitliche Ableitung der Ljapunowfunktion $V(x)$                                  |
| $f(x)$                       | rechte Seite der Zustandsdifferentialgleichung                                   |
| $\Omega$                     | Einzugsgebiet einer Ruhelage des Systems   |
| $\Omega_c$                   | Teilgebiet des exakten Einzugsgebiets,<br>gebildet durch Niveaumengen von $V(x)$ |
| $\partial\Omega_c$           | Rand der Menge $\Omega_c$  |
| $x^0$                        | Anfangszustand des Systems   |
| $x^R$                        | Ruhelage des Systems   |
| $n$                          | meist: Anzahl der Systemzustände   |
| $x^I$                        | Intervall(vektor)  |
| $\check{x}, \text{mid}(x^I)$ | Mittelpunkt eines Intervalls bzw. Intervallvektors                               |
| $F^I(x^I)$                   | (beliebige) Intervallerweiterung der Funktion $f(x)$ auf $x^I$                   |
| $F^N(x^I)$                   | natürliche Intervallerweiterung der Funktion $f(x)$ auf $x^I$                    |
| $F^M(x^I)$                   | Mittelpunktsform der Funktion $f(x)$ auf $x^I$                                   |
| $F^S(x^I)$                   | Sekantenform der Funktion $f(x)$ auf $x^I$                                       |
| $N(x^I)$                     | Newtonoperator auf $x^I$   |
| $K(x^I)$                     | Krawczyk-Operator auf $x^I$  |
| $\mathcal{L}$                | Liste des Optimierungsalgorithmus  |
| $\ x\ $                      | $L_2$ -Norm des Vektors $x$ , $\ x\  = \sqrt{x^T x}$                             |
| sos                          | Summe von Quadraten von Polynomen  |
| LMI                          | Lineare Matrixungleichung(en)  |

## Einleitung

### 1.1 Motivation

Die Stabilität gehört zu den wichtigsten Eigenschaften, die ein dynamisches System aufweisen kann. Erste Betrachtungen der Stabilität wurden deshalb schon von Aristoteles und Archimedes [63, 83] durchgeführt. Ende des 19. beziehungsweise Anfang des 20. Jahrhunderts wurden unter anderem von Hurwitz, Routh und Nyquist [43, 62, 72, 75, 76] vollständige Methoden für lineare Systeme vorgestellt, welche heute noch als Stand der Technik gelten.

Allerdings sind diese Methoden für die meisten dynamischen Systeme nicht direkt anwendbar, da sich diese nur mit nichtlinearen Differentialgleichungen beschreiben lassen. Hier sind zum Beispiel die polynomialen Lotka-Volterra-Gleichungen, welche die Populationsdynamik beschreiben, oder mechanische Systeme wie elektrische Maschinen zu nennen, deren Modelldifferenzialgleichungen aufgrund der Winkelbeziehungen trigonometrische Terme beinhalten.

Für solche Systeme ist die Stabilitätsbetrachtung ein aktuelles Forschungsgebiet. Die meisten Ansätze basieren hierbei auf der Stabilitätstheorie von Ljapunow [59], welche weitgehende Kriterien zur Analyse der Stabilität von nichtlinearen, autonomen Systemen bietet.

Ein wichtiges Teilgebiet der Stabilitätsanalyse für solche Systeme ist die Bestimmung der Größe des Einzugsgebiets. Das Einzugsgebiet beschreibt die Menge der Anfangszustände, für welche Lösung der Systemdifferenzialgleichung in die Ruhelage  $x^R$  mit  $f(x^R) = 0$  einlaufen. Diese Betrachtung führt auf eine nichtlineare

Optimierungsaufgabe mit Nebenbedingungen, bei der numerische Lösungsverfahren im Allgemeinen scheitern. In den letzten Jahren wurden verschiedene Ansätze präsentiert, unter anderem sind hier [10–16, 19, 21, 24, 29, 35–38, 42, 45, 56, 66, 68, 73, 80, 88, 90, 92–95, 95, 96, 99, 102] zu nennen. Allerdings sind mit Ausnahme von [11–13, 80] die meisten Ansätze nur für polynomiale Systeme anwendbar. Dies wird damit begründet, dass sich nach dem Approximationssatz von Weierstraß [101] jede nichtlineare Funktion durch Polynome beliebig genau darstellen lässt. Jedoch sind hierdurch die Ergebnisse immer mit einer gewissen Ungenauigkeit behaftet, was insbesondere für sicherheitsrelevante Betrachtungen nicht akzeptabel ist.

Dies motivierte die Entwicklung eines neuen Algorithmus zur Bestimmung einer garantierten inneren Approximation des Einzugsgebiets eines nichtlinearen autonomen Systems, welcher in Teilen in [65, 100] publiziert und ausführlicher in dieser Arbeit präsentiert werden soll. Dieser Algorithmus nutzt die aus der Verifikationsnumerik stammende Intervallarithmetik, um gesicherte Einzugsgebiete für ein nichtlineares System zu bestimmen.

## 1.2 Zielsetzung und Konzept

Ziel dieser Arbeit ist es, für Systeme mit asymptotischer stabiler Ruhelage im Ursprung mithilfe der Stabilitätstheorie von Ljapunow und der Intervallarithmetik innere Schranken für das Einzugsgebiet der Ruhelage zu bestimmen.

Die Begriffe Stabilität, Ruhelage und Einzugsgebiete werden in Kapitel 2 eingehend anhand der Stabilitätstheorie von Ljapunow [53, 59] behandelt. Es werden dabei die Probleme bei der Bestimmung von Einzugsgebieten aufgezeigt und die bereits existierenden Lösungs- bzw. Relaxationsverfahren präsentiert.

Das in dieser Arbeit vorgestellte Lösungsverfahren nutzt die sogenannte Intervallarithmetik [3, 4]. Hierzu werden in Kapitel 3 Notation, Rechenvorschriften und Erweiterungen der Intervallarithmetik vorgestellt. Die Intervallarithmetik sichert eine garantierte Einschließung des Wertebereichs einer Funktion auf einem Intervall. Allerdings ist hierbei diese Einschließung meist nicht exakt. Aus diesem Grund werden in der vorliegenden Arbeit Möglichkeiten aufgezeigt, die sogenannte Überschätzung zu reduzieren.

Zur Lösung der in Kapitel 2 dargelegten Aufgabenstellung wird in Kapitel 4 ein Branch-and-Bound-Algorithmus vorgestellt, welcher auf die meisten in der Praxis auftretenden nichtlinearen Systeme anwendbar ist. Er wird an Benchmark-Beispielen aus der Literatur evaluiert. Dies geschieht mit dem Ziel, die Ergebnisse mit den Resultaten der konkurrierenden Verfahren zu vergleichen.

Um der aktuellen Entwicklung hinsichtlich der Parallelisierbarkeit von Algorithmen Rechnung zu tragen, werden in Kapitel 5 Aufwand und Vorteile der Parallelisierung anhand des in dieser Arbeit vorgestellten Algorithmus aufgezeigt.

Abschließend beinhaltet Kapitel 6 eine Zusammenfassung und einen Ausblick auf mögliche Weiterentwicklungen.

# Problemstellung und Stand der Forschung

## 2.1 Stabilität von dynamischen Systemen

In dieser Arbeit werden autonome, dynamische Systeme betrachtet, deren Zustandsdifferentialgleichungen in der Form

$$\dot{x}(t) = f(x), \quad x \in \mathbb{R}^n, \quad t \in \mathbb{R}, \quad x(0) = x^0 \quad (2.1)$$

gegeben sind. Hierbei seien  $x(t) \in \mathbb{R}^n$  der zeitliche Verlauf des Zustandsvektors,  $x^0 \in \mathbb{R}^n$  der Anfangszustand des Systems zum Zeitpunkt  $t = 0$  und  $n$  die Anzahl der Systemzustände. Die (Vektor-) Funktion  $f(x)$  sei stetig und differenzierbar. Die Bezeichnung „autonomes System“ bedeutet hierbei, dass das System keine Eingänge besitzt. Dies ist beispielsweise dann der Fall, wenn der Regler schon entworfen und in der Systemdifferentialgleichung berücksichtigt wurde.

Die Stabilitätseigenschaft beschreibt die Reaktion der Systemdynamik auf eine Auslenkung oder Erregung  $x^0$ . Hierbei ist das Ziel, dass die Zustände nach einer Auslenkung für  $t \rightarrow \infty$  in die Ruhelage  $x^R$  mit  $f(x^R) = 0$  einlaufen.

### **Definition 2.1: Stabilität (nach [25, 62])**

Die Ruhelage  $x^R$  mit  $f(x^R) = 0$  des Systems (2.1) heißt *grenzstabil im Sinne von Ljapunow*, wenn für jedes  $\varepsilon > 0$  eine Zahl  $\delta > 0$  existiert, sodass bei einem beliebigen Anfangszustand, der die Bedingung

$$\|x^0 - x^R\| < \delta$$

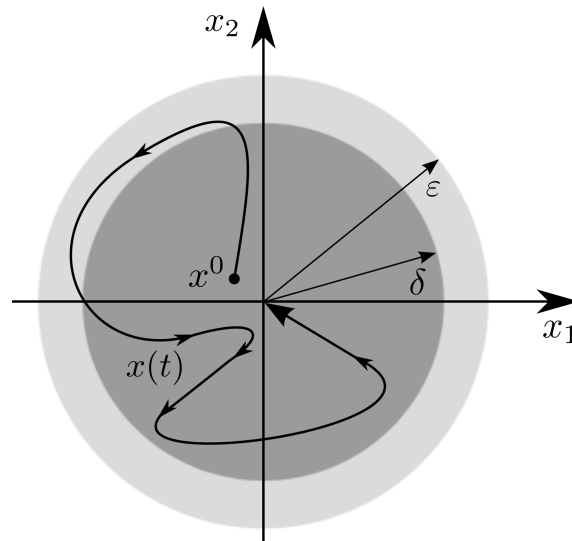


Abbildung 2.1: Trajektorie eines asymptotisch stabilen Systems gemäß Definition 2.1 mit Ruhelage  $x^R = 0$

erfüllt, die Eigenbewegung des Systems die Bedingung

$$\|x(t) - x^R\| < \varepsilon \quad \text{für alle } t > 0$$

erfüllt. Die Ruhelage heißt *asymptotisch stabil*, wenn sie stabil ist und

$$\lim_{t \rightarrow \infty} \|x(t)\| = x^R$$

gilt.

In Abbildung 2.1 ist beispielhaft der Verlauf einer Trajektorie eines asymptotisch stabilen Systems mit der Ruhelage  $x^R = 0$  aufgezeichnet. Da der Anfangszustand die Bedingung  $\|x^0\| < \delta$  erfüllt, verlässt die Trajektorie gemäß Definition 2.1 das Gebiet  $\|x(t)\| < \varepsilon$  nicht und läuft in die Ruhelage ein.

Lässt sich  $f(x)$  in der Form

$$f(x) = Ax$$

mit  $A \in \mathbb{R}^{n \times n}$  darstellen, handelt es sich um ein lineares System. Für die Klasse der linearen Systeme lässt sich die Stabilität einer Ruhelage anhand mehrerer Verfahren eingehend überprüfen [62]. Eine Methode stellt die Betrachtung der Eigenwerte der Systemmatrix  $A$  dar. Liegen alle Eigenwerte von  $A$  in der linken komplexen Halbebene, so ist die Ruhelage asymptotisch stabil. Befindet sich mindestens ein Eigenwert in der rechten Halbebene oder liegen mindestens zwei

Eigenwerte am gleichen Punkt auf der imaginären Achse, so ist die Ruhelage nicht stabil. Andernfalls ist das System grenzstabil.

Da ein asymptotisch stabiles System eine Systemmatrix mit negativen Eigenwerten hat, ist der Rang der Matrix  $A$  gleich der Anzahl der Systemzustände  $n$ , also maximal. Dies bedeutet, dass nur  $x = 0$  die Bedingung  $f(x) = 0$  erfüllt und damit das System nur eine Ruhelage im Ursprung hat. Ist die Systemmatrix hingegen singulär, besitzt das System ein Kontinuum von unendlich vielen Ruhelagen [1].

Bei nichtlinearen Systemen ist aufgrund der höheren Komplexität diese einfache Betrachtung nicht mehr möglich. Da  $f(x)$  nicht durch eine Matrix charakterisiert wird, existieren auch keine Eigenwerte, welche für eine Stabilitätsanalyse verwendet werden können. Weiterhin kann das System nun mehr als eine Ruhelage haben, ohne dass diese ein Kontinuum bilden. Hierbei ist die Stabilität für jede Ruhelage einzeln zu untersuchen, da ein System gleichzeitig stabile und instabile Ruhelagen besitzen kann.

Der erste Ansatz zur Stabilitätsbetrachtung einer Ruhelage eines nichtlinearen Systems ist die Linearisierung der Systemdifferentialgleichung in dieser Ruhelage. Wenn die Methoden für lineare Systeme das linearisierte System als asymptotisch stabil kennzeichnen, dann ist auch die Ruhelage des nichtlinearen Systems asymptotisch stabil. Folglich weiß man, dass eine Umgebung  $\delta$  um die Ruhelage existiert, für welche alle Trajektorien in den Ursprung laufen. Allerdings hat man keine Aussage über die Größe dieser Umgebung.

Im Allgemeinen bezeichnet man die Menge aller Anfangszustände, für welche die Trajektorien des Systems in die Ruhelage  $x^R$  einlaufen, als Einzugsgebiet. Diese Menge wird durch

$$\Omega := \left\{ x^0 \in \mathbb{R}^n \mid \lim_{t \rightarrow \infty} x(t) = x^R \right\}$$

beschrieben.

Das vollständige Einzugsgebiet ist im Allgemeinen nicht algebraisch beschreibbar [50]. Allerdings eröffnet sich mit Hilfe der Stabilitätstheorie von Ljapunow [59] die Möglichkeit, eine innere Approximation des Einzugsgebiets algebraisch anzugeben.

Um nun  $\Omega$  von einer inneren Approximation abzugrenzen, wird  $\Omega$  im Folgenden als exaktes Einzugsgebiet bezeichnet. Weiterhin wird ohne Beschränkung der Allgemeinheit angenommen, dass die zu betrachtende Ruhelage im Ursprung

liegt, also

$$x^R = 0$$

gilt. Wenn dies nicht der Fall ist, kann dies über die lineare Zustandstransformation

$$z = x - x^R$$

erfolgen. Für die weitere Betrachtung müssen die Begriffe der positiven bzw. negativen Definitheit sowie der Ljapunowfunktion definiert werden.

**Definition 2.2: Definitheit von Funktionen**

Eine skalare Funktion  $V(x)$  heißt positiv (semi-)definit auf einer Menge  $M$ , falls folgende Bedingungen gelten:

- i)  $V(0) = 0$
- ii)  $V(x) > 0$  ( $V(x) \geq 0$ )  $\forall x \in M$  und  $x \neq 0$

Eine skalare Funktion  $V(x)$  heißt negativ (semi-)definit auf einer Menge  $M$ , falls folgende Bedingungen gelten:

- i)  $V(0) = 0$
- ii)  $V(x) < 0$  ( $V(x) \leq 0$ )  $\forall x \in M$  und  $x \neq 0$

**Definition 2.3: Ljapunowfunktion**

Eine skalare, stetig differenzierbare Funktion  $V(x)$  heißt Ljapunowfunktion auf einer Menge  $M \ni 0$ , falls  $V(x)$  die folgenden Bedingungen erfüllt:

- i)  $V(x)$  ist stetig differenzierbar und positiv definit auf  $M$ .
- ii) Die zeitliche Ableitung  $\dot{V}(x)$  entlang der Trajektorien von (2.1)

$$\dot{V}(x) = \frac{\partial V}{\partial x} f(x)$$

ist negativ semidefinit auf  $M$ .

Zur Veranschaulichung, inwieweit man die Ljapunowfunktion nutzen kann, um die Stabilität einer Ruhelage zu untersuchen, wird auf ein Beispiel aus [61] zurückgegriffen:



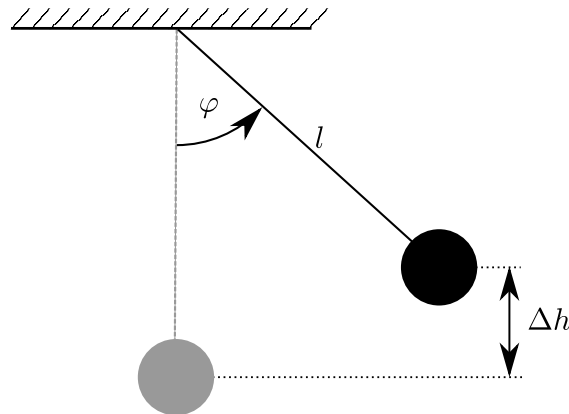


Abbildung 2.2: Pendel nach [61]

**Beispiel 2.1**

Das in Abbildung 2.2 dargestellte Pendel kann mit dem Winkel  $x_1 = \varphi$  ausgelenkt werden, die Winkelgeschwindigkeit sei durch den Zustand  $x_2 = \dot{\varphi}$  gegeben.

Die Ruhelage  $x^R = (0, 0)^T$  ist asymptotisch stabil, denn wenn man das Pendel auf  $x^0$  auslenkt, wird es mit kleiner werdender Amplitude hin- und herschwingen und letztendlich in die Ruhelage mit dem Winkel  $\varphi = 0$  und der Winkelgeschwindigkeit  $\dot{\varphi} = 0$  einlaufen. Dieses Verhalten wird durch den Luftwiderstand verursacht, welcher die Energie des Pendels kontinuierlich verkleinert. Die Energie des Pendels ist die Summe der kinetischen Energie  $V_{\text{kin}}(x(t))$ , welche von der Winkeländerung  $x_2(t)$  abhängt, und der potenziellen Energie  $V_{\text{pot}}(x(t))$ , welche durch die Höhe  $\Delta h(t)$  bestimmt wird, welche aus  $x_1(t)$  berechnet werden kann.

Für die Gesamtenergie

$$V(x(t)) = V_{\text{kin}}(x(t)) + V_{\text{pot}}(x(t))$$

gilt aufgrund der Dämpfung also die Eigenschaft

$$\lim_{t \rightarrow \infty} V(x(t)) = 0 \quad .$$

Aus  $V = 0$  folgt demnach, dass die Höhe  $\Delta h$  und die Geschwindigkeit  $\dot{\varphi}$  verschwinden, was bedeutet, dass das Pendel aus jeder Auslenkung  $x^0$  in die Ruhelage zurückkehrt.

Die Ljapunowfunktion beschreibt demnach eine Art Energiefunktion des Systems, wobei diese nicht der tatsächlichen Energiefunktion des Systems entsprechen, sondern nur die Eigenschaften einer solchen besitzen muss. Die erste Eigenschaft

ist hierbei, dass die Energie in der Ruhelage verschwindet und für alle anderen Zustände positiv ist, d. h. dass  $V(x(t))$  positiv definit ist. Weiterhin soll sich der Wert der Ljapunowfunktion entlang der Trajektorie ständig verkleinern oder zumindest nicht ansteigen, es soll folglich

$$\dot{V}(x(t)) = \frac{\partial V}{\partial x} f(x(t)) \leq 0$$

gelten. Mit diesen Vorbetrachtungen kann nun ein wichtiger Satz zur asymptotischen Stabilität formuliert werden (die Zeitabhängigkeit der Trajektorie  $x(t)$  wird im Folgenden aus Gründen der besseren Lesbarkeit wieder ausgelassen).

**Satz 2.1: Lokale asymptotische Stabilität [35, 53]**

*Wenn auf einer Umgebung  $D$  des Ursprungs eine Ljapunowfunktion  $V(x)$  existiert, deren zeitliche Ableitung  $\dot{V}(x)$  negativ definit auf  $D$  ist, dann ist die Ruhelage im Ursprung asymptotisch stabil.*

Nun kann eine Untermenge des Einzugsgebiets mit dem folgenden Satz angegeben werden.

**Satz 2.2: Garantierte Untermenge des Einzugsgebiets [35, 50]**

*$V(x)$  sei eine Ljapunowfunktion für das autonome System (2.1) auf dem Gebiet*

$$\Omega_c^* = \{x \in \mathbb{R}^n \mid V(x) \leq c\}, \quad c > 0 \quad . \quad (2.2)$$

*Wenn*

- $\Omega_c^*$  zusammenhängend und beschränkt ist sowie den Ursprung enthält und
- $\dot{V}(x)$  negativ definit auf  $\Omega_c^*$  ist,

*dann ist die Ruhelage im Ursprung asymptotisch stabil und  $\Omega_c^*$  ist eine garantierte Untermenge des Einzugsgebiets der Ruhelage, das heißt, es gilt*

$$\Omega_c^* \subseteq \Omega \quad .$$

Nach Satz 2.2 muss das Gebiet

$$\Omega_c^* = \{x \in \mathbb{R}^n \mid V(x) < c\}$$

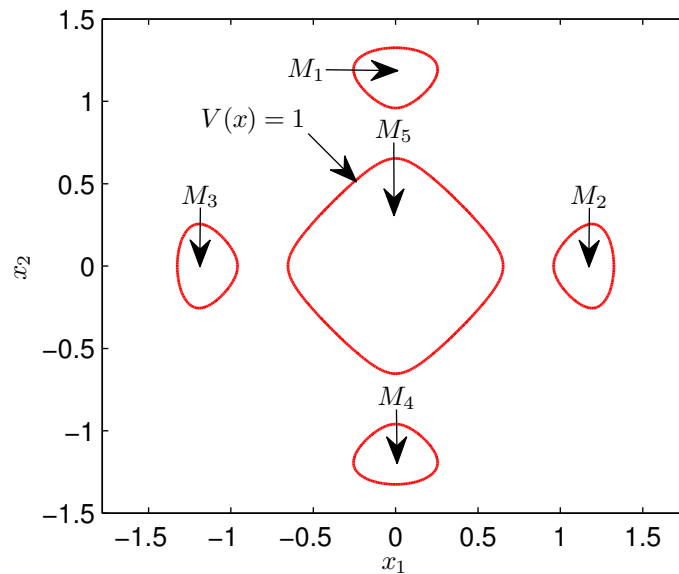


Abbildung 2.3: Nicht zusammenhängendes, durch  $V(x) < 1$  definiertes Gebiet

zusammenhängend sein und den Ursprung enthalten. Dies ist jedoch nicht für alle Ljapunowfunktionen erfüllt. In Abbildung 2.3 ist das durch die Funktion

$$V(x) = 4 \left( x_1^2 + x_2^2 \right) - 4.5 \left( x_1^4 + x_2^4 \right) + 1.45 \left( x_1^6 + x_2^6 \right)$$

und durch die Bedingung  $V(x) < 1$  eingeschlossene Gebiet abgebildet.  $\Omega_c^*$  besteht hier aus fünf nicht zusammenhängenden Teilgebieten  $M_1$ ,  $M_2$ ,  $M_3$ ,  $M_4$  und  $M_5$ , es gilt also

$$\Omega_c^* = M_1 \cup M_2 \cup M_3 \cup M_4 \cup M_5 \quad .$$

Hierbei enthält das Gebiet  $M_5$  den Ursprung. Es ist offensichtlich, dass die Gebiete  $M_1$ ,  $M_2$ ,  $M_3$  und  $M_4$  keine Teilgebiete des garantierten Einzugsgebiets sind. Wäre  $V(x)$  auf den Gebieten  $M_1$ ,  $M_2$ ,  $M_3$ ,  $M_4$  und  $M_5$  eine Ljapunowfunktion, würde dies bedeuten, dass  $\dot{V}(x)$  auf diesen Gebieten negativ semidefinit sein muss. Für die in  $M_5$  startenden Trajektorien folgt daraus, dass die Trajektorien in die Ruhelage im Ursprung einlaufen. Die in den anderen Teilgebieten startenden Trajektorien können nicht gegen die Ruhelage im Ursprung konvergieren, da die Ränder der einzelnen Mengen aufgrund der negativen Semidefinitheit von  $\dot{V}(x)$  und der Bedingung  $V(x) < 1$  (am Rand selbst gilt  $V(x) = 1$ ) eine unüberwindbare Barriere darstellen.

Aus dieser Betrachtung kann nun der folgende Satz abgeleitet werden.

**Satz 2.3: Erweiterung des Satzes 2.2**

Besteht die Menge

$$\Omega_c^* = \{x \in \mathbb{R}^n \mid V(x) < c\}, \quad c > 0$$

aus mehreren nicht zusammenhängenden Gebieten  $M_0, M_1, \dots, M_r$ , dann bezeichnet  $\Omega_c$  diejenige kompakte und zusammenhängende Teilmenge  $M_i$ , welche die Ruhelage im Ursprung enthält. Besteht  $\Omega_c^*$  aus nur einer zusammenhängenden Menge, dann gilt  $\Omega_c = \Omega_c^*$ .

Ist  $V(x)$  eine Ljapunowfunktion für das autonome System (2.1) auf  $\Omega_c$  und ist  $\dot{V}(x)$  negativ definit auf  $\Omega_c$ , dann ist die Ruhelage im Ursprung asymptotisch stabil und  $\Omega_c$  ist eine garantierte Untermenge des Einzugsgebiets der Ruhelage, das heißt, es gilt

$$\Omega_c \subseteq \Omega \quad .$$

## 2.2 Problemstellung

Zur Bestimmung einer Untermenge des Einzugsgebiets einer Ruhelage gemäß Satz 2.3 müssen zwei Problemstellungen gelöst werden:

- Bestimmung einer geeigneten Ljapunowfunktion
- Bestimmung des maximalen  $c$  für die Untermenge des Einzugsgebiets gemäß Satz 2.3

Bei der Bestimmung der Ljapunowfunktion muss bedacht werden, dass man diese nicht immer aus der Energiefunktion des Systems bilden kann. Weiterhin können unterschiedliche Ljapunowfunktionen zu unterschiedlichen Untermengen des Einzugsgebiets führen. Mit der Methode von Zubov [102] lässt sich grundsätzlich eine Ljapunowfunktion  $V(x)$  bestimmen, deren Niveaumenge  $V(x) = 1$  das exakte Einzugsgebiet einschließt. Allerdings ist hierfür eine partielle Differenzialgleichung zu lösen, weswegen sich diese Methode eher als Existenznachweis einer solchen Ljapunowfunktion nutzen lässt. Viele Verfahren, welche die Methode von Zubov nutzen, approximieren die Lösung der partiellen Differenzialgleichung, wobei hierdurch keine Garantie mehr gegeben ist, dass die erhaltene Menge wirklich nur Punkte enthält, die im exakten Einzugsgebiet der Ruhelage liegen.

Zur Konstruktion von Ljapunowfunktionen wurden in den vergangenen Jahren verschiedene Ansätze vorgestellt (z. B. [11, 13, 14, 16, 19, 37, 45, 46, 56, 68, 88, 99]), weswegen in dieser Arbeit der Fokus auf die Ermittlung einer möglichst großen Untermenge des Einzugsgebiets für eine gegebene Ljapunowfunktion gelegt wird.

In Abbildung 2.4 sind die Gebiete  $\Omega_{c_1}$ ,  $\Omega_{c_2}$  und  $\Omega_{c_3}$  aufgezeigt, welche durch die Niveaumengen  $V(x) = c_1$  respektive  $V(x) = c_2$  und  $V(x) = c_3$  begrenzt werden. Während  $\Omega_{c_1}$  und  $\Omega_{c_2}$  vollständig in der Menge liegen, in der  $\dot{V}(x)$  negativ definit ist, enthält das Gebiet  $\Omega_{c_3}$  Punkte aus der Menge, in welcher  $\dot{V}(x)$  positiv ist. Der Zeichnung kann also entnommen werden, dass  $\Omega_{c_1}$  und  $\Omega_{c_2}$  Untermengen des exakten Einzugsgebiets sind, während eine solche Aussage über  $\Omega_{c_3}$  mit der gegebenen Ljapunowfunktion nicht getroffen werden kann.

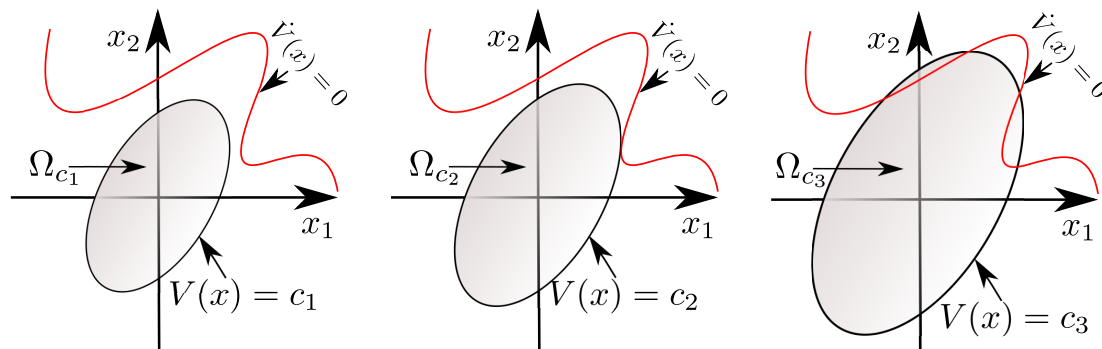


Abbildung 2.4: Gebiete  $\Omega_c$  für verschiedene  $c$

Im Allgemeinen ist eine solche grafische Auswertung nicht anwendbar. Zum Einen ist diese auf Systeme mit bis zu drei Zuständen beschränkt, zum Anderen wird vorausgesetzt, dass die Konturlinien von  $V(x) = c$  und  $\dot{V}(x) = 0$  exakt gezeichnet werden können. Dies kann aber aufgrund der möglichen Komplexität der Konturlinien nicht garantiert werden. Aus diesen Gründen wird versucht, die Aufgabenstellung algebraisch oder numerisch zu lösen.

Nach Forsman [26] kann das maximale  $c$ , welches das Einzugsgebiet  $\Omega_c$  nach Satz 2.3 bestimmt, durch die Optimierungsaufgabe

$$c^* = \min_{\partial\Omega_c \cap \Gamma \neq \emptyset} c \quad (2.3)$$

ermittelt werden, wobei  $\partial\Omega_c$  den Rand von  $\Omega_c$  bezeichnet und  $\Gamma$  durch

$$\Gamma := \{x \in \mathbb{R}^n \mid \dot{V}(x) = 0, x \neq 0\}$$

definiert wird. Gilt  $\Omega_c = \Omega_c^*$ , dann kann Gleichung (2.3) zu

$$c^* = \min_{\{x \in \mathbb{R}^n \mid V(x)=c\} \cap \Gamma \neq \emptyset} c \quad (2.4)$$

umgeschrieben werden. Weiterhin lässt sich dann die Optimierungsaufgabe (2.3) wie folgt vereinfachen. Wenn  $\hat{x}$  einen Punkt auf

$$\Gamma \cap \{x \in \mathbb{R}^n \mid V(x) = c\}$$

bezeichnet, so gelten für diesen Punkt die Gleichungen

$$\begin{aligned} V(\hat{x}) &= c \\ \dot{V}(\hat{x}) &= 0 \\ \hat{x} &\neq 0 \end{aligned} \quad (2.5)$$

Hieraus folgt, dass man Gleichung (2.4) zu

$$c^* = \min_{\substack{V(x) \\ \dot{V}(x) = 0 \\ x \neq 0}} V(x) \quad (2.6)$$

umschreiben kann. Auch wenn die Optimierungsaufgabe (2.6) eine kompakte Darstellung hat, ist sie nicht einfach zu lösen. Dies liegt zum Einen daran, dass die Zielfunktion und die Gleichungsnebenbedingung nichtlinear sind, zum Anderen ist der Ausschluss der Lösung  $x = 0$  ein Problem für globale Löser.

Um diesen Sachverhalt zu veranschaulichen, soll versucht werden, die Optimierungsaufgabe (2.6) mit der Lagrangemultiplikatorenregel [7] zu lösen. Für Optimierungsaufgaben der Form

$$\min_{h(x)=0} g(x) \quad (2.7)$$

ist die sogenannte Lagrangefunktion durch

$$L(x, \lambda) := g(x) + \lambda h(x)$$

definiert. Ist nun  $x^*$  ein lokales Minimum, welches die Aufgabe (2.7) erfüllt, dann existiert ein  $\lambda$ , sodass das Gleichungssystem

$$\begin{aligned} \frac{\partial L}{\partial x_1}(x^*, \lambda) &= 0 \\ &\vdots \\ \frac{\partial L}{\partial x_n}(x^*, \lambda) &= 0 \\ \frac{\partial L}{\partial \lambda}(x^*) &= 0 \end{aligned}$$

erfüllt ist. Wendet man diese Methode nun auf die Optimierungsaufgabe (2.6) an und vernachlässigt die Nebenbedingung  $x \neq 0$ , so ergibt sich mit der Lagrange-funktion

$$L(x, \lambda) = V(x) + \lambda \dot{V}(x)$$

das Gleichungssystem

$$\begin{aligned} \frac{\partial V}{\partial x_1}(x^*) + \lambda \frac{\partial \dot{V}}{\partial x_1}(x^*) &= 0 \\ &\vdots \\ \frac{\partial V}{\partial x_n}(x^*) + \lambda \frac{\partial \dot{V}}{\partial x_n}(x^*) &= 0 \\ \dot{V}(x^*) &= 0 \end{aligned} \quad (2.8)$$

Da  $V(x)$  positiv und  $\dot{V}(x)$  negativ definit auf einer Umgebung um den Ursprung sein müssen, besitzen diese Funktionen im Ursprung ein Extremum, das heißt, die Gleichungen

$$\begin{aligned} \frac{\partial V}{\partial x_1}(0) = 0 & \quad \frac{\partial \dot{V}}{\partial x_1}(0) = 0 \\ & \quad \vdots \\ \frac{\partial V}{\partial x_n}(0) = 0 & \quad \frac{\partial \dot{V}}{\partial x_n}(0) = 0 \end{aligned}$$

sind erfüllt. Da gleichzeitig noch  $\dot{V}(0) = 0$  gilt, erfüllt der Ursprung für ein beliebiges  $\lambda$  das Gleichungssystem (2.8). Das bedeutet, dass die Lösungsmenge des Gleichungssystems 2.8 neben diskreten Punkten, von denen mindestens einer ein Minimalpunkt des Optimierungsproblems 2.6 ist, auch eine eindimensionale Mannigfaltigkeit  $\{x = 0, \lambda \text{ beliebig}\}$  beinhaltet. Viele numerische Löser scheitern hierbei genau an dieser Eigenschaft. Die Lösung mit Computer-Algebra-Programmen ist nur bei einfachen Aufgabenstellungen möglich, wie beispielsweise der Untersuchung von polynomialen Systemen in wenigen Variablen mit quadratischer Ljapunowfunktion.

## 2.3 Stand der Forschung für polynomiale Systeme

Wie im vorherigen Abschnitt aufgezeigt, ist die Lösung der Optimierungsaufgabe 2.6 ein nichttriviales Problem. Für Systeme, deren rechte Seite der Zustandsdifferentialgleichung polynomial ist, wurden in den vergangenen Jahren verschiedene Verfahren vorgestellt.

Viele dieser Verfahren führen auf die Überprüfung eines Polynoms darauf, ob es als Summe von Quadraten, welche aufgrund der englischen Bezeichnung sum-of-squares [97] auch als sos bezeichnet wird, darstellbar ist. Dies führt wiederum auf sogenannte lineare Matrixungleichungen, welche wegen der englischen Bezeichnung linear matrix inequalities auch LMI genannt werden. Für LMI-Probleme stehen Lösungsverfahren, wie sie in SeDuMi [86] oder LMILAB [27,71] der Robust Control Toolbox in MATLAB [91] implementiert sind, zur Verfügung. Diese Lösungsverfahren sind sehr leistungsfähig und relativ schnell, weswegen viele Verfahren zur Bestimmung eines Einzugsgebiets die Problemstellung in eine LMI-Aufgabe umformulieren. Allerdings arbeiten die zur Verfügung stehenden LMI-Löser nicht in jeder Situation numerisch stabil.

Dennoch sollen einige der Verfahren, welche die Aufgabenstellung in die Lösung einer linearen Matrixungleichung überführen, im Folgenden vorgestellt werden. Hierbei wird stets angenommen, dass die Menge

$$\Omega_c = \{x \in \mathbb{R}^n | V(x) \leq c\}, \quad c > 0$$

zusammenhängend ist und den Ursprung enthält.

Tibken stellt in [92] ein Verfahren vor, welches einen Satz aus der algebraischen Geometrie von Jacobi und Prestel [44] nutzt. Hierzu definiert man  $\Omega_c$  als semi-algebraische Menge [8]  $S(c - V(x))$ . Wenn nun  $\dot{V}(x)$  auf  $S(c - V(x))$  negativ ist, dann existiert nach [44] die Darstellung

$$-\dot{V}(x) = q_0(x) + q_1(x)(c - V(x)) \quad , \quad (2.9)$$

wobei  $q_0(x)$  und  $q_1(x)$  sos-Polynome mit jeweils unbekanntem geradem Grad sind. Um also ein  $c$  zu ermitteln, mit welchem ein Teilgebiet des Einzugsgebiets berechnet werden kann, müssen entsprechende  $q_0(x)$  und  $q_1(x)$  bestimmt werden. Hierzu muss für  $q_1(x)$  ein Ansatz aufgestellt werden und danach Gleichung (2.9) nach  $q_0(x)$  umgestellt werden. Die Überprüfung, ob  $q_0(x)$  und  $q_1(x)$  Summe von Quadraten sind, kann dann über lineare Matrixungleichungen erfolgen. Problematisch ist nach [44] der Wert von  $\dot{V}(x)$  an der Stelle 0, welcher nicht größer null ist. Allerdings ist hier ein Ausnahmefall von Scheiderer [81] erfüllt, sodass das Verfahren von Tibken wie beschrieben angewandt werden kann.

Chesi zeigt in [12, 13], dass, wenn ein Polynom  $s(x)$  existiert, sodass  $s(x)$  und  $-\dot{V}(x) + (V(x) - c)s(x)$  als Summe von Quadraten von Polynomen darstellbar sind und nur im Ursprung verschwinden,  $c \leq c^*$  gilt. Dies ist eine Umstellung der



Methode von Tibken, der Beweis erfolgt hier aber nicht über die Sätze von Jacobi und Prestel sowie Scheiderer.

In [35, 36] wendet Hachicho die Ergebnisse von Laserre [54] zur Theorie der Momente an, um einen Algorithmus zur Bestimmung eines Einzugsgebiets zu entwickeln.

Weitere Methoden, welche auf die Lösung einer LMI-Aufgabe führen, sind in [10, 14, 73, 90] zu finden.

Dilaver und Tibken erweitern in [21, 93] die Methode von Tibken und Hachicho [95], indem Sie einen Satz aus der Analysis von Ehlich und Zeller [23] und dessen Erweiterungen von Ruttmann [79] und Gärtel [28] nutzen, um Schranken für das polynomiale  $\dot{V}(x)$  zu ermitteln und daraus ein möglichst großes Einzugsgebiet zu bestimmen. Als weitere Methoden, welche die Optimierungsaufgabe nicht mithilfe von linearen Matrixungleichungen lösen, sind beispielsweise [24, 29] zu nennen.

## 2.4 Stand der Forschung für nichtpolynomiale Systeme

In [11, 12] präsentiert Chesi ein Verfahren zur Ermittlung von Einzugsgebieten von nichtlinearen Systemen der Form

$$\dot{x} = g(x) + \sum_{i=1}^r h_i(x) \xi_i(x_{a_i}) \quad , \quad (2.10)$$

wobei  $g(x)$  und  $h_i(x)$  polynomiale Funktionen,  $a_i$  Indizes mit der Eigenschaft  $a_i \in \{1, \dots, n\}$  und  $\xi_i(x_{a_i})$  nichtpolynomiale Funktionen in der Variable  $x_{a_i}$  sind. Unter der Voraussetzung, dass  $\xi_i(x_{a_i})$   $\delta$ -mal stetig differenzierbar auf der Menge

$$V_{a_i, c} = \{x_{a_i} \in \mathbb{R} \mid x \in V_c\} \text{ mit } V_c = \{x \in \mathbb{R}^n \mid V(x) \leq c\} \setminus 0$$

ist, lassen sich für  $k \in \mathbb{N}$  und  $0 < k \leq \delta$  die  $\xi_i(x_{a_i})$  mit

$$\xi_i(x_{a_i}) = l_i(x_{a_i}) + w_i \frac{x_{a_i}^k}{k!}$$

in Taylorreihen mit Restglied zerlegen. Hierbei sind die  $l_i(x_{a_i})$  durch

$$l_i(x_{a_i}) = \sum_{j=0}^{k-1} \left. \frac{d^j \xi_i(x_{a_i})}{dx_{a_i}^j} \right|_{x_{a_i}=0} \frac{x_{a_i}^j}{j!}$$

gegeben. Mit der Definition der Polynome

$$\begin{aligned} p(x) &= \frac{\partial V(x)}{\partial x} \left( g(x) + \sum_{i=1}^r h_i(x) l_i(x_{a_i}) \right) \\ q_i(x) &= \frac{\partial V(x)}{\partial x} h_i(x) \frac{x_{a_i}^k}{k!} \\ q(x) &= \left( q_1(x) \quad \dots \quad q_r(x) \right)^T \end{aligned}$$

kann das folgende Theorem vorgestellt werden.

**Satz 2.4: Einzugsgebiet für nichtlineare Systeme [12]**

$\underline{c}$  sei eine Lösung der Optimierungsaufgabe

$$\underline{c} = \sup_{c \in \mathbb{R}_+} c \quad \text{sodass} \quad p(x) + q(x)^T w < 0, \quad \forall x \in V_c, \quad \forall w \in \text{Eckpunkte}(w^I) \quad , \quad (2.11)$$

wobei  $w^I$  den Intervallvektor

$$w^I = \begin{pmatrix} w_1^I \\ \vdots \\ w_r^I \end{pmatrix}$$

mit  $w_i^I = [\underline{\sigma}_i, \bar{\sigma}_i]$  beschreibt. Wenn die  $\underline{\sigma}_i$  und  $\bar{\sigma}_i$

$$\underline{\sigma}_i \leq \frac{d^k \xi_i(x_{a_i})}{dx_{a_i}^k} \leq \bar{\sigma}_i \quad \forall x_{a_i} \in V_{a_i, c}$$

erfüllen, dann ist  $\underline{c}$  eine untere Schranke für  $c^*$  und  $\Omega_{\underline{c}} \subseteq \Omega_{c^*}$  ein gültiges Einzugsgebiet der Ruhelage.

Die Optimierungsaufgabe 2.11 kann analog zu den Lösungsverfahren für polynomiale Systeme über Summen von Quadraten als LMI-Aufgabe geschrieben werden. Allerdings muss hier relativ viel Aufwand betrieben werden, um die Aufgabenstellung für den LMI-Löser aufzubereiten. Weiterhin treten hier die bereits geschilderten numerischen Probleme beim Lösen von LMI-Aufgaben auf. Eine obere Schranke für  $c^*$  kann mit diesem Verfahren nicht bestimmt werden. Allerdings kann überprüft werden, ob ein bestimmtes  $\underline{c}$  gleich dem gesuchten  $c^*$  ist. Diese Überprüfung ist in [12, Theorem 3] angegeben. Weiterhin ist anzumerken, dass die hier geltenden Sätze nur für Systeme gelten, deren Systemdifferentialgleichung die Form (2.10) besitzen, die nichtpolynomiale Terme dürfen also jeweils nur von einer Variable abhängen. Es ist allerdings schon in [11] erwähnt, dass durch Anpassung der Taylorreihen die Methode auch auf Systeme mit nichtpolynomialen Termen in mehreren Variablen anwendbar ist.

Ein alternativer Zugang zur Bestimmung von Einzugsgebieten stellen Saleme et al. in [80] vor. Ähnlich dem Verfahren von Dilaver und Tibken werden in diesem Verfahren die Sätze von Ehlich und Zeller, Ruttmann und Gärtel angewandt. Zusätzlich werden Abschätzungen für die nichtpolynomialen Terme und ein Algorithmus zur Bestimmung von unterer und oberer Schranke von  $c^*$  vorgestellt.

Beim Vergleich der beiden existierenden Methoden fällt auf, dass bei der Methode von Saleme et al. keine Ableitungen der nichtpolynomialen Terme bestimmt werden müssen, während bei der Methode von Chesi für eine möglichst genaue innere Schranke eine Ableitung vom hohen Grad berechnet werden muss, was je nach Aufgabenstellung aufwendig sein kann. Weiterhin ist man bei der Methode von Chesi auf einen LMI-Löser angewiesen, welcher zu numerischen Problemen führen kann. Als weiterer Vorteil der Methode von Saleme et al. ist die Bestimmung von unterer und oberer Schranke zu nennen. Während hier sogar die gewünschte Genauigkeit vorgegeben werden kann, kann beim Verfahren von Chesi nur a posteriori verifiziert werden, ob die ermittelte untere Schranke gleich  $c^*$  ist.

Als Nachteile des Verfahrens von Saleme et al. sind die notwendigen Zustands-  
transformationen und Abschätzungen zu nennen, welche in höherem Maße aufwendig sein können. Weiterhin ist dieses Verfahren auf quadratische Ljapunow-funktionen beschränkt, während diese Einschränkung beim Verfahren von Chesi nicht notwendig ist.

## Intervallarithmetik

## 3.1 Einführung

Bei der Speicherung von reellen Zahlen in einem Digitalcomputer treten aufgrund der diskreten Speicherung und begrenzten Genauigkeit Rundungsfehler auf. Zwar ist die Darstellung einer Zahl bei doppelter Genauigkeit bis auf  $2^{-52} \approx 2.22 \times 10^{-16}$  genau möglich, allerdings kann sich dieser Fehler bei der Verarbeitung der Zahl vergrößern, ohne dass es auffällt. Ein Beispiel ist das von Rump in [77] vorgestellte Problem: Die Funktion

$$f(a, b) = 333.75b^6 + a^2 (11a^2b^2 - b^6 - 121b^4 - 2) + 5.5b^8 + \frac{a}{2b}$$

soll an den Stellen  $a = 77617$  und  $b = 33096$  ausgewertet werden. Hierzu wurde auf einer IBM S/370 das Ergebnis mit einfacher, doppelter und erweiterter Genauigkeit bestimmt; die Ergebnisse lauteten:

|                        |                       |
|------------------------|-----------------------|
| einfache Genauigkeit   | 1.172603 ...          |
| doppelte Genauigkeit   | 1.1726039400531 ...   |
| erweiterte Genauigkeit | 1.172603940053178 ... |

Obwohl alle Ergebnisse auf den Funktionswert 1.17260 ... hindeuten, sind alle Stellen und sogar das Vorzeichen falsch. Das tatsächliche Ergebnis lautet

$$f(a = 77617, b = 33096) = -0.8273960599468213681411650954 \dots$$

Eine Untersuchung der Ursache und eine Betrachtung des Problems mit IEEE-754-Arithmetik sind in [60] zu finden.

Werden solche Rundungsfehler nicht beachtet, können schwerwiegende Folgen auftreten. In [41] wird der Fall eines MIM-104 Patriot-Raketenabwehrsystems beschrieben, welches aufgrund von Rundungsfehlern im Steuerrechner [98] eine Boden-Boden-Rakete R-11 (Scud) nicht abfangen konnte, was zum Tod von 28 Menschen führte.

Die Idee liegt nun darin, eine Zahl  $x$ , welche sich eventuell nicht exakt im Floating-Point-Raster darstellen lässt, durch ein Intervall zu ersetzen, welches  $x$  garantiert enthält. Hierbei müssen die Grenzen des Intervalls so gewählt werden, dass diese im Floating-Point-Raster darstellbar sind. Rechnet man nun mit diesem Intervall, kann man zwar nicht die Genauigkeit erhöhen, jedoch kann man jederzeit ein Maß für die Genauigkeit beziehungsweise den Fehler bestimmen.

Aufgrund der vielfältigen Anwendungsgebiete der Intervallarithmetik gibt es für verschiedene Programmiersprachen Implementierungen der Intervallarithmetik. Historisch sind hierbei `Fortran-XSC` und `Pascal-XSC` [51] zu nennen. Für C und C++ stehen beispielsweise die Bibliotheken `FI_LIB` bzw. `FILIB++` [55] zur Verfügung. Das Computer-Algebra-System `Maple` kann mit Hilfe von `intpakX` [52] Intervallrechnungen (inklusive dem Intervall-Newton-Verfahren) durchführen. Für `MATLAB` wurde die umfangreiche Toolbox `INTLAB` [78] entwickelt, welche neben der Intervallarithmetik eine automatische Differenziation und Sekantenarithmetik liefert.

## 3.2 Notation

Ein abgeschlossenes Intervall  $x^I$  ist gegeben durch die untere Schranke  $\underline{x}$ , welche im Folgenden Infimum genannt wird, und die obere Schranke  $\bar{x}$ , im Folgenden Supremum genannt, mit  $\underline{x} \leq \bar{x}$ , sodass

$$\underline{x} \leq x \leq \bar{x} \quad \forall x \in x^I = [\underline{x}, \bar{x}] \quad (3.1)$$

gilt. Eine andere Möglichkeit, Intervalle eindeutig zu bezeichnen, ist die Angabe des Mittelpunkts

$$\check{x} = \text{mid } x^I = \frac{x + \bar{x}}{2}$$

und des Radius des Intervalls

$$\text{rad}(x^I) = \frac{x - \bar{x}}{2} .$$

Die Umrechnung in die Darstellung über Infimum und Supremum kann über

$$x^I = \left[ \check{x} - \text{rad} \left( x^I \right), \check{x} + \text{rad} \left( x^I \right) \right]$$

erfolgen. Alternativ zum Radius kann auch der Durchmesser des Intervalls angegeben werden, welcher dem doppelten Radius entspricht. Im Folgenden wird die Darstellung durch Infimum und Supremum verwendet.

Ein Punktintervall  $[x, x]$ , ein sogenanntes degeneriertes Intervall, stellt die reelle Zahl  $x$  dar. Hierbei ist zu beachten, dass sich bei der praktischen Implementierung nicht jede Zahl als degeneriertes Intervall darstellen lässt. Dies liegt an der schon im vorherigen Abschnitt geschilderten Genauigkeit des Floating-Point-Rasters.

Der Schnitt zweier Intervalle ist durch

$$x^I \cap y^I := \begin{cases} \emptyset & \bar{x} < \underline{y} \text{ oder } \bar{y} < \underline{x} \\ \left[ \max(\underline{x}, \underline{y}), \min(\bar{x}, \bar{y}) \right] & \text{sonst} \end{cases}$$

definiert. Überlappen sich zwei Intervalle in mindestens einem Punkt, das heißt, es gilt  $\bar{x} \geq \underline{y}$  oder  $\bar{y} \geq \underline{x}$ , dann ist die Vereinigung der beiden Intervalle das Intervall

$$x^I \cup y^I := \left[ \min(\underline{x}, \underline{y}), \max(\bar{x}, \bar{y}) \right] .$$

Sind die beiden Intervalle disjunkt, ist die Vereinigung eine nicht zusammenhängende Menge und kann nicht als Intervall dargestellt werden. Es ist jedoch mithilfe der Definition

$$x^I \sqcup y^I := \left[ \min(\underline{x}, \underline{y}), \max(\bar{x}, \bar{y}) \right]$$

möglich, eine Hülle zu bilden, welche die beiden Intervalle beinhaltet und selbst ein Intervall darstellt.

### 3.3 Arithmetische Operatoren für Intervalle

Ziel der Intervallarithmetik ist nicht nur die Einschließung mit Hilfe von Intervallen, sondern auch die Rechnung mit ihnen. Um für eine gegebene Funktion mit Intervallen rechnen zu können, muss die Funktion in ihre einzelnen Operationen aufgespalten werden. Für jede dieser Operationen muss dann die obere und untere Schranke des möglichen Wertebereichs bestimmt werden, damit Infimum und Supremum der Funktionsauswertung auf einem Intervall angegeben werden können.

Hierbei soll für die binären Operationen die Bedingung

$$x^I \circ y^I \subseteq \left\{ x \circ y, \quad x \in x^I \in \mathbb{R}, \quad y \in y^I \in \mathbb{R} \right\}$$

gelten. Um eine solche Einschließung zu garantieren, müssen die Regeln für die Verknüpfung  $\circ$  ausgearbeitet werden. Im Folgenden wird dies für die Grundrechenarten  $\circ = \{+, -, \cdot, \div\}$  hergeleitet (vergleiche auch [30,40,70]).

Bei der Addition (+) von zwei Intervallen ist es offensichtlich, dass der Wertevorrat durch die Addition der jeweiligen Schranken der Intervalle beschränkt wird. Hieraus folgt die Definition

$$x^I + y^I := \left[ \underline{x} + \underline{y}, \bar{x} + \bar{y} \right] \quad . \quad (3.2)$$

Bei der Subtraktion (−) des Intervalls  $y^I$  vom Intervall  $x^I$  berechnet sich der minimale Wert des Wertevorrats durch die Subtraktion des Supremums von  $y^I$  vom Infimum von  $x^I$ . Der größtmögliche Wert des Wertevorrats kann nur bei der Subtraktion des Infimums von  $y^I$  vom Supremum von  $x^I$  entstehen. Dies kann nachgewiesen werden, indem man die Intervalle  $\underline{x} \leq x \leq \bar{x}$  und  $-\bar{y} \leq -y \leq -\underline{y}$  addiert. Dann folgt

$$\underline{x} - \bar{y} \leq x - y \leq \bar{x} - \underline{y}$$

und hieraus die Regel für die Subtraktion

$$x^I - y^I := \left[ \underline{x} - \bar{y}, \bar{x} - \underline{y} \right] \quad . \quad (3.3)$$

Bei der Multiplikation ( $\cdot$ ) von zwei Intervallen ist die triviale Einschränkung für den Wertevorrat

$$x^I \cdot y^I := [\min(S), \max(S)] \quad \text{mit } S = \left\{ \underline{x} \cdot \underline{y}, \bar{x} \cdot \underline{y}, \underline{x} \cdot \bar{y}, \bar{x} \cdot \bar{y} \right\} \quad . \quad (3.4)$$

Eine ausführlichere Fallunterscheidung findet sich in [70, S. 13f].

Die Division ( $\div$ ) von zwei Intervallen  $x^I$  und  $y^I$  kann in eine Multiplikation von  $x^I$  mit dem Reziproken von  $y^I$  umgewandelt werden: Dann gilt für den Fall  $0 \notin y^I$

$$x^I \div y^I = x^I \cdot \frac{1}{y^I} \quad \text{mit } \frac{1}{y^I} = \left[ \frac{1}{\bar{y}}, \frac{1}{\underline{y}} \right] \quad . \quad (3.5)$$

Für den Fall  $0 \in y^I$  ist es mithilfe der erweiterten Intervallarithmetik (vergleiche [70, S. 109ff]) möglich, ein Ergebnis zu berechnen. Hierbei handelt es sich jedoch

entweder um ein uneigentliches Intervall oder um zwei disjunkte Intervalle. Im Folgenden wird auf die weitere Betrachtung dieses Falls verzichtet.

Für Operationen mit einem Operanden (unäre Operationen) müssen genauso entsprechende Betrachtungen durchgeführt werden. So gelten beispielsweise für die Potenzierung von  $x^I$  mit  $m \in \mathbb{N}$

$$(x^I)^{2m} \begin{cases} [x^{2m}, \bar{x}^{2m}] & x \geq 0 \\ [\bar{x}^{2m}, x^{2m}] & \bar{x} \leq 0 \\ [0, \max(x^{-2m}, \bar{x}^{2m})] & x \leq 0 \leq \bar{x} \end{cases} \quad (3.6)$$

und

$$(x^I)^{2m-1} = [x^{2m-1}, \bar{x}^{2m-1}] \quad .$$

Da der Logarithmus und die Exponentialfunktion streng monoton steigend sind, können die Intervallerweiterungen

$$\begin{aligned} e^{x^I} &= [e^{\underline{x}}, e^{\bar{x}}] \\ \log_b(x^I) &= [\log_b \underline{x}, \log_b \bar{x}] \end{aligned}$$

angewandt werden. Für weitere Operationen kann die Intervalloperation äquivalent hergeleitet werden.

### 3.4 Intervallvektoren

Bis jetzt beschränkte sich die Betrachtung auf Intervalle. Im Folgenden soll die Betrachtung auf  $n$ -dimensionale Intervallvektoren hin erweitert werden. Hierbei bezeichnet

$$x^I = \begin{pmatrix} x_1^I \\ \vdots \\ x_n^I \end{pmatrix} = \begin{pmatrix} [\underline{x}_1, \bar{x}_1] \\ \vdots \\ [\underline{x}_n, \bar{x}_n] \end{pmatrix}$$

den Intervallvektor mit  $n$  Intervallen  $x_i^I$ , der alle Vektoren  $x = (x_1, \dots, x_n)^T$  mit der Eigenschaft

$$\underline{x}_1 \leq x_1 \leq \bar{x}_1, \quad \dots, \quad \underline{x}_n \leq x_n \leq \bar{x}_n$$

beinhaltet. Die meisten Eigenschaften von Intervallen können angepasst auch auf Intervallvektoren angewandt werden, wenn diese komponentenweise betrachtet



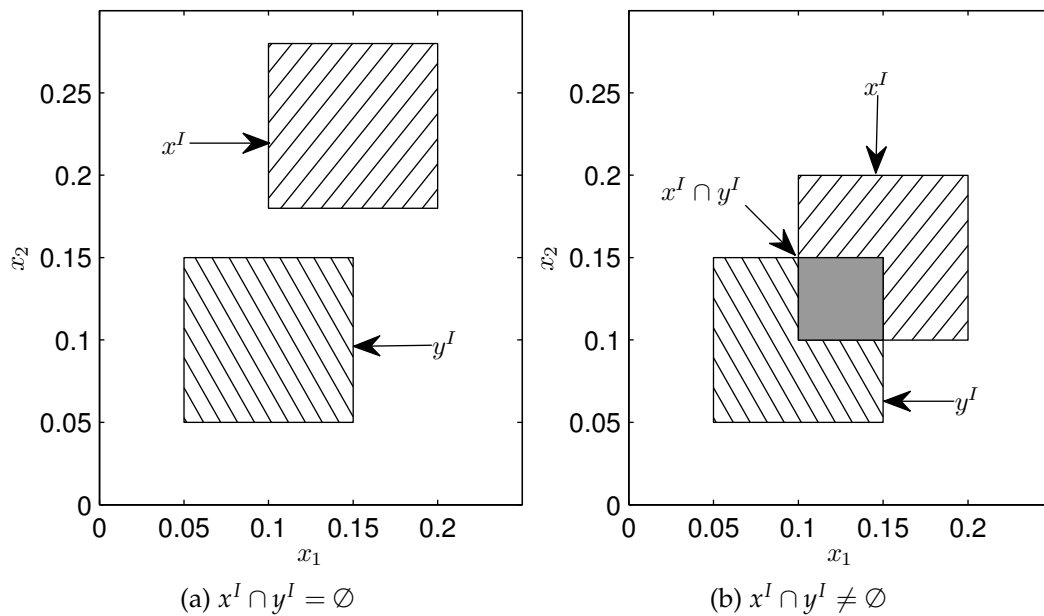


Abbildung 3.1: Schnitt von zwei Intervallvektoren

werden. Der Mittelpunkt eines Intervallvektors wird mit

$$\check{x} = \text{mid}(x^I) = \begin{pmatrix} \check{x}_1 \\ \vdots \\ \check{x}_n \end{pmatrix}$$

bezeichnet. Um den Schnitt von zwei Intervallvektoren zu berechnen, bestimmt man den Schnitt der einzelnen Komponenten der Intervallvektoren. Wenn

$$x_i^I \cap y_i^I = \emptyset$$

für mindestens ein  $i$  gilt, dann ist der Schnitt der Intervallvektoren  $x^I$  und  $y^I$  auch  $\emptyset$ , wie in Abbildung 3.1a dargestellt. Andernfalls gilt für die Vereinigung

$$x^I \cap y^I = \begin{pmatrix} x_1^I \cap y_1^I \\ \vdots \\ x_n^I \cap y_n^I \end{pmatrix},$$

was auch der Abbildung 3.1b entnommen werden kann.

## 3.5 Intervallerweiterung

Das Ziel der Intervallarithmetik ist die Bestimmung von Schranken für den Wertebereich einer Funktionen auf einem Intervall bzw. Intervallvektor. Hierbei bezeich-

net man mit  $F^I : \mathbb{R}^n \mapsto \mathbb{R}$  genau dann eine Intervallerweiterung von  $f : \mathbb{R}^n \mapsto \mathbb{R}$ , wenn

$$F^I(x^I) \supseteq \{f(x) | x \in x^I\} \quad (3.7)$$

gilt. Das Infimum von  $F^I(x^I)$  wird mit  $\underline{F}$ , das Supremum als  $\bar{F}$  bezeichnet, es gilt folglich

$$F^I(x^I) = [\underline{F}, \bar{F}] \quad .$$

Allerdings ist die Darstellung von  $F^I(x^I)$  nicht eindeutig, da theoretisch unendlich viele Intervalle angegeben werden können, welche die Inklusionseigenschaft (3.7) erfüllen. Es ist allerdings das Ziel der Intervallarithmetik, eine Intervallerweiterung möglichst einfach bestimmen zu können. Die einfachste Intervallerweiterung ist die sogenannte natürliche oder naive Intervallerweiterung. Hierzu werden alle in der Funktion vorkommenden Operationen durch die entsprechenden Intervalloperationen und alle Fließkommazahlen durch Intervalle ersetzt.

### Beispiel 3.1

Die Funktion

$$f(x) = x^2 - 2x + 3$$

soll auf dem Intervall  $[0, 2]$  betrachtet werden. Durch Austausch der Operatoren gemäß den Gleichungen (3.2), (3.3), (3.4) und (3.6) folgt die natürliche Intervallerweiterung

$$\begin{aligned} F^I([0, 2]) &= [0, 2]^2 - [2, 2] \cdot [0, 2] + [3, 3] \\ &= [0, 4] - [0, 4] + [3, 3] \\ &= [3, 7] - [0, 4] \\ &= [-1, 7] \quad . \end{aligned}$$

Wie Abbildung 3.2 demonstriert, ist der exakte Wertevorrat von  $f(x)$  auf  $x^I$  durch das Intervall  $[2, 3]$  gegeben, mithilfe der natürlichen Intervallerweiterung erhält man das Intervall  $[-1, 7]$ , welches  $[2, 3]$  zwar vollständig einschließt, allerdings nicht exakt, das heißt

$$\{f(x) | x \in x^I\} \subsetneq F^N(x^I) \quad .$$

Dieses Problem wird als Abhängigkeitsproblem oder Überschätzung bezeichnet. Es tritt dann auf, wenn mindestens eine Intervallvariable mehrfach im Funktionsterm vorkommt. Während dies bei einer Addition keine Folgen hat, ist eine Überschätzung schon bei der Subtraktion festzustellen: So ist das Ergebnis von

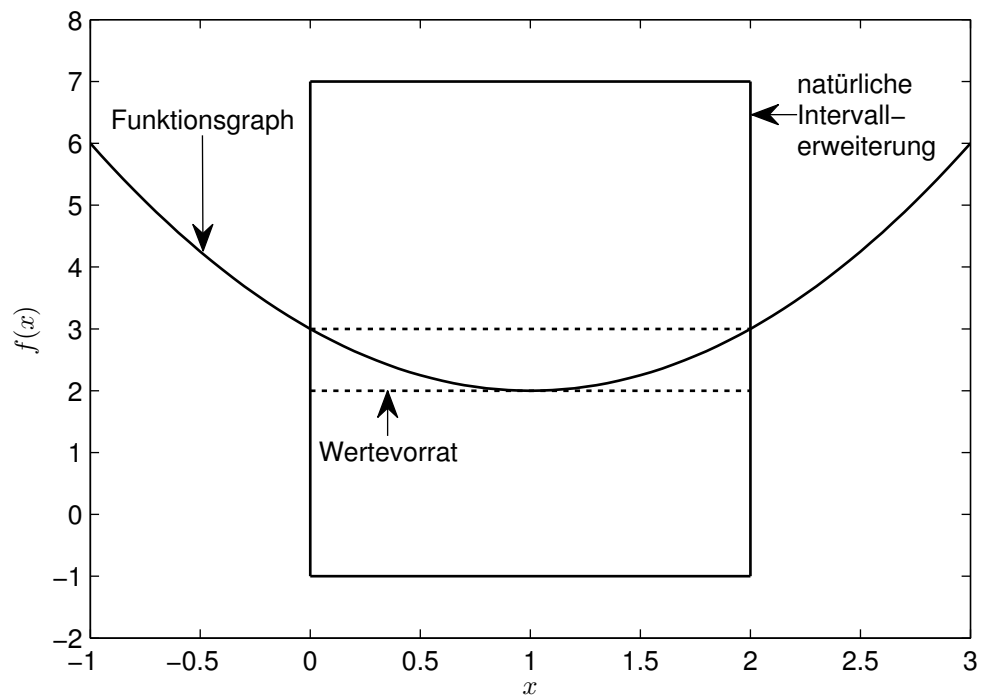


Abbildung 3.2: Infimum und Supremum von  $F^I(x^I)$  auf  $x^I = [0, 2]$  verglichen mit dem Funktionsverlauf von  $f(x)$  auf  $x^I$  (aus Beispiel 3.1)

Gerne wieder

$x^I - x^I$  nicht wie erwartet  $[0, 0]$ , sondern  $[x - \bar{x}, \bar{x} - x]$ . Das Problem der Überschätzung lässt sich nur dadurch verhindern, dass die Funktion soweit umgestellt wird, dass jede Intervallvariable nur einmal im Funktionsterm auftaucht.  $f(x)$  aus Beispiel 3.1 lässt sich auch durch  $f(x) = (x - 1)^2 + 2$  beschreiben. Die Auswertung der natürlichen Intervallerweiterung ergibt hier

$$F^N(x^I) = ([0, 2] - [1, 1])^2 + [2, 2] = [2, 3] \quad ,$$

es findet also keine Überschätzung statt.

Nun kann eine solche Umstellung im Allgemeinen nicht immer erfolgen, weswegen weitere Methoden entwickelt wurden, die Überschätzung zu reduzieren. Eine dieser Methoden ist die sogenannte Bisektion. Hierbei wird ausgenutzt, dass die Überschätzung linear mit der Verkleinerung der Intervalle abnimmt [17]. Hierzu muss das zu untersuchende Intervall in mehrere Teilintervalle aufgeteilt werden, woraufhin jeweils das Infimum und Supremum dieser Teilintervalle bestimmt und danach die Vereinigung der Teilintervalle berechnet werden.

**Beispiel 3.2**

Die Funktion

$$f(x) = x^2 - 2x + 3$$

soll auf dem Intervall  $[0, 2]$  betrachtet werden. Hierzu wird das Intervall in zwei gleich große Teilintervalle  $x^{I_1} = [-0, 1]$  und  $x^{I_2} = [1, 2]$  aufgeteilt. Die natürliche Intervallerweiterung für  $x^{I_1}$  bestimmt sich aus

$$\begin{aligned} F^N([0, 1]) &= [0, 1^2] - [2, 2] \cdot [0, 1] + [3, 3] \\ &= [0, 1] - [0, 2] + [3, 3] \\ &= [3, 4] - [0, 2] \\ &= [1, 4] \end{aligned}$$

und für  $x^{I_2}$

$$\begin{aligned} F^N([1, 2]) &= [1, 2^2] - [2, 2] \cdot [1, 2] + [3, 3] \\ &= [1, 4] - [2, 4] + [3, 3] \\ &= [4, 7] - [2, 4] \\ &= [0, 5] \quad . \end{aligned}$$

Die Vereinigung von  $F^N(x^{I_1})$  und  $F^N(x^{I_2})$  berechnet sich zu

$$[1, 4] \cup [0, 5] = [0, 5] \quad .$$

Wie auch Abbildung 3.3 entnommen werden kann, liefern die neu berechneten Schranken eine bessere Einschließung für den tatsächlichen Wertebereich  $[2, 3]$  von  $f(x)$  auf  $x^I$  als die natürliche Intervallerweiterung.

Die Überschätzung der natürlichen Intervallerweiterung bei der Bisektion verringert sich linear mit der Verkleinerung. Es gibt jedoch auch weitere Intervallerweiterungen, welche insbesondere bei kleinen Intervallen eine geringere Überschätzung bieten können, zum Beispiel die Mittelpunktsform.

Um die Mittelpunktsform herzuleiten, wird die Taylorreihe der Funktion  $f : \mathbb{R}^n \mapsto \mathbb{R}$  an der Stelle  $\check{x}$  (Mittelpunkt des Intervalls  $x^I$ ) nach dem konstanten Term abgebrochen:

$$f(x) = f(\check{x}) + \left( \frac{\partial f}{\partial x}(\zeta) \right) (x - \check{x})$$

Dabei bezeichnet  $\zeta \in \mathbb{R}^n$  eine Stelle zwischen  $x$  und  $\check{x}$ . Da  $x$  und  $\check{x}$  in  $x^I$  liegen, liegen auch alle Zwischenstellen in  $x^I$ , weswegen man  $x$  und  $\zeta$  durch  $x^I$  ersetzen kann, um die Mittelpunktsform von  $f$  zu erhalten:

$$F^M(x^I) = f(\check{x}) + \left( F'^I(x^I) \right) (x^I - \check{x}) \quad (3.8)$$

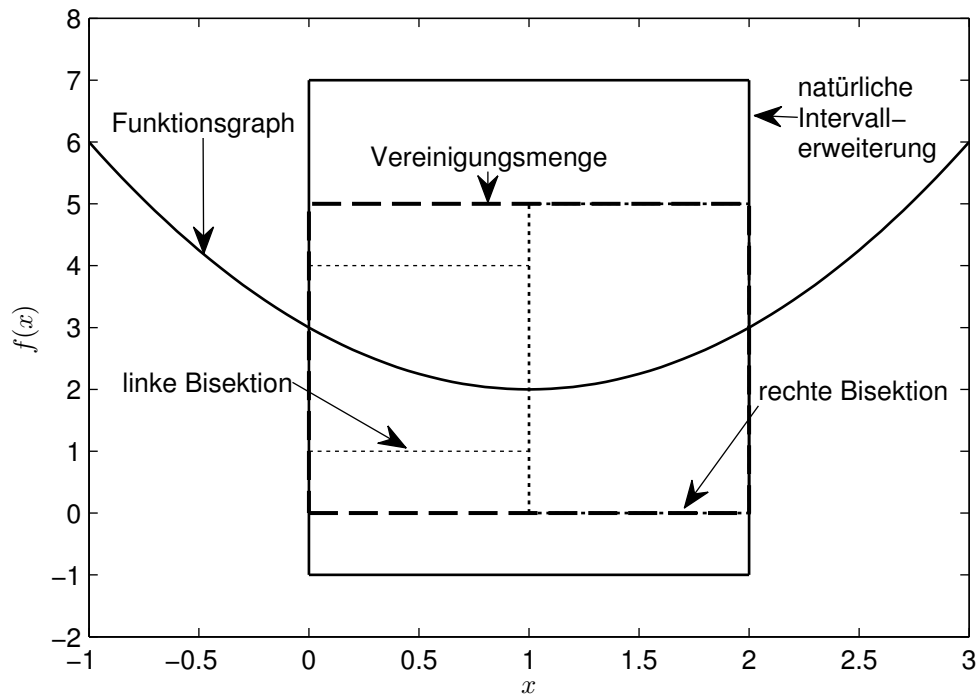


Abbildung 3.3: Vergleich natürlicher Intervallerweiterung und Bisektion (aus Beispiel 3.2)

Hierbei bezeichnet  $F^I(x^I)$  eine Intervallerweiterung der Ableitung von  $f$  nach  $x$  (für  $n \geq 2$  die Intervallerweiterung des Zeilenvektors mit den partiellen Ableitungen von  $f$ ), welche beispielsweise mithilfe der natürlichen Intervallerweiterung ermittelt wurde. Dann gelten

$$F^I(x^I) \supseteq \left\{ \frac{\partial f}{\partial x}(x) \mid x \in x^I \right\}$$

und

$$F^M \supseteq \left\{ f(x) \mid x \in x^I \right\} .$$

### Beispiel 3.3

Betrachtet wird wie in Beispiel 3.1 die Funktion

$$f(x) = x^2 - 2x + 3$$

auf dem Intervall  $[0, 2]$ . Die Ableitung von  $f(x)$  bezüglich  $x$  lautet

$$f'(x) = 2x - 2 ,$$

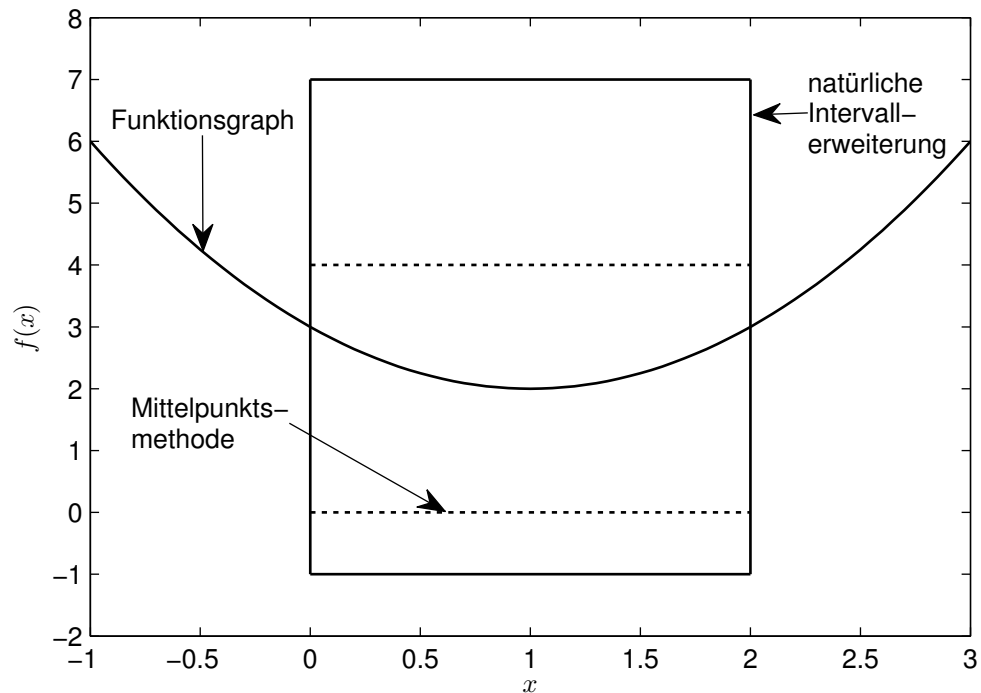


Abbildung 3.4: Vergleich natürlicher Intervallerweiterung und Mittelpunktsform (aus Beispiel 3.3)

womit sich die Schranken

$$\begin{aligned}
 FM([0, 2]) &= (1^2 - 2 \cdot 1 + 3) + ([2, 2] \cdot [0, 2] - [2, 2]) \cdot ([0, 2] - [1, 1]) \\
 &= [2, 2] + ([0, 4] - [2, 2]) \cdot [-1, 1] \\
 &= [2, 2] - [-2, 2] \\
 &= [0, 4]
 \end{aligned}$$

für  $f(x)$  auf  $x^I$  berechnen lassen. Die mithilfe der Mittelpunktsform ermittelten Schranken sind in Abbildung 3.4 den mithilfe der natürlichen Intervallerweiterung bestimmten Schranken gegenübergestellt.

Eine weitere Intervallerweiterung ist die Sekantenform. Die Sekante einer Funktion zwischen den Stellen  $x$  und  $y$  ist gegeben durch

$$s(x, y) := \frac{f(x) - f(y)}{x - y}, \quad (3.9)$$

es handelt sich also um den Differenzialquotienten. Stellt man (3.9) nach  $f(x)$  um, so erhält man

$$f(x) = f(y) + s(x, y) \cdot (x - y) \quad .$$

Kann man  $s(x, y)$  zum Beispiel durch Polynomdivision soweit vereinfachen, dass keine Division durch null mehr auftreten kann, ist es möglich, eine Einschließung

des Wertevorrats von  $f$  auf  $x^I$  anzugeben. Hierzu ersetzt man  $x$  durch  $x^I$  und  $y$  durch  $\check{x}$  und erhält die Intervallerweiterung

$$F^S(x^I) = f(\check{x}) + S^I(x^I, \check{x})(x^I - \check{x}) \quad ,$$

wobei  $S^I(x^I, \check{x})$  eine Intervallerweiterung der vereinfachten Sekantengleichung (3.9) darstellt.

### Beispiel 3.4

Betrachtet wird die Funktion

$$f(x) = x^2 - 2x + 3$$

auf dem Intervall  $[0, 2]$ . Die allgemeine Gleichung für die Sekante lässt sich mit Polynomdivision umschreiben zu

$$\begin{aligned} s(x, y) &= \frac{x^2 - 2y + 3 - (y^2 - 2y + 3)}{x - y} \\ &= \frac{x^2 - y^2 - (2x - 2y)}{x - y} \\ &= x + y - 2 \quad . \end{aligned}$$

Daraus folgt

$$S^I(x^I, \check{x}) = x^I + \check{x} - 2.$$

Nun kann die Sekantenform mit

$$\begin{aligned} F^S([0, 2]) &= (1^2 - 2 \cdot 1 + 3) + ([0, 2] + [1, 1] - [2, 2]) \cdot ([0, 2] - [1, 1]) \\ &= [2, 2] + ([1, 3] - [2, 2]) \cdot [-1, 1] \\ &= [2, 2] + [-1, 1] \cdot [-1, 1] \\ &= [2, 2] - [-1, 1] \\ &= [1, 3] \end{aligned}$$

bestimmt werden. Wie in Abbildung 3.5 zu sehen ist, ist die mit der Sekantenform ermittelte obere Schranke bereits exakt und die untere Schranke näher am Minimum der Funktion  $f(x)$  als die mit der natürlichen Intervallerweiterung bestimmte Schranke.

In den Beispielen 3.3 und 3.4 fallen die mithilfe der Mittelpunktsform bzw. Sekantenform berechneten Schranken für den Wertebereich schärfer aus als die in

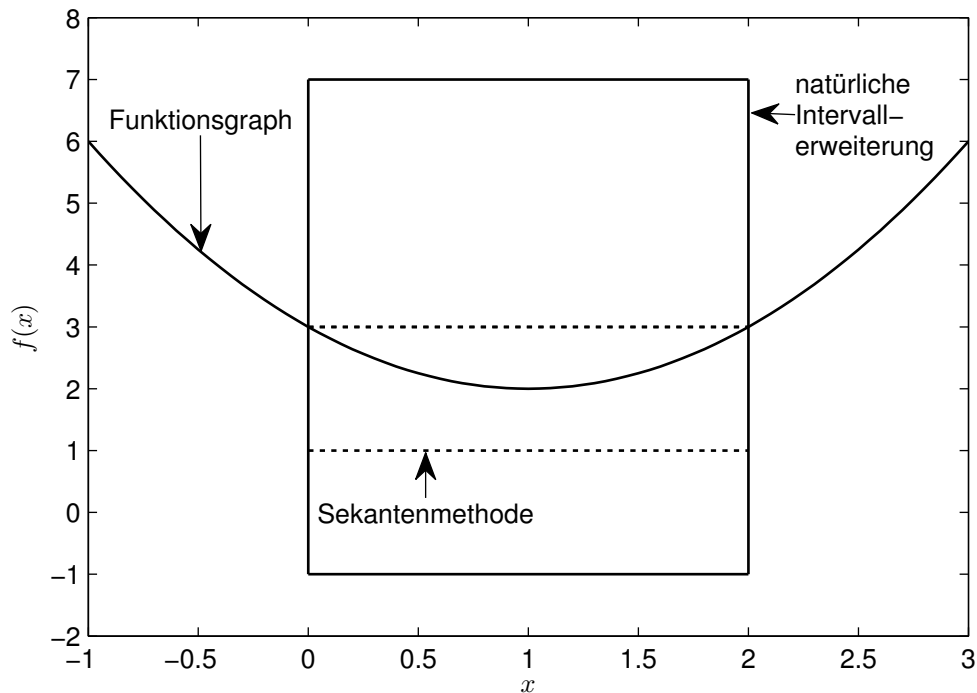


Abbildung 3.5: Vergleich natürlicher Intervallerweiterung und Sekantenform (aus Beispiel 3.4)

Beispiel 3.1 mithilfe der natürlichen Intervallerweiterung bestimmten Schranken. Dies kann jedoch nicht verallgemeinert werden. Insbesondere bei weiten Intervallen liefert die natürliche Intervallerweiterung engere Schranken als die Schranken zweiter Ordnung. Wenn möglich, sollten alle drei Intervallerweiterungen bestimmt und danach sollte der Schnitt der Ergebnisintervalle mit

$$F^I(x^I) = F^N(x^I) \cap F^M(x^I) \cap F^S(x^I)$$

berechnet werden. Da alle Intervallerweiterungen garantiert den Wertevorrat von  $f(x)$  auf dem Intervall  $x^I$  einschließen, enthält auch der Schnitt den Wertevorrat.

## 3.6 Globaler Optimierungsalgorithmus

Eine Anwendung der Intervallarithmetic ist die Ermittlung von globalen Schranken von Funktionsauswertungen auf einem Intervall. Der hierfür genutzte Algorithmus geht auf Skelboe [84] zurück, wurde danach unter anderem in [39, 40] weiterentwickelt und in modifizierter Form [70] als Skelboe-Moore-Algorithmus zur Ermittlung von oberen und unteren Schranken einer Funktion auf einem



Intervall vorgestellt. Insofern ist der in [70] vorgestellt Algorithmus selbst auch wieder eine Intervallerweiterung.

Hier soll nun eine leicht vereinfachte Version aufgezeigt werden, welche das globale Minimum einer Funktion  $f(x)$  auf dem Intervall  $x^{l_0} \in \mathbb{R}^n$  bis auf eine Genauigkeit  $\varepsilon$  bestimmen kann. Die Grundstruktur dieses Algorithmus wird in Kapitel 4.1 abgewandelt, um ihn zur Lösung des Optimierungsproblems (2.6) nutzen zu können.

Im Algorithmus wird während der Laufzeit eine Liste  $\mathcal{L}$  verwendet, welche nach  $i$  sortierte Paare  $(\gamma_i, x^{l_i})$  mit der Eigenschaft

$$\gamma_1 \leq \gamma_2 \leq \dots \leq \gamma_r \quad . \quad (3.10)$$

enthält. Initialisiert wird die Liste, indem man mit

$$\begin{aligned} \left[ \underline{F}, \bar{F} \right] &= F(x^{l_0}) \\ \underline{f} &= \underline{F} \\ \bar{f} &= f(\check{x}) \quad \text{mit } \check{x} \text{ Mittelpunkt von } x^{l_0} \end{aligned}$$

initiale Schranken für das Minimum von  $f(x)$  auf  $x^{l_0}$  bestimmt und in die neu zu erzeugende Liste  $\mathcal{L}$  das Paar  $(\underline{f}, x^{l_0})$  einträgt. Nach dieser Initialisierung kann der eigentliche Algorithmus gestartet werden:

```

while  $\bar{f} - \underline{f} \geq \varepsilon$  do
  Entferne das erste Paar  $(\underline{f}_1, x^{l_1})$  aus der Liste  $\mathcal{L}$ 
  Bisektiere  $x^{l_1}$  entlang der längsten Kante in zwei Intervalle  $y^{l_1}$  und  $y^{l_2}$ 
  for  $i = 1 \rightarrow 2$  do
     $y^l \leftarrow y^{l_i}$ 
     $\check{y} \leftarrow \text{mid}(y^l)$ 
    if  $f(\check{y}) < \bar{f}$  then
       $\bar{f} \leftarrow f(\check{y})$ 
      lösche alle Paare  $(\gamma_i, x^{l_i})$  mit der Eigenschaft  $\gamma_i > \bar{f}$  aus der Liste  $\mathcal{L}$ 
      {diese Intervalle können kein Minimum enthalten}
    end if
     $\left[ \underline{F}, \bar{F} \right] \leftarrow F(y^l)$ 
    if  $\underline{F} < \bar{f}$  then
      sortiere das Paar  $(\underline{F}, y^l)$  zurück in die Liste  $\mathcal{L}$  gemäß der Vorschrift in
      (3.10)
    end if

```

```

     $\underline{f} \leftarrow \gamma_1$ 
end for
end while

```

Nach Abbruch des Algorithmus wird das Minimum  $f^*$  von  $f(x)$  auf  $x^{I_0}$  durch

$$\underline{f} \leq f^* \leq \bar{f}$$

eingeschlossen. Die Liste  $\mathcal{L}$  enthält dann alle Kandidaten, welche das Minimum enthalten können. Hierbei handelt es sich nicht nur um Intervalle, welche das Minimum beinhalten, sondern auch um solche, welche genügend nah am Minimum liegen, sodass das mithilfe der Intervallarithmetik bestimmte Infimum kleiner als die globale obere Schranke ist. Diesen unter anderem in [22, 47, 49] beschriebenen Effekt bezeichnet man als Clusterproblem. Auch wenn bereits Verfahren zur Reduzierung vorgestellt wurden, zum Beispiel die Exclusion-Regions-Methode von Schichl und Neumaier [82], liegt die Lösung dieses Problems in einem aktuellen Forschungsfeld.

### 3.7 Lösen von Intervallgleichungssystemen

Eine Erweiterung des globalen Optimierungsalgorithmus ist es, die notwendige Bedingung für das Vorliegen eines Minimums zu überprüfen. An den Extremstellen  $x^*$  einer Funktion  $f$  gilt

$$\frac{\partial f}{\partial x}(x^*) = 0 \quad .$$

Unter der Voraussetzung, dass das Minimum von  $f$  nicht an den Randpunkten des Startintervalls liegt, können alle Intervalle  $x^{I_j}$  verworfen werden, deren Intervallableitung des Gradienten die Null nicht enthält, das heißt

$$0 \neq F^{I_j}(x^{I_j}) \quad .$$

Dies bedeutet im Gegenzug aufgrund der Überschätzung jedoch nicht, dass die Intervalle, welche

$$0 \in F^{I_j}(x^{I_j}) \tag{3.11}$$

erfüllen, tatsächlich ein Extremum enthalten. Ist die Gleichung (3.11) nicht erfüllt, kann das Intervall  $x^{I_j}$  verworfen werden. Diesen Test bezeichnet man als Gradiententest.

Wenn Gleichung (3.11) erfüllt ist, kann das sogenannte Intervall-Newton-Verfahren angewandt werden, um das Intervall, welches die Null enthalten kann, zu verkleinern.

Im Allgemeinen soll für ein System von Gleichungen

$$g(x) = \begin{pmatrix} g_1(x) \\ g_2(x) \\ \vdots \\ g_m(x) \end{pmatrix},$$

zum Beispiel für einen Gradienten einer Funktion, die Nullstelle(n)  $x^*$  mit der Eigenschaft

$$\begin{aligned} g_1(x^*) &= 0 \\ g_2(x^*) &= 0 \\ &\vdots \\ g_m(x^*) &= 0 \end{aligned}$$

ermittelt werden. Die Herleitung für das Intervall-Newton-Verfahren in diesem Abschnitt folgt dabei der aus [70]. Für eine kontinuierlich differenzierbare Funktion  $f : \mathbb{R} \mapsto \mathbb{R}$  weiß man aufgrund des Mittelwertsatzes der Differenzialrechnung, dass für

$$g(x) = g(y) + g'(\xi)(x - y)$$

ein  $\xi$  zwischen  $x$  und  $y$  existiert. Gesucht wird nun auf dem Intervall  $x^{I_0}$  eine Lösung der Gleichung  $g(x) = 0$ . Sofern eine Lösung existiert, erfüllt sie die Bedingung

$$0 = g(y) + g'(\xi)(x - y)$$

mit  $y \in x^{I_0}$ . Hier wird der Einfachheit halber  $y = \check{x}$  gewählt. Dann gilt

$$x = \check{x} - \frac{g(\check{x})}{g'(\xi)}.$$

Hierbei ist von der Stelle  $\xi$  nur bekannt, dass sie zwischen  $x \in x^{I_0}$  und  $\check{x} \in x^{I_0}$  liegt. Dies lässt sich durch die Bedingung  $\xi \in x^{I_0}$  verallgemeinern. Mit der Intervallerweiterung  $G^{II}(x^{I_0})$  für die Ableitung von  $g$  nach  $x$  lässt sich das Intervall durch die Iterationsvorschrift

$$x^{I_{k+1}} = x^{I_k} \cap N(x^{I_k}) \quad (k = 0, 1, 2, \dots)$$

| Intervall | $\underline{x}$ | $\bar{x}$ |
|-----------|-----------------|-----------|
| $x^{I_0}$ | 1.0000000       | 4.0000000 |
| $x^{I_1}$ | 1.3749999       | 2.2187500 |
| $x^{I_2}$ | 1.9706756       | 2.0773260 |
| $x^{I_3}$ | 1.9994967       | 2.0007548 |

Tabelle 3.1: Verkleinerung des Startintervalls mithilfe des Intervall-Newton-Verfahrens im Beispiel 3.5

mit dem sogenannten Newtonoperator

$$N(x^I) = \check{x} - \frac{g(\check{x})}{G^I(x^I)} \quad (3.12)$$

verkleinern. Diese Verbesserung lässt sich allerdings nur ausführen, wenn  $0 \notin G^I(x^{I_0})$  gilt.

**Beispiel 3.5**

Es soll die Nullstelle der Funktion  $g(x) = x^2 - 4$  auf dem Intervall  $x^{I_0} = [1, 4]$  ermittelt werden. Die Ableitung  $g'(x) = 2x$  kann durch  $G^I(x^I) = 2x^I$  eingeschlossen werden. Hieraus bestimmt sich der Newtonoperator zu

$$N(x^I) = \check{x} - \frac{\check{x}^2 - 4}{2x^I}$$

und die Iterationsvorschrift lautet

$$x^{I_{k+1}} = x^{I_0} \cap \left\{ \check{x}^{(k)} - \frac{(\check{x}^{(k)})^2 - 4}{2x^{I_k}} \right\} ,$$

wobei  $\check{x}^{(k)}$  den Mittelpunkt des Intervalls  $x^{I_k}$  bezeichnet. Wie Tabelle 3.1 zeigt, wird das Intervall effektiv verringert, ohne die Eigenschaft der Einschließung der Lösung  $x^* = 2$  zu verlieren. In [69] wird gezeigt, dass das Intervall-Newton-Verfahren eine quadratische Konvergenz für den eindimensionalen Fall besitzt.

Eine weitere wichtige Eigenschaft ist das folgende Theorem:

**Theorem 3.1** (Inklusionseigenschaft). *Liegt  $N(x^I)$  vollständig in  $x^I$ , das heißt*

$$N(x^I) \subset x^I \quad ,$$

*dann besitzt  $g(x)$  auf dem Intervall  $x^I$  eine eindeutige Nullstelle.*

Der Beweis kann beispielsweise in [9] nachgeschlagen werden.

Für den mehrdimensionalen Fall ist die Ermittlung des Newtonoperators nicht so einfach durchführbar, da es sich in diesem Fall bei  $G^I(x^{I_k})$  um eine mehrdimensionale Intervallmatrix handelt, von der die Inverse berechnet werden muss, vergleiche auch [40] und [70]. Eine Möglichkeit ist der sogenannte Krawczykoperator [9,20,40,48,70], der wie folgt gebildet wird:

Die Matrix  $L$  soll die Inverse von  $G^I(x^{I_k})$  approximieren. Ist das zu untersuchende Intervall nicht weit, kann hierzu beispielsweise

$$L = \left( \text{mid} \left( G^I \left( x^{I_k} \right) \right) \right)^{-1} .$$

gewählt werden. Dann wird der Krawczykoperator gebildet durch

$$K(x^I) = \check{x} - LG^I(\check{x}) + (I - L)(x^I - \check{x}) , \quad (3.13)$$

wobei  $I$  die Einheitsmatrix in der Dimension von  $L$  darstellt. Die Iterationsvorschrift lautet nun

$$x^{I_{k+1}} = x^{I_k} \cap K(x^{I_k}) \quad (k = 0, 1, 2, \dots) . \quad (3.14)$$

Auch für den Krawczykoperator gilt die in Theorem 3.1 beschriebene Inklusionseigenschaft der Nullstelle.

Als Alternativen zum Krawczyk-Schritt wären das Gauß-Seidel-Iterationsverfahren [48] oder das Gauß-Eliminationsverfahren [2] zu nennen. Für den Anwendungsfall in dieser Arbeit bringen diese Alternativen jedoch keine Vorteile.

## KAPITEL VIER

# Berechnung gesicherter Einzugsgebiete für nichtlineare Systeme

### 4.1 Erster algorithmischer Ansatz

Im Folgenden soll nun ein Algorithmus entwickelt werden, um für die Lösung der in Kapitel 2.2 hergeleiteten Optimierungsaufgabe

$$\begin{aligned} V(x^*) &= c^* \\ \dot{V}(x^*) &= 0 \\ x^* &\neq 0 \end{aligned}$$

einschließende Schranken zu bestimmen. Hierbei es sich wie bei dem in Kapitel 3.6 vorgestellten Optimierungsalgorithmus um einen Branch-and-Bound-Algorithmus, welcher jedoch grundlegende Unterschiede aufweist. Eine initiale Intervallbox soll in kleinere Intervallboxen aufgeteilt werden und anhand der Intervallerweiterung der Ljapunowfunktion  $V(x)$  und ihrer zeitlichen Ableitung  $\dot{V}(x)$  auf diesen Intervallboxen klassifiziert werden. Ziel des Algorithmus ist es, eine garantierte Einschließung für  $c^*$  sowie eine Liste mit Kandidaten für  $x^*$  mit zu erhalten.

Mithilfe der Intervallerweiterung aus Kapitel 3.5 besteht die Möglichkeit, für eine gegebene Ljapunowfunktion  $V(x)$  und ihre zeitliche Ableitung  $\dot{V}(x)$  die

| Liste $\mathcal{L}$ |             |             |          |             |
|---------------------|-------------|-------------|----------|-------------|
| $\underline{V}_1$   | $x_1^{I_1}$ | $x_2^{I_1}$ | $\cdots$ | $x_n^{I_1}$ |
| $\underline{V}_2$   | $x_1^{I_2}$ | $x_2^{I_2}$ | $\cdots$ | $x_n^{I_2}$ |
| /                   |             |             |          |             |
| $\underline{V}_r$   | $x_1^{I_r}$ | $x_2^{I_r}$ | $\cdots$ | $x_n^{I_r}$ |

Abbildung 4.1: Aufbau der Liste  $\mathcal{L}$

Einschließungen

$$V^I(x^I) = \left[ \underline{V}, \bar{V} \right] \tag{4.1}$$

und

$$\dot{V}^I(x^I) = \left[ \underline{\dot{V}}, \bar{\dot{V}} \right] \tag{4.2}$$

mit den Eigenschaften

$$\begin{aligned} \underline{V} &\leq \min_{x \in x^I} V(x), & \bar{V} &\geq \max_{x \in x^I} V(x) \\ \underline{\dot{V}} &\leq \min_{x \in x^I} \dot{V}(x), & \bar{\dot{V}} &\geq \max_{x \in x^I} \dot{V}(x) \end{aligned} \tag{4.3}$$

zu bestimmen.

Das optimale  $c^*$  soll bis auf einen maximalen Fehler  $\varepsilon$  bestimmt werden, das heißt

$$c^* \in \left[ \underline{c}, \bar{c} \right] \quad \text{und} \quad \bar{c} - \underline{c} \leq \varepsilon \quad .$$

Weiterhin müssen vor dem Start des Algorithmus initiale Schranken für  $c^*$  gesetzt werden. Die obere Schranke  $\bar{c}$  kann ohne Beschränkung der Allgemeinheit auf  $+\infty$  gesetzt werden. Die initiale untere Schranke  $\underline{c} = c_{\text{init}}$  muss entweder mit einer bereits bestehenden Methode oder der Abschätzung aus [92] bestimmt werden. Hilfsweise kann ein sehr kleiner positiver Wert geraten werden, was jedoch bei einem Wert größer  $c^*$  zu einem Fehler beim Ablauf des Algorithmus führen kann.

Des Weiteren benötigt müssen während der Laufzeit die zu untersuchenden Intervalle gespeichert werden. Hierzu bietet sich eine geordnete Liste an, wie sie in Abbildung 4.1 dargestellt ist, an. Hierzu werden jeweils Tupel bestehend aus dem Intervall  $x^{I_j}$  und der unteren Schranke von  $V(x)$  auf  $x^{I_j}$  gespeichert, wobei die Liste nach diesen unteren Schranken aufsteigend sortiert sein soll.

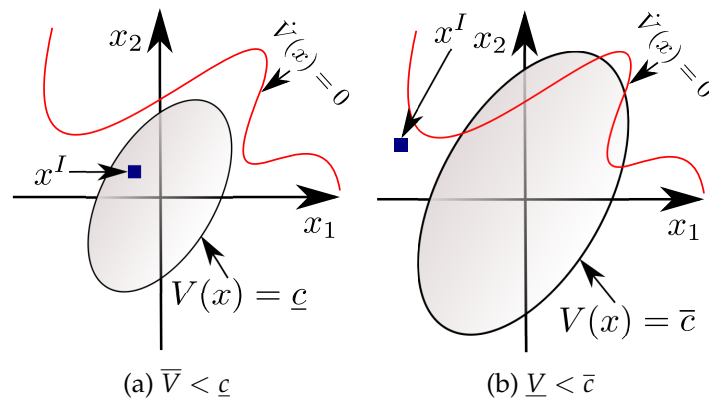


Abbildung 4.2: Darstellung der Fälle  $\bar{V} < \underline{c}$  und  $\underline{V} < \bar{c}$

Aufbauend auf dem Algorithmus aus Kapitel 3.6 sollen nun die Bedingungen herausgearbeitet werden, um ein Intervall als Kandidat für die Lösung zu klassifizieren oder es zu verwerfen. Hierzu wird zuerst angenommen, dass  $V(x)$  quadratisch ist. Für Ljapunowfunktionen höherer Ordnung wird der Algorithmus im weiteren Verlauf des Kapitels angepasst werden.

Für das zu klassifizierende Intervall  $x^I$  werden mithilfe der Gleichungen (4.1) und (4.2) Schranken für die Ljapunowfunktion und ihre zeitliche Ableitung auf  $x^I$  bestimmt.

Gilt  $\underline{V} > \bar{c}$  oder  $\bar{V} < \underline{c}$ , folgt daraus, dass das Intervall weder die Lösung  $x^*$  beinhalten kann noch dass die Schranken für  $c^*$  verbessert werden können. Aus diesem Grund kann das Intervall verworfen werden. In Abbildung 4.2 sind beispielhaft 2 Intervalle gezeigt, welche in diesem Schritt verworfen werden würden.

Gilt  $\underline{\dot{V}} > 0$ , bedeutet dies, dass das Intervall  $x^I$  vollständig im Gebiet

$$\{x \in \mathbb{R}^n \mid \dot{V}(x) > 0\}$$

liegt, also die Lösung  $x^*$  nicht beinhaltet. In Abbildung 4.3a ist dies für ein Beispielintervall illustriert. Bevor das Intervall verworfen werden kann, kann  $V(x)$  am Mittelpunkt  $\check{x}$  mit

$$V_m = V(\check{x})$$

ausgewertet werden. Gilt  $V_m < \bar{c}$ , so kann die obere Schranke für  $c^*$  mit

$$\bar{c} = V_m$$

verbessert werden. Wenn das Intervall nicht im vorherigen Schritt verworfen



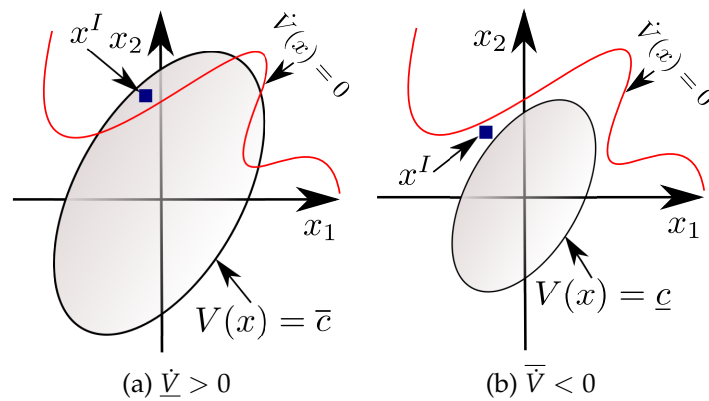


Abbildung 4.3: Darstellung der Fälle  $\dot{V} > 0$  und  $\bar{V} < 0$

wurde, muss überprüft werden, ob  $\bar{V} < 0$  gilt, wie es in Abbildung 4.3b illustriert wird. In diesem Fall kann das Intervall die Lösung  $x^*$  nicht beinhalten und kann daher verworfen werden. Eine Aktualisierung der Schranken kann daher nicht durchgeführt werden.

Wurde das Intervall nicht verworfen, gilt  $0 \in \dot{V}^I(x^I)$ , das bedeutet, dass

$$x^* \in x^I$$

gelten kann. Bevor das Paar  $(\underline{V}, x^I)$  zurück in die Liste sortiert wird, kann eventuell noch die obere Schranke verbessert werden. Gelten

$$\dot{V}(\hat{x}) > 0 \text{ und } V(\hat{x}) < \bar{c} \quad ,$$

kann die obere Schranke für  $c^*$  mit

$$\bar{c} = V_m = V(\hat{x})$$

aktualisiert werden.

Wenn  $V_1$  des ersten Elements in der Liste  $\mathcal{L}$  größer als das bisherige  $\underline{c}$  sowie das initiale  $c_{-init}$  ist, kann die untere Schranke für  $c^*$  mit

$$\underline{c} = V_{-1}$$

aktualisiert werden. Eine Implementierung dieser Klassifizierung in einem Pseudocode findet sich in Anhang A.1.

## 4.2 Erweiterungen der Methode

Als Äquivalent zum in Kapitel 3.7 vorgestellten Gradiententest lässt sich der im vorherigen Abschnitt beschriebene Algorithmus durch die Überprüfung der Lagrangebedingungen erweitern, um frühzeitig Intervalle aus der Liste entfernen zu können. Wie bereits bei der Lösung des Optimierungsproblems mit der Lagrange-multiplikatorenregel erwähnt (vergleiche Kapitel 2), erfüllen alle Optimalpunkte  $x^*$  die Bedingungen

$$\begin{aligned} L(x, \lambda) &= V(x) + \lambda \dot{V}(x) \\ \frac{\partial L}{\partial \lambda}(x^*) &= \frac{\partial V}{\partial x}(x^*) + \lambda \frac{\partial \dot{V}}{\partial x}(x^*) = 0 \\ \frac{\partial L}{\partial x}(x^*, \lambda) &= \dot{V}(x^*) = 0 \end{aligned} \quad (4.4)$$

Da nur Intervalle in die Liste  $\mathcal{L}$  zurücksortiert werden, für welche die Intervall-erweiterung  $\dot{V}^I(x) \ni 0$  erfüllt, müssen nur noch  $n$  Bedingungen

$$\frac{\partial L}{\partial x_i}(x, \lambda) = \frac{\partial V(x)}{\partial x_i} + \lambda \frac{\partial \dot{V}(x)}{\partial x_i} = 0, \quad i = 1, \dots, n \quad (4.5)$$

überprüft werden. Durch Umformulierung von (4.5) erhält man

$$\lambda_i = - \frac{\frac{\partial V}{\partial x_i}}{\frac{\partial \dot{V}}{\partial x_i}} \quad \text{für } \frac{\partial \dot{V}}{\partial x_i} \neq 0 \quad (4.6)$$

Die Idee liegt nun darin, eine neue Intervallvariable  $\lambda^I \in \mathbb{R}$  einzuführen, welche mit den Schranken  $[-\infty, \infty]$  initialisiert wird. Die Einträge der Liste  $\mathcal{L}$  müssen entsprechend um die zum Intervall  $x^{I_j}$  zugehörige neue Variable  $\lambda^{I_j}$  ergänzt werden. Die erweiterte Datenstruktur der Liste ist in Abbildung 4.4 beispielhaft dargestellt.

Gilt  $0 \in V^I(x^I)$ , so können  $n$  temporäre Variablen  $\lambda_i^I$  mit der Gleichung

$$\lambda_i^I = \begin{cases} [-\infty, +\infty] & \text{falls } \frac{\partial \dot{V}^I}{\partial x_i}(x^I) \ni 0 \\ \frac{\partial V^I}{\partial x_i}(x^I) & \\ -\frac{\partial \dot{V}^I}{\partial x_i}(x^I) & \text{sonst} \end{cases} \quad (4.7)$$

| Liste $\mathcal{L}$ |             |             |          |             |                 |
|---------------------|-------------|-------------|----------|-------------|-----------------|
| $\underline{V}_1$   | $x_1^{I_1}$ | $x_2^{I_1}$ | $\cdots$ | $x_n^{I_1}$ | $\lambda^{I_1}$ |
| $\underline{V}_2$   | $x_1^{I_2}$ | $x_2^{I_2}$ | $\cdots$ | $x_n^{I_2}$ | $\lambda^{I_2}$ |
|                     |             |             |          |             |                 |
| $\underline{V}_r$   | $x_1^{I_r}$ | $x_2^{I_r}$ | $\cdots$ | $x_n^{I_r}$ | $\lambda^{I_r}$ |

 Abbildung 4.4: Aufbau der erweiterten Liste  $\mathcal{L}$ 

bestimmt werden. Da  $\lambda$  für alle Gleichungen (4.5) gleich sein muss, gilt es noch, den Schnitt der  $\lambda_i^I$  mit

$$\lambda_{\text{neu}}^I = \lambda_1^I \cap \dots \cap \lambda_n^I \cap \lambda_{\text{alt}}^I$$

zu bestimmen. Hierbei bezeichnet  $\lambda_{\text{alt}}^I$  das Intervall  $\lambda_I$  vor diesem Test. Ist das Schnittintervall  $\lambda_{\text{neu}}^I$  nicht leer, wird es zusammen mit  $(\underline{V}, x^I)$  in der Liste  $\mathcal{L}$  gespeichert. Andernfalls kann  $x^I$  verworfen werden, da es keine Lösung der Optimierungsaufgabe (2.6) enthalten kann.

Ist das zu klassifizierende Intervall klein, kann der Algorithmus mit einem weiteren Test erweitert werden. Hierzu wird nach dem Lagrangetest und vor dem Zurücksortieren in die Liste ein Krawczykschritt (3.14) auf die Intervalle  $x^I$  und  $\lambda^I$  angewandt, mit dem Ziel, entweder das Intervallpaar verwerfen oder verkürzen zu können. Hierzu muss der Krawczykschritt auf das Gleichungssystem

$$G(x, \lambda) = \begin{pmatrix} \frac{\partial V(x)}{\partial x_1} + \lambda \frac{d\dot{V}(x)}{dx_1} \\ \vdots \\ \frac{\partial V(x)}{\partial x_n} + \lambda \frac{d\dot{V}(x)}{dx_n} \\ \dot{V}(x^I) \end{pmatrix} = 0 \quad (4.8)$$

mit  $x \in x^I, \lambda \in \lambda^I$  angewandt werden.

Die Jacobimatrix von  $G(x, \lambda)$  lautet

$$G'(x, \lambda) = \begin{pmatrix} \frac{\partial^2 V(x)}{\partial x_1 \partial x_1} + \lambda \frac{\partial^2 \dot{V}(x)}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 V(x)}{\partial x_1 \partial x_n} + \lambda \frac{\partial^2 \dot{V}(x)}{\partial x_1 \partial x_n} & \frac{\partial \dot{V}(x)}{\partial x_1} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 V(x)}{\partial x_1 \partial x_n} + \lambda \frac{\partial^2 \dot{V}(x)}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 V(x)}{\partial x_n \partial x_n} + \lambda \frac{\partial^2 \dot{V}(x)}{\partial x_n \partial x_n} & \frac{\partial \dot{V}(x)}{\partial x_n} \\ \frac{\partial \dot{V}(x)}{\partial x_1} & \cdots & \frac{\partial \dot{V}(x)}{\partial x_n} & 0 \end{pmatrix}$$

und kann durch

$$G^I(x^I, \lambda^I) = \begin{pmatrix} \frac{\partial^2 V(x^I)}{\partial x_1 \partial x_1} + \lambda^I \frac{\partial^2 \dot{V}(x^I)}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 V(x^I)}{\partial x_1 \partial x_n} + \lambda^I \frac{\partial^2 \dot{V}(x^I)}{\partial x_1 \partial x_n} & \frac{\partial \dot{V}(x^I)}{\partial x_1} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial^2 V(x^I)}{\partial x_1 \partial x_n} + \lambda^I \frac{\partial^2 \dot{V}(x^I)}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 V(x^I)}{\partial x_n \partial x_n} + \lambda^I \frac{\partial^2 \dot{V}(x^I)}{\partial x_n \partial x_n} & \frac{\partial \dot{V}(x^I)}{\partial x_n} \\ \frac{\partial \dot{V}(x^I)}{\partial x_1} & \cdots & \frac{\partial \dot{V}(x^I)}{\partial x_n} & 0 \end{pmatrix}$$

auf  $x^I$  und  $\lambda^I$  eingeschlossen werden. Um die Intervallmatrix  $G'(x^I, \lambda^I)$  durch eine skalare Matrix  $L$  zu approximieren, wird der Mittelpunkt von  $G'(x^I, \lambda^I)$  gewählt, das heißt, die Approximation der Inversen wird durch

$$L = \left( \text{mid} \left( G'(x^I, \lambda^I) \right) \right)^{-1}$$

berechnet. Der Krawczykoperator wird nach Gleichung (3.13) durch

$$K(x^I, \lambda^I) = \begin{pmatrix} \check{x} \\ \check{\lambda} \end{pmatrix} - LG^I \begin{pmatrix} \check{x} \\ \check{\lambda} \end{pmatrix} + (I - L) \begin{pmatrix} x^I - \check{x} \\ \lambda^I - \check{\lambda} \end{pmatrix} \quad (4.9)$$

gebildet. Ist der Schnitt von  $K(x^I, \lambda^I)$  mit  $(x^I, \lambda^I)^T$  leer, kann das Intervall verworfen werden. Gilt

$$K(x^I, \lambda^I) \subset \begin{pmatrix} x^I \\ \lambda^I \end{pmatrix}, \quad (4.10)$$

liegt im Intervall  $x^I$  eine Lösung der Optimierungsaufgabe (2.6). Diese Eigenschaft wird im folgenden Abschnitt noch von Nutzen sein, um die Stabilitätsanalyse mit nichtquadratischen Ljapunowfunktionen durchführen zu können.

## 4.3 Anpassungen für Lyapunowfunktionen höherer Ordnung

In Abschnitt 4.1 wurde von der Voraussetzung ausgegangen, dass  $V(x)$  quadratisch ist. Dies stellt jedoch in vielen Fällen eine erhebliche Einschränkung dar, da hierdurch die Form des ermittelten Einzugsgebiets auf Kreise und Ellipsen beschränkt wird. Durch Anpassung der Aktualisierung der oberen Schranke lässt sich dieses Verfahren jedoch auch auf Lyapunowfunktionen höherer Ordnung anwenden, wodurch auch kompliziertere Einzugsgebiete bestimmt werden können. Solange mit den Niveaumengen von  $V(x)$  ein geschlossenes und zusammenhängendes Gebiet gebildet werden kann, ist es möglich, den in den vorherigen Abschnitten beschriebenen Algorithmus anzuwenden.

Problematisch wird es jedoch, wenn die Menge  $\{x \in \mathbb{R}^n \mid V(x) < c\}$  aus mehreren Teilgebieten besteht. Hierdurch können an Punkten in der Menge  $Y = \{x \in \mathbb{R}^n \mid \dot{V}(x) > 0\}$  Werte für  $V(x)$  auftreten, welche kleiner als der Optimalwert  $c^*$  sind. Aus diesem Grund kann die obere Schranke  $\bar{c}$  für  $c^*$  nicht mehr wie im ursprünglichen Algorithmus durch eine Auswertung an einem Punkt in der Menge  $Y$  aktualisiert werden.

Die neue Herangehensweise ist nun, die obere Schranke der Intervallauswertung  $V^I(x^I) = [\underline{V}, \bar{V}]$  - also  $\bar{V}$  - als potentielle obere Schranke zu nutzen. Dies ist jedoch nur dann erlaubt, wenn bekannt ist, dass im entsprechenden Intervall  $x^I$  ein Optimalpunkt liegt. Verifiziert werden muss dies mit dem im Abschnitt 4.2 beschriebenen Krawczykschritt. Nur wenn Gleichung (4.10) erfüllt ist, ist sichergestellt, dass  $V(x)$  auf dem Intervall  $x^I$  mindestens den Wert von  $c^*$  annimmt.

Eine Teilmenge des Einzugsgebiets  $\Omega_c$  wird nach Beendigung des Algorithmus nicht mehr durch Gleichung (2.2) beschrieben, sondern durch diejenige zusammenhängende Untermenge von

$$\Omega_c^* = \{x \in \mathbb{R}^n \mid V(x) \leq c\}, \quad c > 0 \quad ,$$

welche den Ursprung enthält.

Der angepasste Algorithmus lässt sich auch auf die Stabilitätsbetrachtung mit quadratischen Lyapunowfunktionen anwenden. Da die obere Schranke für  $c^*$  nun deutlich konservativer geschätzt und seltener aktualisiert wird, kann sich die Laufzeit des angepassten Algorithmus erhöhen.

Eine Beschreibung des angepassten Algorithmus in Pseudocode ist in Anhang A.2 zu finden.

## 4.4 Beispiele

Zur Verifikation des in den vorherigen Abschnitten vorgestellten Verfahrens und zur Beurteilung von dessen Effektivität wurden die in der Literatur verwendeten Beispiele mit dem neuen Verfahren nachgerechnet. Hierbei ist der Algorithmus in MATLAB [91] mithilfe der Toolbox INTLAB [78] implementiert und auf einem Standard x86-PC ausgeführt worden. Die hier dargestellten Ergebnisse sind nach der siebten Nachkommastelle nach außen gerundet, das heißt, die untere Schranke ist ab-, die obere Schranke hingegen aufgerundet.

In den Beispielen 4.1 bis 4.14 werden Einzugsgebiete für polynomiale Systeme mithilfe einer quadratischen Ljapunovfunktion bestimmt, während in den Beispielen 4.15 bis 4.20 Ljapunovfunktionen höherer Ordnung verwendet werden. In den Beispielen 4.21 bis 4.25 werden Einzugsgebiete für nichtpolynomiale Systeme ermittelt.

### Beispiel 4.1

Es soll das System

$$\begin{aligned}\dot{x}_1 &= -x_1 \\ \dot{x}_2 &= -x_2 + x_1^2 x_2\end{aligned}$$

aus [35] mit der Ljapunowfunktion  $V(x) = x_1^2 + x_2^2$  betrachtet werden. Mithilfe der hier vorgestellten Methode konnten die Schranken

$$\begin{aligned}\underline{c} &= 3.9999999 \\ \bar{c} &= 4.0000000\end{aligned}$$

für  $c^*$  ermittelt werden, welche konsistent mit der Lösung aus [35] sind. Wie Abbildung 4.5 ersichtlich ist, berühren sich die Linien  $V(x) = \underline{c}$  und  $\dot{V}(x) = 0$  nahezu.

### Beispiel 4.2

Betrachtet werden soll das System aus [35] mit der Zustandsraumdarstellung

$$\begin{aligned}\dot{x}_1 &= -x_1 + x_2^2 \\ \dot{x}_2 &= -2x_2\end{aligned}$$

mit der Ljapunowfunktion  $V(x) = 0.5 (x_1^2 + x_2^2)$ . In [35] ist als untere Schranke  $\underline{c} = 8.81831$  angegeben, während mithilfe der hier vorgestellten Methode die Schranken

$$\begin{aligned} c &= 8.8182996 \\ \underline{c} &= 8.8182998 \end{aligned}$$

für  $c^*$  bestimmt wurden. Die hier ermittelten Schranken wurden jedoch sowohl graphisch (vergleiche Abbildung 4.6) als auch mit der Methode von [80] verifiziert, sodass hier ein Fehler bei der Ermittlung des Ergebnisses in [35] angenommen werden muss. Es ist davon auszugehen, dass beim verwendeten LMI-Löser numerische Probleme auftraten.

### Beispiel 4.3

Das in [35] vorgestellte Beispielsystem

$$\begin{aligned} \dot{x}_1 &= -2x_1 + x_1x_2 \\ \dot{x}_2 &= -x_2 + x_1x_2 \end{aligned}$$

soll mithilfe der Ljapunowfunktion  $V(x) = x_1^2 + x_2^2$  untersucht werden. Die ermittelte Einschließung

$$\begin{aligned} c &= 4.0954839 \\ \underline{c} &= 4.0954841 \end{aligned}$$

ist konsistent mit dem Ergebnis aus [35]. Das resultierende Einzugsgebiet kann Abbildung 4.7 entnommen werden.

### Beispiel 4.4

Das System aus [35] mit der Zustandsraumdarstellung

$$\begin{aligned} \dot{x}_1 &= -x_1 + 2x_1^2x_2 \\ \dot{x}_2 &= -x_2 \end{aligned}$$

soll mit der Ljapunowfunktion  $V(x) = 0.33x_1^2 + 0.498x_1x_2 + 0.376x_2^2$  untersucht werden. Mithilfe der ermittelten Schranken

$$\begin{aligned} c &= 0.9961677 \\ \underline{c} &= 0.9961679 \end{aligned}$$

kann das Einzugsgebiet  $\Omega_c = \{x \in \mathbb{R}^2 \mid V(x) < 0.9961677\}$  bestimmt werden. Aus Abbildung 4.8 kann entnommen werden, dass  $\Omega_c$  die Linie  $\dot{V}(x)$  nahezu berührt.

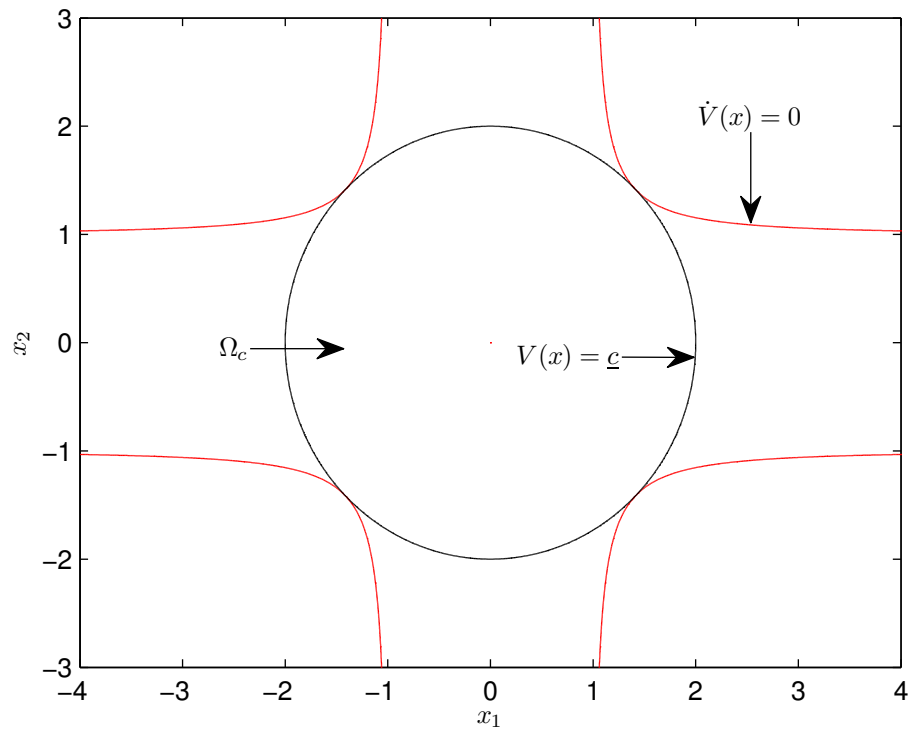


Abbildung 4.5: Ermitteltes Einzugsgebiet für Beispiel 4.1

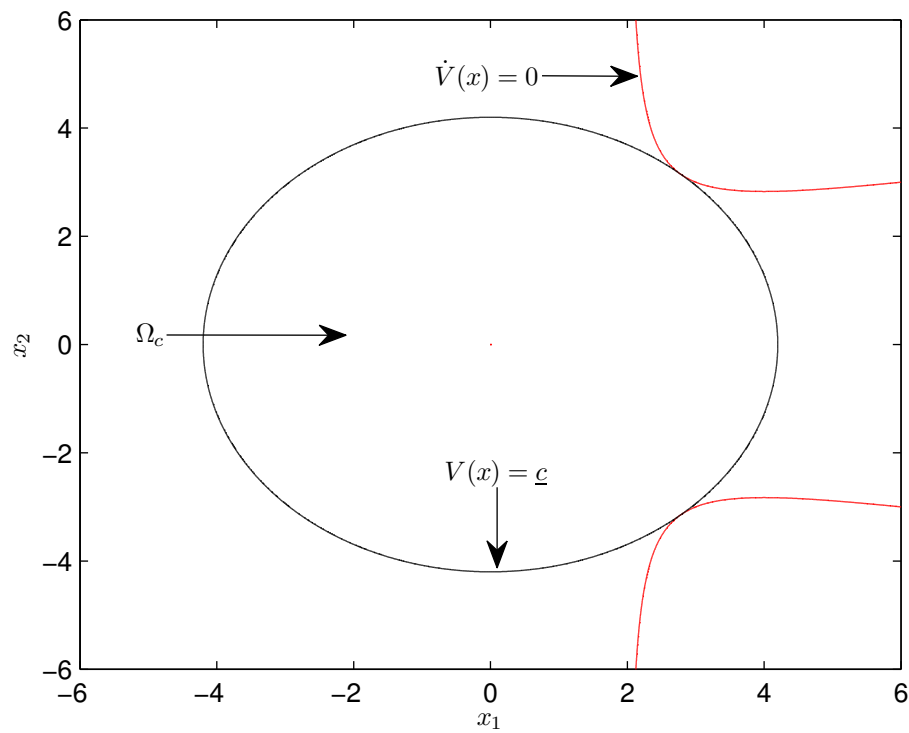


Abbildung 4.6: Ermitteltes Einzugsgebiet für Beispiel 4.2



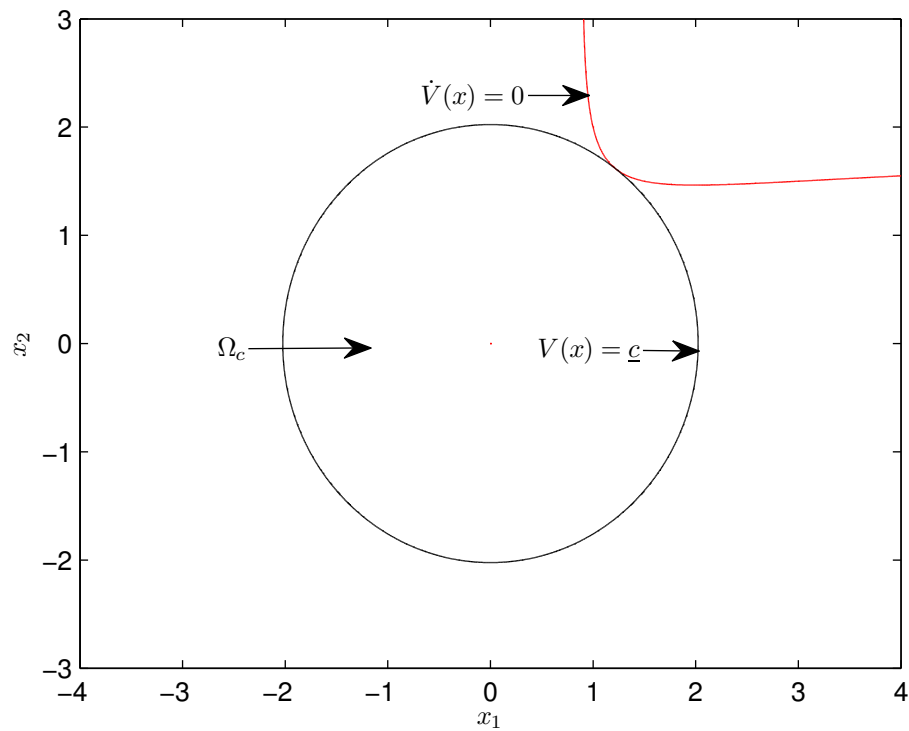


Abbildung 4.7: Ermitteltes Einzugsgebiet für Beispiel 4.3

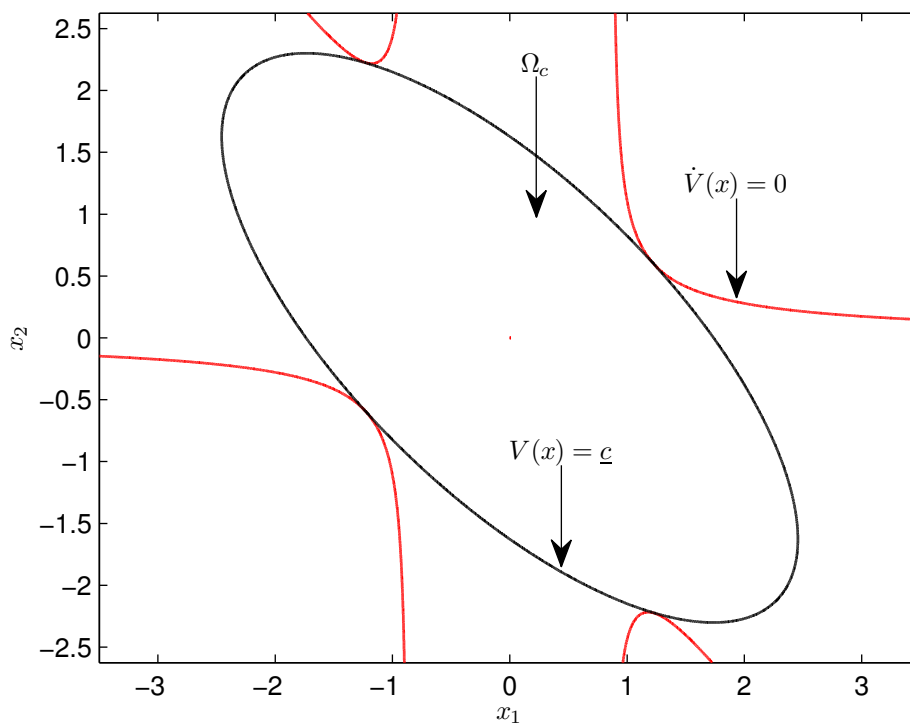


Abbildung 4.8: Ermitteltes Einzugsgebiet für Beispiel 4.4

**Beispiel 4.5**

Das in [35] vorgestellte Beispielsystem

$$\begin{aligned}\dot{x}_1 &= -x_1 + x_2 \\ \dot{x}_2 &= 0.1x_1 - x_2 - x_1^2 - 0.1x_1^3\end{aligned}$$

soll mithilfe der Ljapunowfunktion  $V(x) = x_1^2 + x_2^2$  untersucht werden. Die ermittelte Einschließung

$$\begin{aligned}c &= 7.1114710 \\ \underline{c} &= 7.1114711\end{aligned}$$

weist auf ein kleineres  $c^*$  als die in [35] angegebene untere Schranke  $\underline{c} = 7.112889$  hin. Wie in Beispiel 4.2 wurde das ermittelte Ergebnis sowohl graphisch (siehe Abbildung 4.9) als mit der Methode aus [80] verifiziert, sodass hier ein Problem mit dem verwendeten LMI-Löser beim Verfahren von [35] vermutet werden muss. Man kann in Abbildung 4.9 erkennen, dass die ermittelte untere Schranke für  $c^*$  ein Einzugsgebiet liefert, welches die Linie  $\dot{V}(x) = 0$  beinahe berührt.

**Beispiel 4.6**

Das in [35] vorgestellte Beispielsystem

$$\begin{aligned}\dot{x}_1 &= -x_1 + x_2x_3^2 \\ \dot{x}_2 &= -x_2 + x_1x_2 \\ \dot{x}_3 &= -x_3\end{aligned}$$

soll mithilfe der quadratischen Ljapunowfunktion  $V(x) = x_1^2 + x_2^2 + x_3^2$  untersucht werden. Die ermittelte Einschließung

$$\begin{aligned}c &= 4.9187584 \\ \underline{c} &= 4.9187585\end{aligned}$$

bestätigt die in [35] gefundene untere Schranke  $\underline{c} = 4.9187584$ . Das resultierende Einzugsgebiet kann Abbildung 4.10 entnommen werden.

**Beispiel 4.7**

Für das nichtlineare System aus [36]

$$\dot{x} = \left( (x - \delta)^T (x - \delta) - 1 \right) x$$

mit dem Zustandsvektor

$$x = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{pmatrix}^T \in \mathbb{R}^5$$

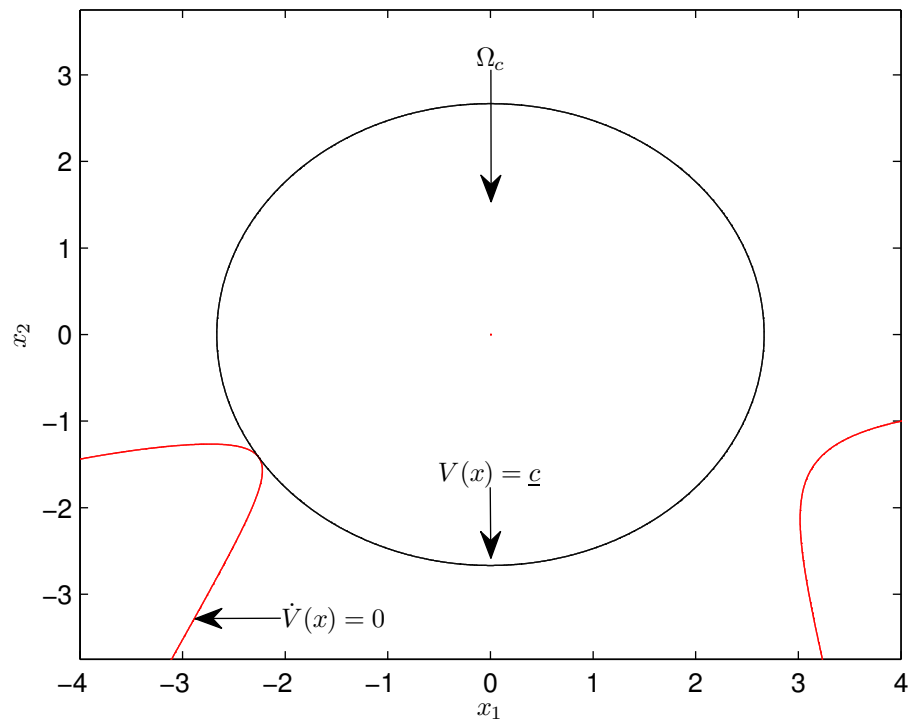


Abbildung 4.9: Ermitteltes Einzugsgebiet für Beispiel 4.5

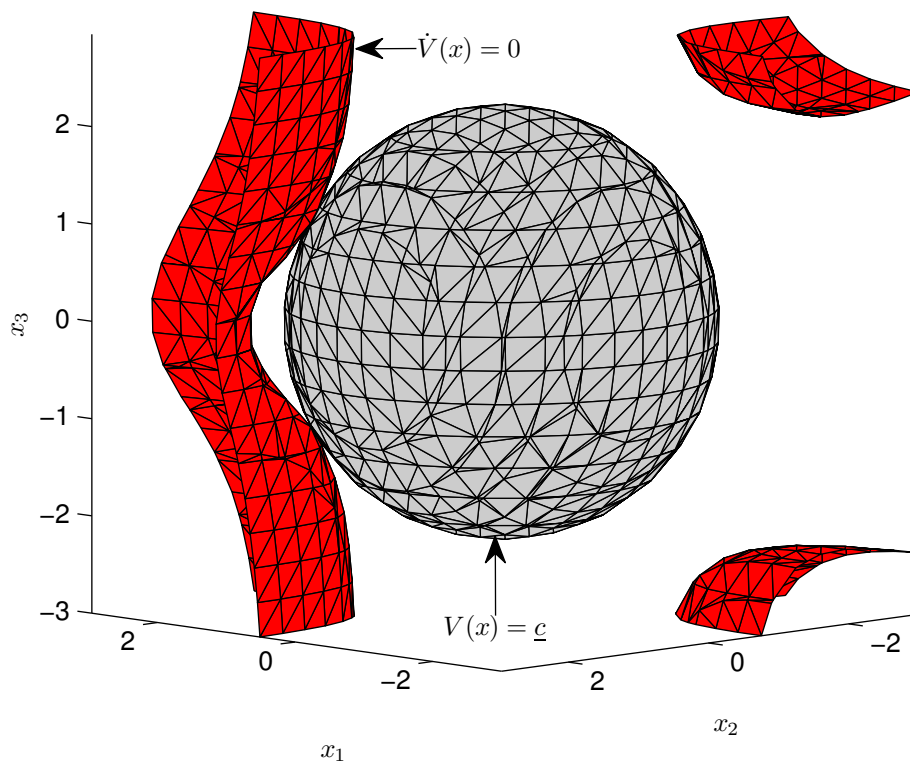


Abbildung 4.10: Ermitteltes Einzugsgebiet für Beispiel 4.6

und

$$\delta = (0.2 \ 0.2 \ 0.2 \ 0.2 \ 0.2)^T \in \mathbb{R}^5$$

soll ein Einzugsgebiet mithilfe der Ljapunowfunktion

$$V(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2$$

bestimmt werden. Mit den Schranken

$$\begin{aligned} c &= 4.9187584 \\ \underline{c} &= 4.9187585 \end{aligned}$$

kann  $c^*$  eng eingeschlossen werden.

#### Beispiel 4.8

Ein Van-der-Pol-Oszillator [50] kann durch die Differenzialgleichungen

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 - x_2(1 - x_1^2) \end{aligned}$$

beschrieben werden. Mit der in [13] gegebenen Ljapunowfunktion

$$V(x) = x_1^2 + x_1x_2 + x_2^2$$

soll nun ein Einzugsgebiet bestimmt werden. Mit der Einschließung

$$\begin{aligned} c &= 0.9152361 \\ \underline{c} &= 0.9152362 \end{aligned}$$

kann das Einzugsgebiet

$$\Omega_c = \left\{ x \in \mathbb{R}^2 \mid V(x) \leq 0.9152361 \right\}$$

bestimmt werden. Die ermittelten Schranken sind hierbei konsistent mit der in [13] ermittelten unteren Schranke  $\underline{c} = 0.915$ . Das resultierende Einzugsgebiet kann Abbildung 4.11 entnommen werden.

#### Beispiel 4.9

In [13] ist ein Räuber-Beute-Modell durch die Systemdifferenzialgleichungen

$$\begin{aligned} \dot{x}_1 &= -3x_1 + 4x_1^2 - 0.5x_1x_2 - x_1^3 \\ \dot{x}_2 &= -2.1x_2 + x_1x_2 \end{aligned}$$

gegeben. Dieses System besitzt die vier Ruhelagen

$$x^{R_1} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad x^{R_2} = \begin{pmatrix} 2.1 \\ 1.98 \end{pmatrix}, \quad x^{R_3} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{und} \quad x^{R_4} = \begin{pmatrix} 3 \\ 0 \end{pmatrix},$$

von welchen die ersten beiden asymptotisch stabil sind.

Die Stabilitätsbetrachtung der Ruhelage  $x^{R_1}$  mit der Ljapunowfunktion

$$V(x) = 4x_1^2 + x_1x_2 + 2x_2^2$$

führt zu der Einschließung

$$\begin{aligned} c &= 3.3762171 \\ \underline{c} &= 3.3762172 \quad , \end{aligned}$$

welche konsistent mit der unteren Schranke aus [13]  $\underline{c} = 3.376$  ist. Wie Abbildung 4.12 entnommen werden kann, berührt die Linie  $V(x) = \underline{c}$  nahezu die Linie  $\dot{V}(x) = 0$ .

#### Beispiel 4.10

Betrachtet werden soll das Einzugsgebiet der Ruhelage  $x^{R_2}$  des Systems aus Beispiel 4.10. Transformiert man das System so, dass die Ruhelage  $x^{R_2}$  im Ursprung liegt, erhält man die Zustandsraumdarstellung

$$\begin{aligned} \dot{x}_1 &= -0.42x_1 - 2.3x_1^2 - 0.5x_1x_2 - 1.05x_2 - x_1^3 \\ \dot{x}_2 &= x_1x_2 + 1.98x_1 \end{aligned}$$

Mithilfe der Ljapunowfunktion

$$V(x) = 4x_1^2 + x_1x_2 + 2x_2^2$$

konnten die Schranken

$$\begin{aligned} c &= 0.0216741 \\ \underline{c} &= 0.0216742 \end{aligned}$$

eruiert werden. Die in [13] angegebene untere Schranke  $\underline{c} = 0.022$  ist laut persönlicher Mitteilung des Autors gerundet. Abbildung 4.13 kann entnommen, dass mit der neuen Methode für die gegebene Ljapunowfunktion nahezu das optimale Einzugsgebiet bestimmt werden.

#### Beispiel 4.11

In [26] wurde das Einzugsgebiet des Systems

$$\begin{aligned} \dot{x}_1 &= -x_1 + 2x_2^2 \\ \dot{x}_2 &= x_1^2 - x_2 + x_3^2 \\ \dot{x}_3 &= -x_1^2 - x_3 \end{aligned}$$

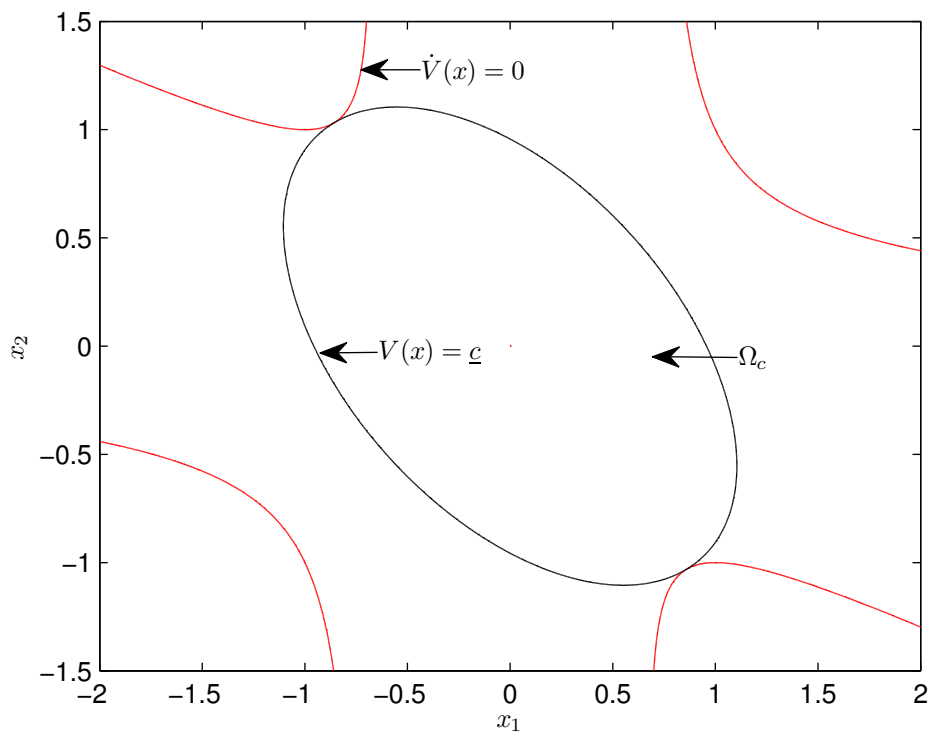


Abbildung 4.11: Ermitteltes Einzugsgebiet für Beispiel 4.8

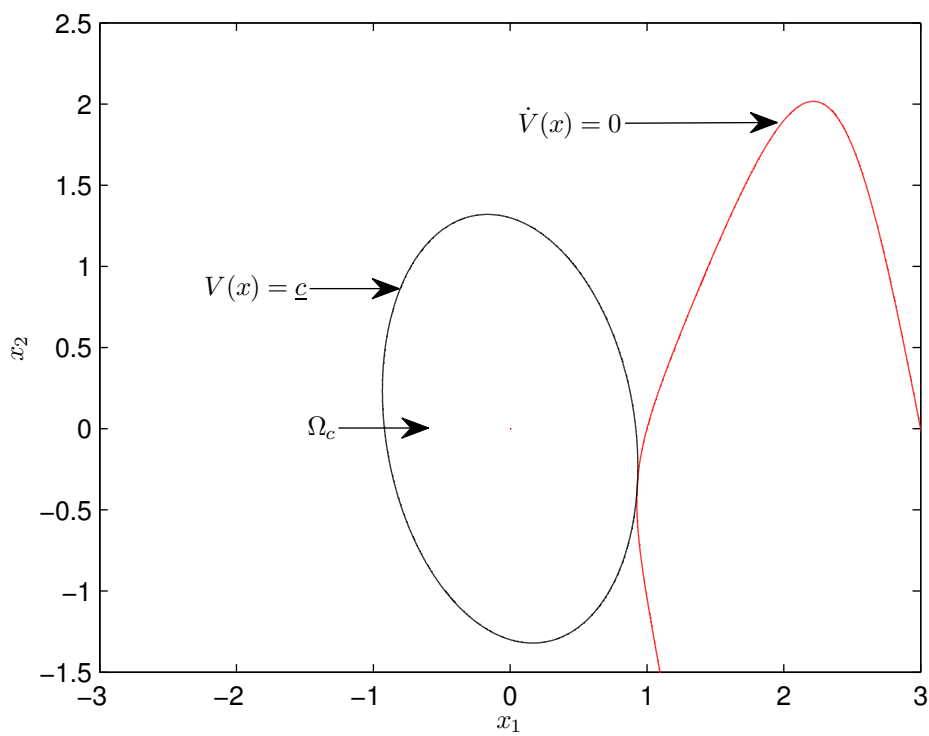


Abbildung 4.12: Ermitteltes Einzugsgebiet für Beispiel 4.9

mit der Ljapunowfunktion  $V(x) = 0.5 (x_1^2 + x_2^2 + x_3^2)$  betrachtet und das Einzugsgebiet

$$\Omega_c = \left\{ x \in \mathbb{R}^3 \mid V(x) \leq 0.395 \right\}$$

bestimmt. Mithilfe der bestimmten Schranken

$$\begin{aligned} c &= 0.3950264 \\ \underline{c} &= 0.3950265 \end{aligned}$$

kann dieses Einzugsgebiet bestätigt beziehungsweise auf das in Abbildung 4.14 dargestellte Einzugsgebiet erweitert werden.

#### Beispiel 4.12

In [26] wurde das System

$$\begin{aligned} \dot{x}_1 &= -x_1 - x_1 x_2 \\ \dot{x}_2 &= -x_2 - x_1^3 \end{aligned}$$

vorgelegt, für welches mit der Ljapunowfunktion  $V(x) = x_1^2 + 2x_1 x_2 + 2x_2^2$  ein Einzugsgebiet bestimmt werden soll. Mit der Methode aus [26] war dies nicht möglich, mit der neuen Methode kann die Einschließung

$$\begin{aligned} c &= 0.5455994 \\ \underline{c} &= 0.5455995 \end{aligned}$$

ermittelt werden.

#### Beispiel 4.13

Für das nichtlineare System mit der Zustandsraumdarstellung

$$\begin{aligned} \dot{x}_1 &= -x_1 + x_1^2 x_2 \\ \dot{x}_2 &= -2x_2 \end{aligned}$$

wird in [58] die Ljapunowfunktion  $V(x) = \left(\frac{x_1}{2.59}\right)^2 + \left(\frac{x_1}{1.32}\right)^2$  angegeben, um das Einzugsgebiet

$$\Omega_c = \{ x \in \mathbb{R}^n \mid V(x) < 1 \}$$

zu bestimmen. Mit dem hier vorgestellten Verfahren konnten die Schranken

$$\begin{aligned} c &= 1.0952614 \\ \underline{c} &= 1.0952616 \end{aligned}$$

eruiert werden, wodurch sich das Einzugsgebiet

$$\Omega_c = \{ x \in \mathbb{R}^n \mid V(x) < 1.0952614 \}$$

ergibt. Wie in Abbildung 4.16 ersichtlich, berührt die Linie  $V(x) = \underline{c}$  die Linie  $\dot{V}(x) = 0$  nicht.

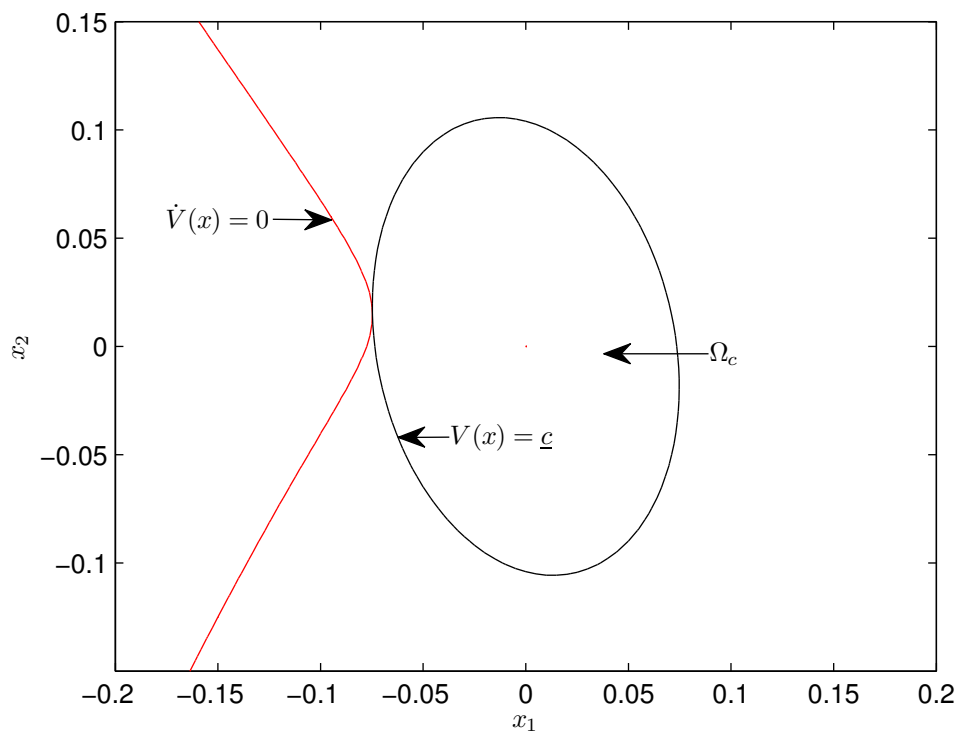


Abbildung 4.13: Ermitteltes Einzugsgebiet für Beispiel 4.10

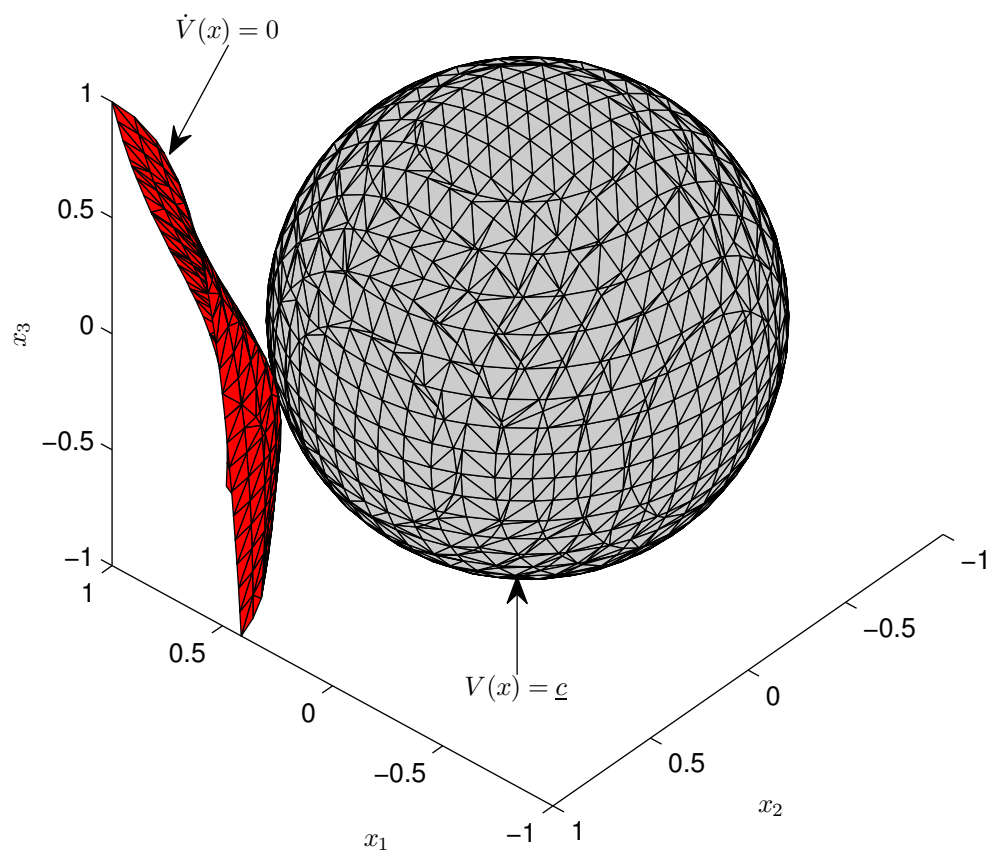


Abbildung 4.14: Ermitteltes Einzugsgebiet für Beispiel 4.11



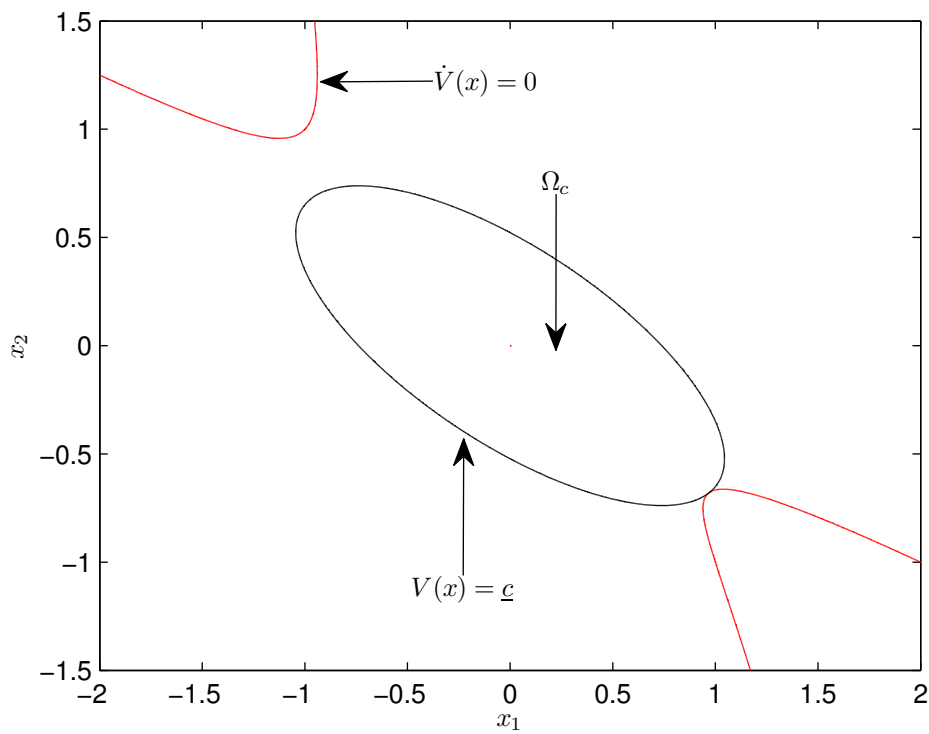


Abbildung 4.15: Ermitteltes Einzugsgebiet für Beispiel 4.12

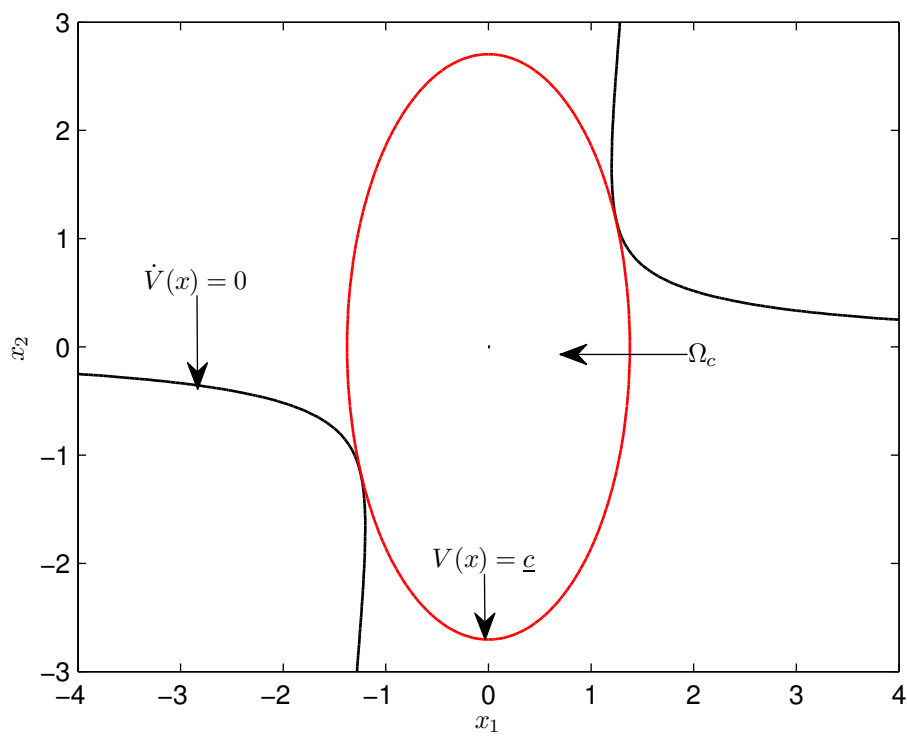


Abbildung 4.16: Ermitteltes Einzugsgebiet für Beispiel 4.13

**Beispiel 4.14**

Das sogenannte SIR-Modell beschreibt in der mathematischen Epidemiologie ein Modell zur Ausbreitung von ansteckenden Krankheiten. In [57] wird das SIR-Modell mit

$$\begin{aligned}\dot{S} &= A - \beta SI - dS + cI + \delta R \\ \dot{I} &= \beta SI - rI - dI - \alpha I - cI \\ \dot{R} &= rI - dR - \delta R\end{aligned}$$

betrachtet, wobei  $S$  die Anzahl der anfälligen (aber gesunden),  $I$  die Anzahl der der reversibel erkrankten und ansteckenden und  $R$  die nach Infektion wieder gesunden Individuen beschreibt. Die Gesamtpopulation ist  $N = S + I + R$ . Die Konstanten  $A, c, d, r, \alpha, \beta$  und  $\delta$  sind hierbei positiv.

Mit  $A = 4, c = d = r = \alpha = \beta = \delta = 0.5$  und der Verschiebung der Ruhelage  $(4 \ 1.6 \ 0.8)^T$  in den Ursprung folgt die Zustandsraumdarstellung

$$\begin{aligned}\dot{x}_1 &= -1.3x_1 - 1.5x_2 + 0.5x_3 - 0.5x_1x_2 \\ \dot{x}_2 &= 0.8x_1 + 0.5x_1x_2 \\ \dot{x}_3 &= 0.5x_2 - x_3 \quad .\end{aligned}$$

Bei der Analyse mit der Ljapunowfunktion

$$V(x) = \frac{299}{475}x_1^2 + \frac{6457}{3800}x_2^2 + \frac{286}{475}x_3^2 + \frac{378}{475}x_1x_2 + \frac{184}{475}x_2x_3 + \frac{194}{475}x_1x_3$$

konnten die Schranken für  $c^*$  zu

$$\begin{aligned}c &= 2.6094420 \\ \underline{c} &= 2.6094421\end{aligned}$$

bestimmt werden, womit welche konsistent mit der in [57] ermittelten unteren Schranke  $\underline{c} = 2.6094$  sind. Das entsprechende Einzugsgebiet kann Abbildung 4.17 entnommen werden.

**Beispiel 4.15**

Das System aus [16, 35, 90] mit der Zustandsraumdarstellung

$$\begin{aligned}\dot{x}_1 &= -x_1 + 2x_1^2x_2 \\ \dot{x}_2 &= -x_2\end{aligned}$$

soll mit der Ljapunowfunktion vom Grad  $m = 4$

$$\begin{aligned}V(x) &= 0.2991x_1^2 + 0.06027x_1^3x_2 + 0.4513x_1x_2 \\ &\quad + 0.003036x_1^4x_2^2 + 0.04548x_1^2x_2^2 + 0.34079x_2^2\end{aligned}$$

betrachtet werden. Während in [35] die untere Schranke zu  $\underline{c} = 0.450$  bestimmt wurde, konnten mit dem neuen Verfahren die Schranken

$$\begin{aligned} c &= 0.4647959 \\ \underline{c} &= 0.4647960 \end{aligned}$$

ermittelt werden. Das Einzugsgebiet ist nun das zusammenhängende Gebiet von

$$\Omega_c^* = \left\{ x \in \mathbb{R}^2 \mid V(x) \leq 0.4647959 \right\} ,$$

welches den Ursprung enthält. Dieses kann auch Abbildung 4.18 entnommen werden.

#### Beispiel 4.16

Für das System

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -2x_1 - 3x_2 + x_1^2 x_2 \end{aligned}$$

wurde in [35] mit der Ljapunowfunktion

$$V(x) = 2.333x_1^2 + 1.353x_1x_2 + 0.1152x_1^3x_2 + 0.4306x_2^2 + 0.092x_1^2x_2^2 + 0.00509x_1^4x_2^2$$

das Gebiet

$$\Omega_c^* = \left\{ x \in \mathbb{R}^2 \mid V(x) \leq 6.680 \right\}$$

als Untermenge des exakten Einzugsgebiets bestimmt. Mithilfe der in dieser Arbeit vorgestellten Methode konnte die Einschließung

$$\begin{aligned} c &= 7.9296479 \\ \underline{c} &= 7.9296480 \end{aligned}$$

bestimmt werden, was zum in Abbildung 4.19 dargestellten Gebiet

$$\Omega_c^* = \left\{ x \in \mathbb{R}^2 \mid V(x) \leq 7.9296479 \right\}$$

führt.

#### Beispiel 4.17

Das Einzugsgebiet der Ruhelage im Ursprung des Systems

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -2x_1 - 3x_2 + x_1^2 x_2 \end{aligned}$$

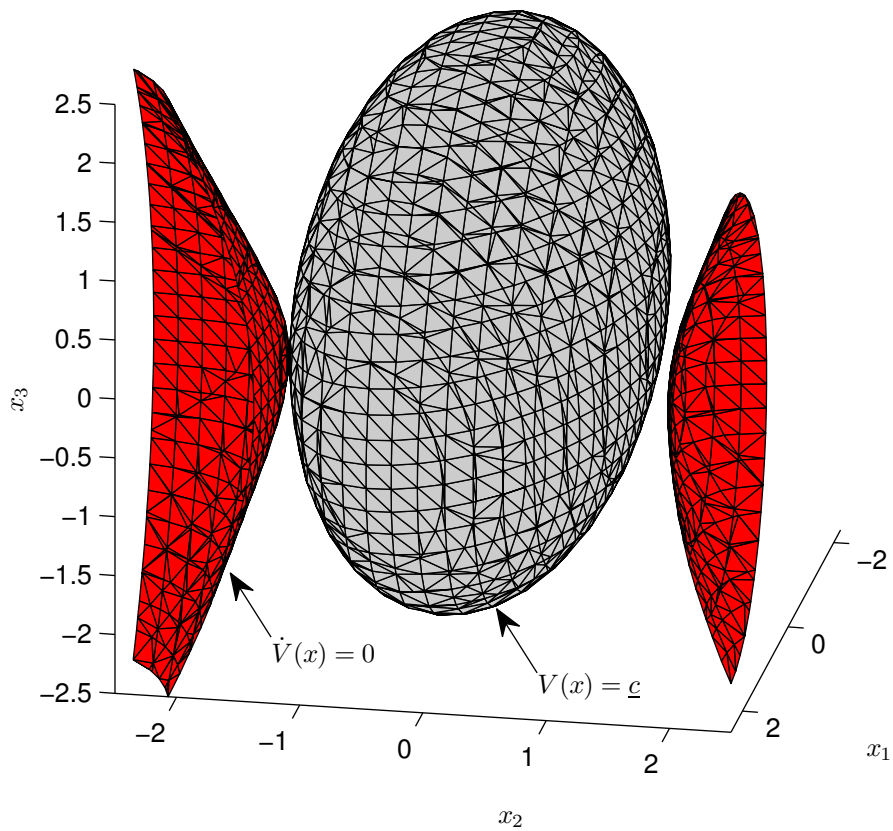


Abbildung 4.17: Ermitteltes Einzugsgebiet für Beispiel 4.14

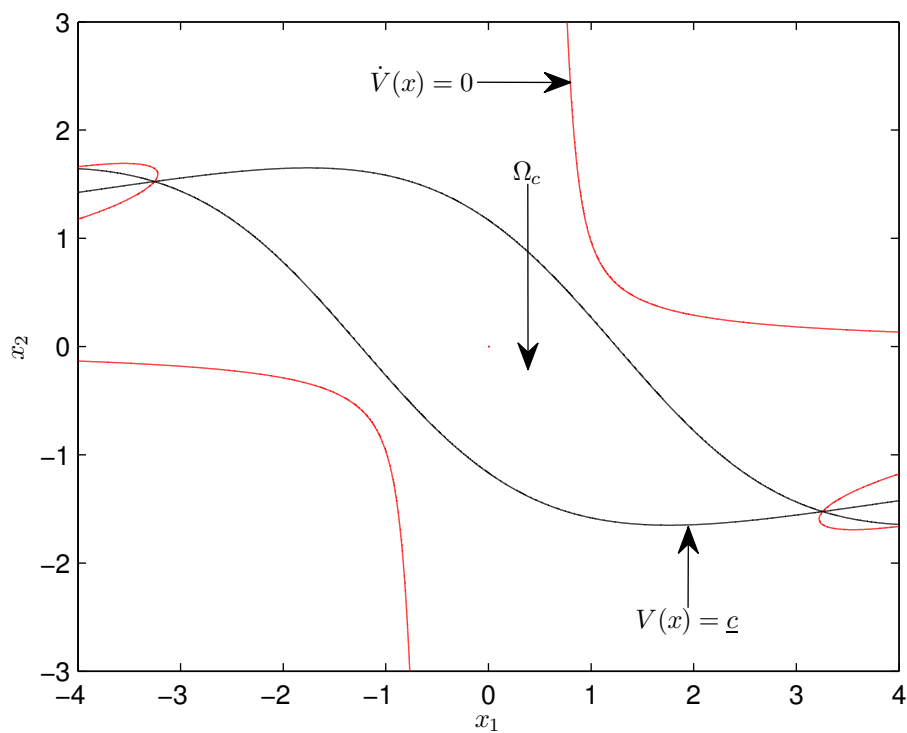


Abbildung 4.18: Ermitteltes Einzugsgebiet für Beispiel 4.15

soll mit der Ljapunowfunktion

$$\begin{aligned} V(x) = & 1.398x_1x_2 + 2.173 \times 10^{-3}x_1^3x_2 - 8.5276 \times 10^{-3}x_1x_2^3 \\ & -1.326 \times 10^{-5}x_1^3x_2^3 + 3.816x_1^2 + .011785x_1^4 + 0.3774x_2^2 \\ & +4.7097 \times 10^{-3}x_2^4 + 9.0992 \times 10^{-6}x_1^6 + 1.46952 \times 10^{-5}x_2^6 \end{aligned}$$

analysiert werden. In [35] wurde die untere Schranke für  $c^*$  zu

$$5.848 \leq c^*$$

bestimmt. Diese untere Schranke kann mit der neuen Methode und der Einschließung

$$\begin{aligned} c &= 7.5608322 \\ \underline{c} &= 7.5608323 \end{aligned}$$

verbessert werden. Hieraus folgt das in Abbildung 4.20 dargestellte Einzugsgebiet.

#### Beispiel 4.18

Für das System

$$\begin{aligned} \dot{x}_1 &= -2x_1 + x_2 + x_1^3 + x_2^5 \\ \dot{x}_2 &= -x_1 - x_2 - x_1^2x_2^3 \end{aligned}$$

soll mit der Ljapunowfunktion

$$\begin{aligned} V(x) = & 1.3637x_2^2 + 0.31873x_1^2 + 0.14717x_1x_2 + 0.0989x_1^3x_2 + 0.11578x_1x_2^5 \\ & +0.40147x_1^2x_2^4 + 0.004794x_1^3x_2^3 + 0.029774x_1^4x_2^6 + 0.011599x_1^2x_2^8 \\ & +0.01159x_1^5x_2^3 + 0.02296x_1^3x_2^5 + 0.011481x_2^{10} + 0.011481x_1^6 \\ & +0.115782x_1^4 + 0.098905x_2^6 \end{aligned}$$

eine garantierte Untermenge des Einzugsgebiets bestimmt werden. Mit der Methode aus [35] konnte die untere Schranke für  $\underline{c} = 0.992$  ermittelt werden, welche mit der neuen Methode auf

$$\begin{aligned} c &= 1.2549328 \\ \underline{c} &= 1.2549329 \end{aligned}$$

verbessert werden konnte. Das ermittelte Einzugsgebiet kann der Abbildung 4.21 entnommen werden.

#### Beispiel 4.19

Das System

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 - x_2(1 - x_1^2) \end{aligned}$$

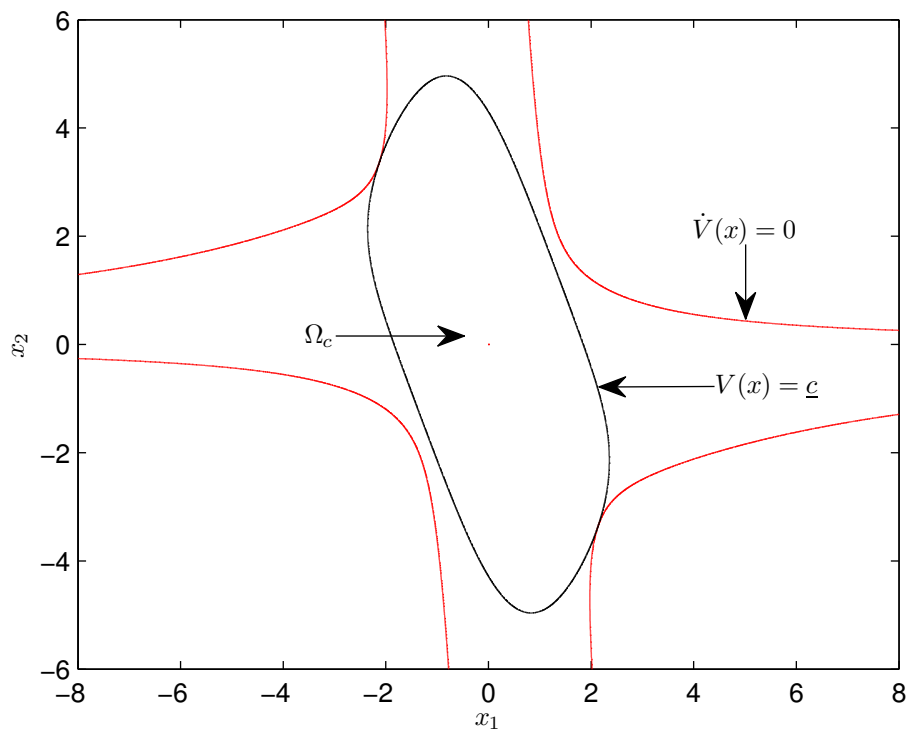


Abbildung 4.19: Ermitteltes Einzugsgebiet für Beispiel 4.16

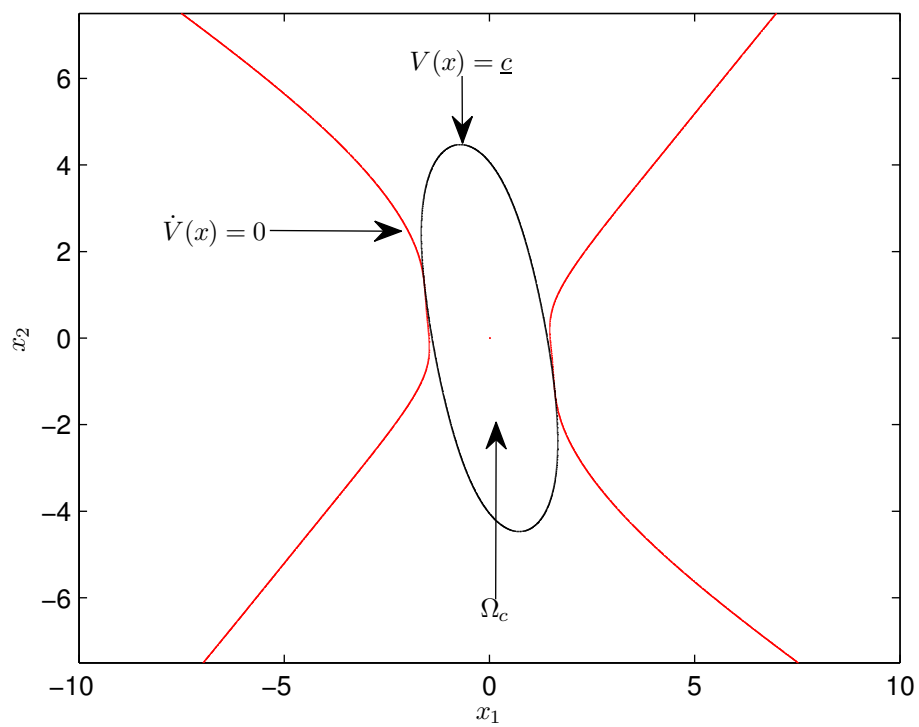


Abbildung 4.20: Ermitteltes Einzugsgebiet für Beispiel 4.17

wurde bereits in Beispiel 4.8 mit einer quadratischen Ljapunowfunktion betrachtet. Nun soll die folgende Ljapunowfunktion vom Grad  $m = 4$  verwendet werden:

$$V(x) = x_1^2 + x_1x_2 + x_2^2 + x_1^4 + x_1^2x_2^2 + x_2^4$$

In [13] konnte eine untere Schranke  $\underline{c} = 3.047$  ermittelt werden, mit dem hier vorgestellten Verfahren kann dieses Ergebnis mithilfe der berechneten Schranken

$$\begin{aligned} c &= 3.0470403 \\ \underline{c} &= 3.0470404 \end{aligned}$$

verifiziert werden. Das resultierende Einzugsgebiet kann Abbildung 4.22 entnommen werden.

#### Beispiel 4.20

Für das System mit der Zustandsraumdarstellung

$$\begin{aligned} \dot{x}_1 &= x_2^4 - 2x_1 \\ \dot{x}_2 &= 3x_1x_2^3 - x_2 \end{aligned}$$

aus [26] soll mit der Ljapunowfunktion  $V(x) = x_1^4 + x_2^2$  ein Einzugsgebiet bestimmt werden. Während für die Methode [26] der Aufwand zu groß ist, kann mit dem hier vorgestellten Verfahren die Einschließung

$$\begin{aligned} c &= 0.8211190 \\ \underline{c} &= 0.8211191 \end{aligned}$$

für  $c^*$  bestimmt werden. Hieraus folgt das in Abbildung 4.23 dargestellte Einzugsgebiet.

#### Beispiel 4.21

In [11] wird

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_2 - \sin(x_1) \end{aligned}$$

als Beschreibung der Systemdynamik eines einfachen Pendels angenommen. Mit der Ljapunowfunktion  $V(x) = 4x_1^2 + 2x_1x_2 + 3x_2^2$  können mit der in dieser Arbeit präsentierten Methode die Schranken

$$\begin{aligned} c &= 23.0071867 \\ \underline{c} &= 23.0071868 \end{aligned}$$

bestimmt werden, womit die untere Schranke aus [11],  $\underline{c} = 22.94$ , verbessert wird. Wie in Abbildung 4.24 berührt die Linie  $V(x) = \underline{c}$  nahezu die Linie  $\dot{V}(x) = 0$ .

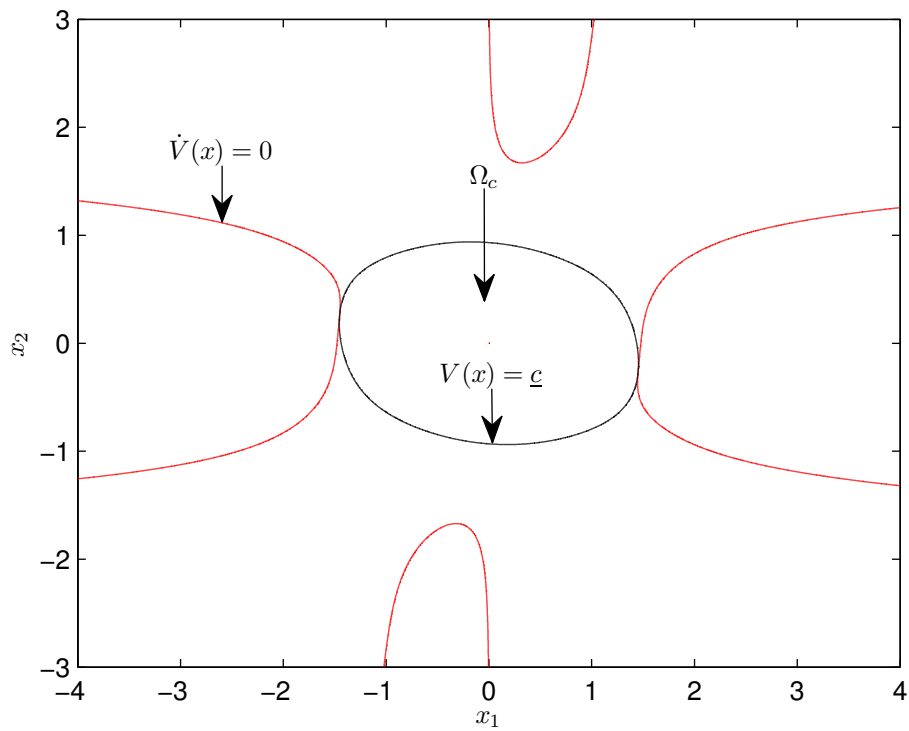


Abbildung 4.21: Ermitteltes Einzugsgebiet für Beispiel 4.18

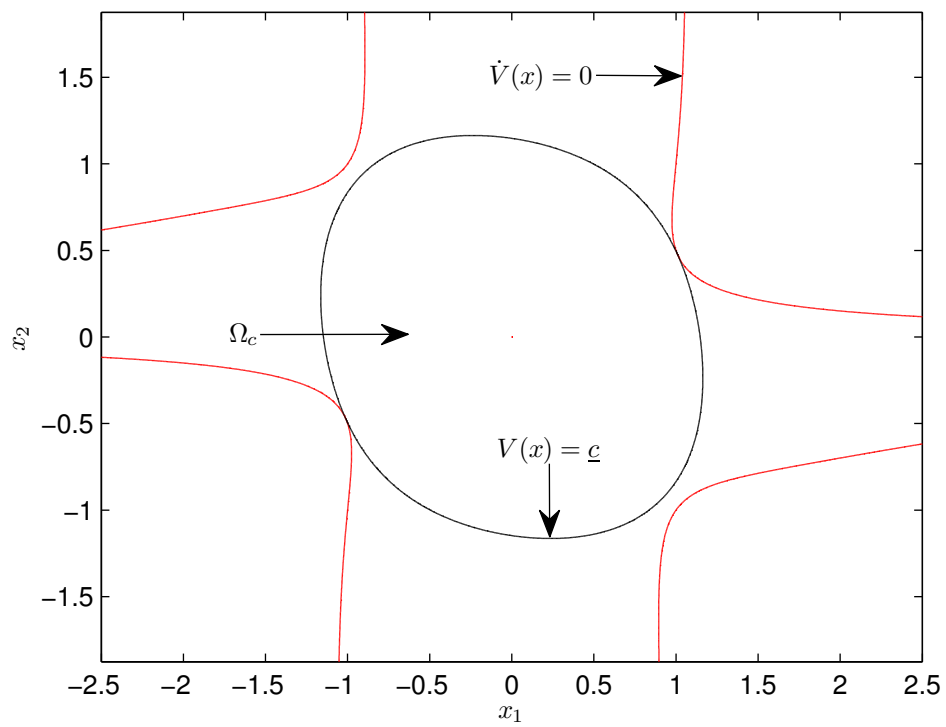


Abbildung 4.22: Ermitteltes Einzugsgebiet für Beispiel 4.19



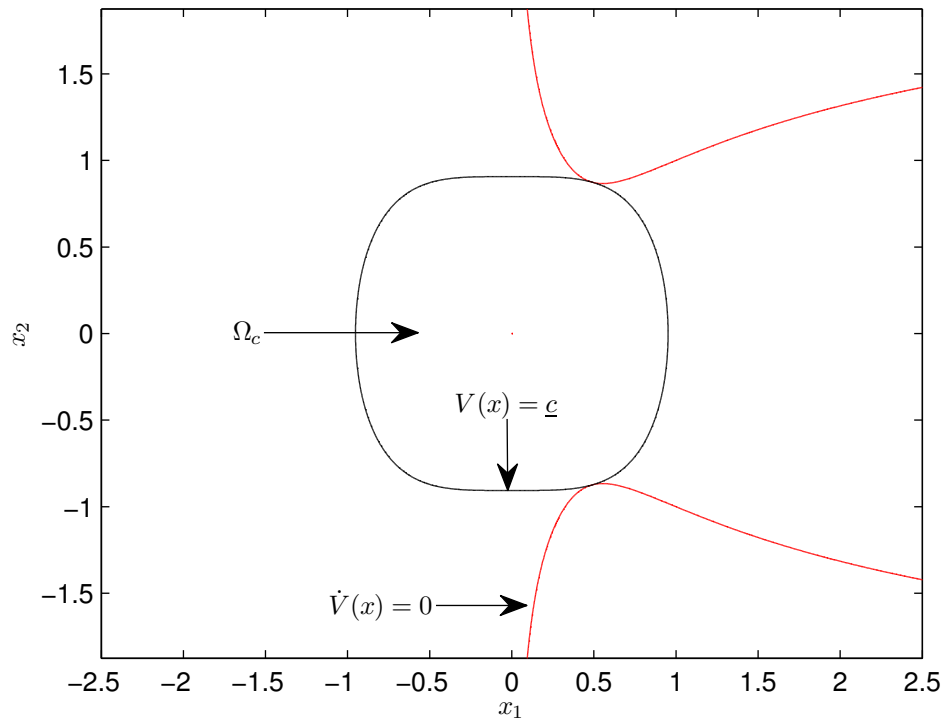


Abbildung 4.23: Ermitteltes Einzugsgebiet für Beispiel 4.20

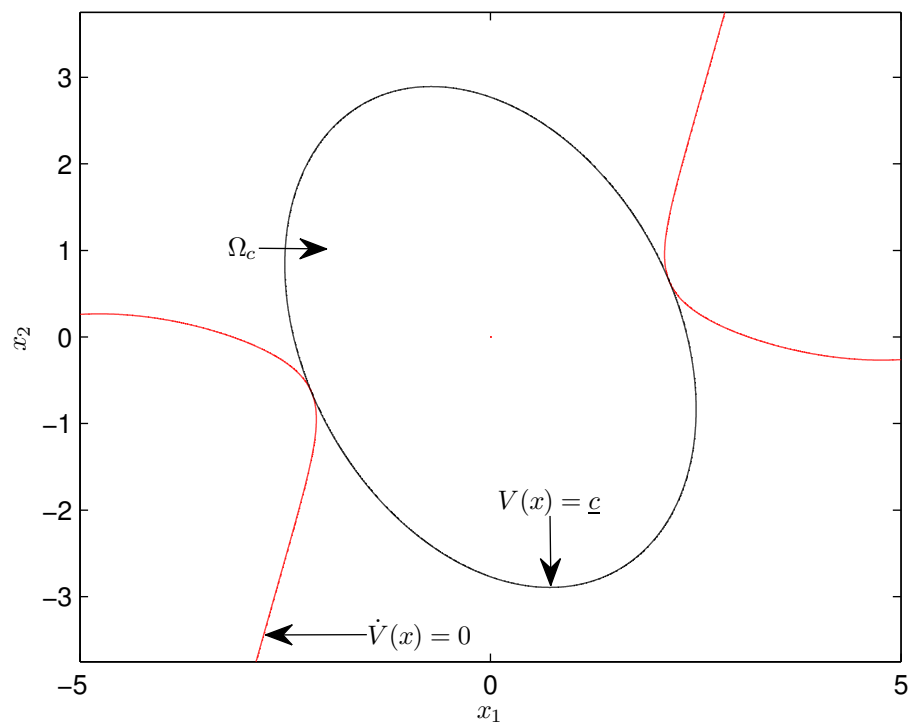


Abbildung 4.24: Ermitteltes Einzugsgebiet für Beispiel 4.21

**Beispiel 4.22**

Für das System

$$\begin{aligned}\dot{x}_1 &= -0.25x_1 + \ln(1 + x_2) \\ \dot{x}_2 &= -0.275x_1 - 0.2x_1x_2 + (0.125x_1 - x_2) \cos x_1\end{aligned}$$

soll für die Ruhelage im Ursprung ein Einzugsgebiet mit der Ljapunowfunktion

$$V(x) = x_1^2 + x_2^2$$

bestimmt werden. Mit der hier vorgestellten Methode können die Schranken

$$\begin{aligned}c &= 0.2737075 \\ \underline{c} &= 0.2737076\end{aligned}$$

ermittelt werden, welche das optimale  $c^*$  enger einschließen als die in [11] ermittelte untere Schranke  $\underline{c} = 0.2606$ . Das resultierende Einzugsgebiet kann Abbildung 4.25 entnommen werden.

**Beispiel 4.23**

Das System

$$\begin{aligned}\dot{x}_1 &= 1 + x_3 + 0.125x_3^2 - \exp(x_1) \\ \dot{x}_2 &= -x_2 - x_3 \\ \dot{x}_3 &= -x_2 - 2x_3 - 0.5x_1^2\end{aligned}$$

soll mit der quadratischen Ljapunowfunktion

$$V(x) = x_1^2 + x_2^2 + x_3^2$$

betrachtet werden. Mithilfe der berechneten Schranken

$$\begin{aligned}c &= 2.6613837 \\ \underline{c} &= 2.6613839\end{aligned}$$

kann das in Abbildung 4.26 dargestellte Einzugsgebiet

$$\Omega_c = \left\{ x \in \mathbb{R}^3 \mid V(x) \leq 2.6613839 \right\}$$

bestimmt werden. Die hier bestimmte untere Schranke für  $c^*$  ist enger als das in [11] ermittelte  $\underline{c} = 2.655$ .

**Beispiel 4.24**

In [12] wird das System

$$\begin{aligned}\dot{x}_1 &= -x_1 + x_2 + 0.5(\exp(x_1) - 1) \\ \dot{x}_2 &= -x_1 - x_2 + x_1x_2 + x_1 \cos x_1\end{aligned}$$

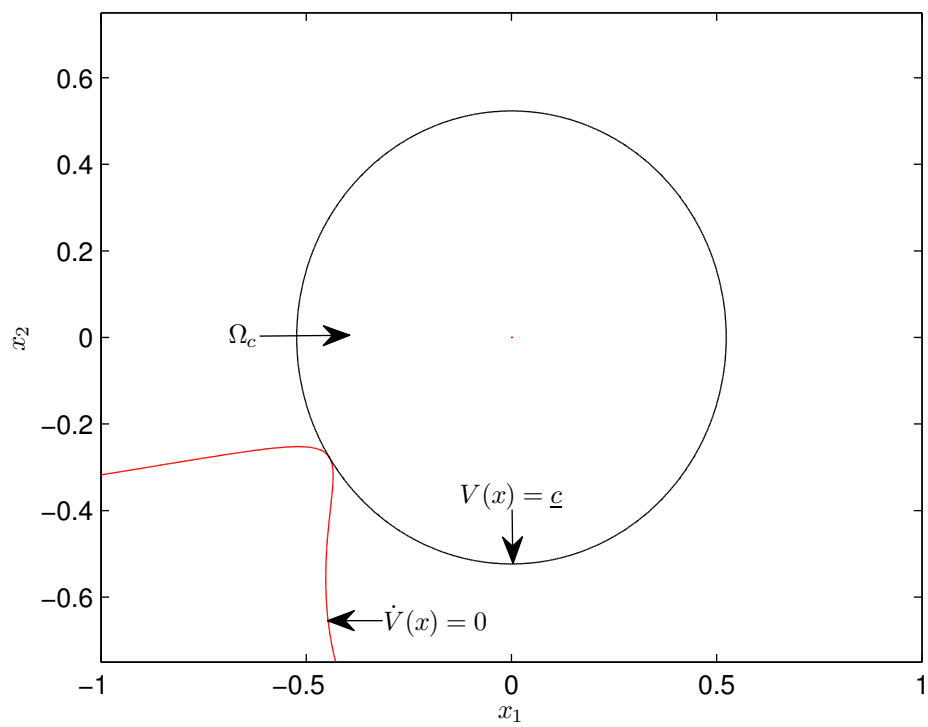


Abbildung 4.25: Ermitteltes Einzugsgebiet für Beispiel 4.22

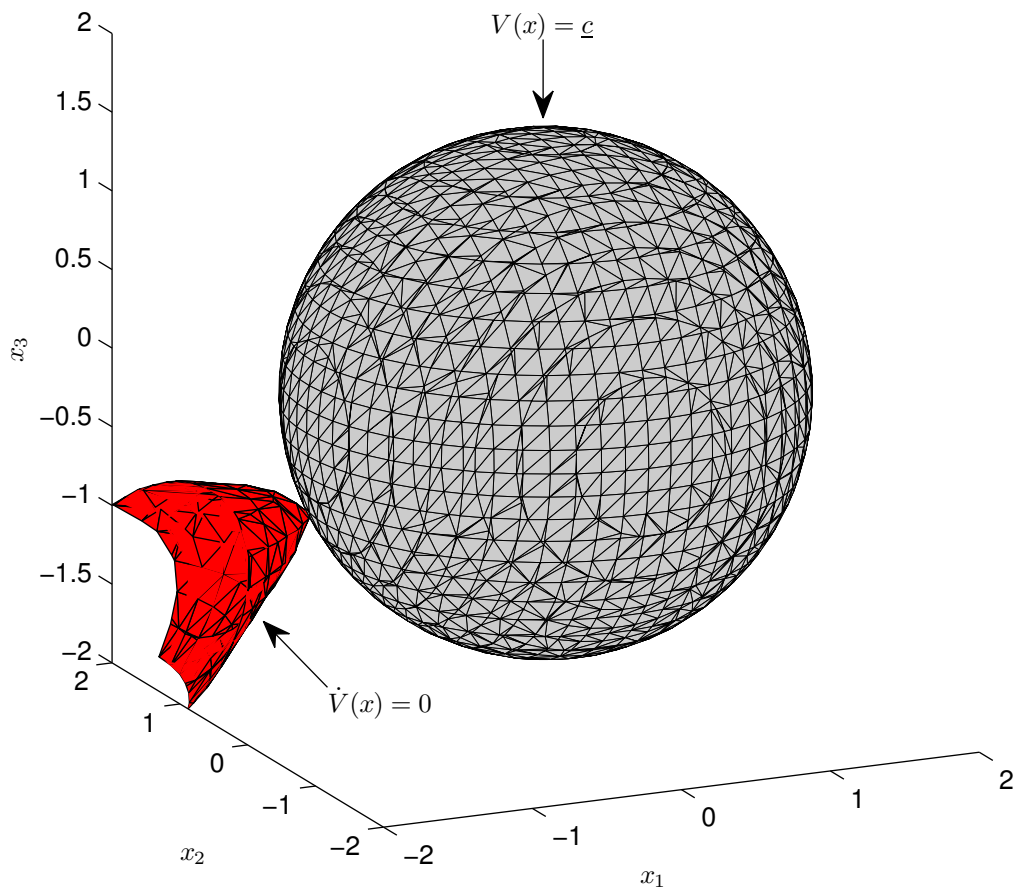


Abbildung 4.26: Ermitteltes Einzugsgebiet für Beispiel 4.23

mit der Ljapunowfunktion

$$V(x) = x_1^2 + x_2^2$$

betrachtet. Das in [12] ermittelte  $\underline{c} = 0.3210$  konnte anhand der mit der neuen Methode berechneten Schranken

$$\begin{aligned} c &= 0.3210740 \\ \underline{c} &= 0.3210741 \end{aligned}$$

verifiziert werden.

Das resultierende Einzugsgebiet kann Abbildung 4.27 entnommen werden.

#### Beispiel 4.25

Die Systemdynamik eines Rotationspendels in [12] wird beschrieben durch die Differenzialgleichungen

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -0.2x_2 + 0.81 \sin x_1 \cos x_1 - \sin x_1 \quad . \end{aligned}$$

Mit der quadratischen Ljapunowfunktion

$$V(x) = x_1^2 + x_1x_2 + 4x_2^2$$

wird in [12] die Menge

$$\Omega_c = \{x \in \mathbb{R}^n \mid V(x) \leq 0.6990\}$$

als gesichertes Einzugsgebiet bestimmt. Mit den ermittelten Schranken

$$\begin{aligned} c &= 0.6992996 \\ \underline{c} &= 0.6992998 \end{aligned}$$

kann das Einzugsgebiet auf

$$\Omega_c = \left\{ x \in \mathbb{R}^2 \mid V(x) \leq 0.6992996 \right\}$$

vergrößert werden. Dieses Gebiet ist auch in Abbildung 4.28 dargestellt.

Zum schnellen Nachschlagen sind die für die Beispiele 4.1 bis 4.25 mit der neuen Methode bestimmten Schranken in den Tabellen 4.1 bis 4.5 die untere Schranke der jeweiligen kompetitiven Methode gegenübergestellt.

Beim Vergleich fällt auf, dass die Ergebnisse mit Ausnahme der Beispiele 4.2 und 4.5 die mit den kompetitiven Methoden ermittelten Schranken mindestens

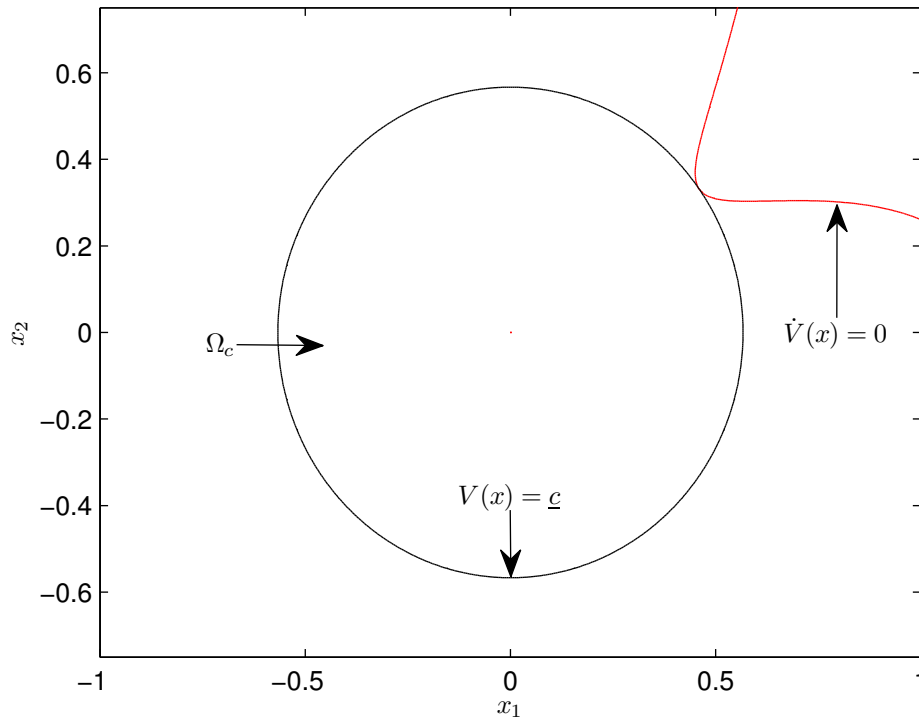


Abbildung 4.27: Ermitteltes Einzugsgebiet für Beispiel 4.24

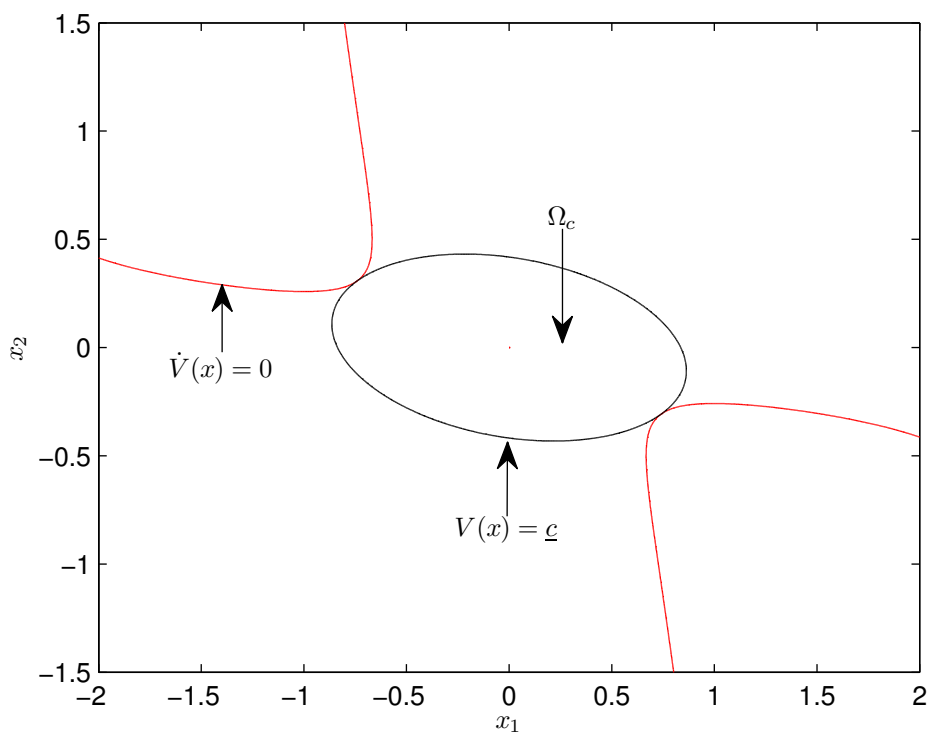


Abbildung 4.28: Ermitteltes Einzugsgebiet für Beispiel 4.25

bestätigt und zum Teil wie Fall der Beispiele 4.13, 4.15, 4.21, 4.22, 4.23 und 4.25 leicht und im Falle der Beispiele 4.16, 4.17 sowie 4.18 sogar deutlich verbessert werden.

Bei der Betrachtung der Beispiele 4.2 und 4.5 konnte nachgewiesen werden, dass die mit der Methode aus [35] ermittelten unteren Schranken zu groß sind, woraus ein Gebiet als Einzugsgebiet angegeben wird, welches im schlimmsten Fall Punkte enthält, welche nicht zum tatsächlichen Einzugsgebiet gehören. Das Problem liegt hier nicht an der Methode selbst, sondern wahrscheinlich an der numerischen Instabilität des verwendeten LMI-Lösers. Da die meisten Verfahren zur Bestimmung von Einzugsgebieten auf LMI-Löser zurückgreifen, kann bei diesen Verfahren nicht ausgeschlossen werden, dass diese eventuell ebenfalls zu große Gebiete liefern. Der hier vorgestellte Algorithmus weist diese Problematik hingegen nicht auf.

|     | System  | Ljapunowfunktion                        | Benchmarkergebnis  | neue Methode                                   |
|-----|---|---|--|--|
| 4.1 | $\dot{x}_1 = -x_1$<br>$\dot{x}_2 = -x_2 + x_1^2 x_2$  | $x_1^2 + x_2^2$                         | $c = 3.999999$<br>$\underline{c} = 4.000000$<br>[35, Bsp. 1]     | $c = 3.9999999$<br>$\underline{c} = 4.0000000$ |
| 4.2 | $\dot{x}_1 = -x_1 + x_2^2$<br>$\dot{x}_2 = -2x_2$   | $0.5(x_1^2 + x_2^2)$                    | $c = 8.81831$<br>$\underline{c} = 8.8182998$<br>[35, Bsp. 2]     | $c = 8.8182996$<br>$\underline{c} = 8.8182998$ |
| 4.3 | $\dot{x}_1 = -2x_1 + x_1 x_2$<br>$\dot{x}_2 = -x_2 + x_1 x_2$   | $x_1^2 + x_2^2$                         | $c = 4.0954841$<br>$\underline{c} = 4.0954841$<br>[35, Bsp. 3]   | $c = 4.0954839$<br>$\underline{c} = 4.0954841$ |
| 4.4 | $\dot{x}_1 = -x_1 + 2x_1^2 x_2$<br>$\dot{x}_2 = -x_2$   | $0.33x_1^2 + 0.498x_1 x_2 + 0.376x_2^2$ | $c = 0.996167$<br>$\underline{c} = 0.9961679$<br>[35, Bsp. 4]    | $c = 0.9961677$<br>$\underline{c} = 0.9961679$ |
| 4.5 | $\dot{x}_1 = -x_1 + x_2$<br>$\dot{x}_2 = 0.1x_1 - x_2 - x_1^2 - 0.1x_1^3$   | $x_1^2 + x_2^2$                         | $c = 7.112889$<br>$\underline{c} = 7.1114711$<br>[35, Bsp. 5]    | $c = 7.1114710$<br>$\underline{c} = 7.1114711$ |
| 4.6 | $\dot{x}_1 = -x_1 + x_2 x_3^2$<br>$\dot{x}_2 = -x_2 + x_1 x_2$<br>$\dot{x}_3 = -x_3$  | $x_1^2 + x_2^2 + x_3^2$                 | $c = 4.9187584$<br>$\underline{c} = 4.9187585$<br>[35, Bsp. 6]   | $c = 4.9187584$<br>$\underline{c} = 4.9187585$ |
| 4.7 | $\dot{x} = \begin{pmatrix} (x - \delta)^T (x - \delta) - 1 \end{pmatrix} x$<br>$x, \delta \in \mathbb{R}^5, \delta = 0.2 \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \end{pmatrix}^T$ | $x^T x$                                 | $c = 0.305572809$<br>$\underline{c} = 0.3055729$<br>[35, Bsp. 7] | $c = 0.3055728$<br>$\underline{c} = 0.3055729$ |

Tabelle 4.1: Berechnung von Einzugsgebieten für polynomial Systeme

|      | System  | Ljapunowfunktion   | Benchmarkergebnis                                | neue Methode                             |
|------|---|--|--|--|
| 4.8  | $\dot{x}_1 = x_2$<br>$\dot{x}_2 = -x_1 - x_2 (1 - x_1^2)$   | $x_1^2 + x_1x_2 + x_2^2$   | $c = 0.915$<br>$\bar{c} = [13, \text{Bsp. 4.4}]$ | $c = 0.9152361$<br>$\bar{c} = 0.9152362$ |
| 4.9  | $\dot{x}_1 = -3x_1 + 4x_1^2 - 0.5x_1x_2 - x_1^3$<br>$\dot{x}_2 = -2.1x_2 + x_1x_2$  | $4x_1^2 + x_1x_2 + 2x_2^2$   | $c = 3.376$<br>$\bar{c} = [13, \text{Bsp. 4.5}]$ | $c = 3.3762171$<br>$\bar{c} = 3.3762172$ |
| 4.10 | $\dot{x}_1 = -0.42x_1 - 2.3x_1^2 - 0.5x_1x_2 - 1.05x_2 - x_1$<br>$\dot{x}_2 = x_1x_2 + 1.98x_1$   | $4x_1^2 + x_1x_2 + 2x_2^2$   | $c = 0.022$<br>$\bar{c} = [13, \text{Bsp. 4.5}]$ | $c = 0.0216741$<br>$\bar{c} = 0.0216742$ |
| 4.11 | $\dot{x}_1 = -x_1 + 2x_2^2$<br>$\dot{x}_2 = x_1^2 - x_2 + x_3^2$<br>$\dot{x}_3 = -x_1^2 - x_3$  | $0.5(x_1^2 + x_2^2 + x_3^2)$   | $c = 0.395$<br>$\bar{c} = [26, \text{Bsp. 6.3}]$ | $c = 0.3950264$<br>$\bar{c} = 0.3950265$ |
| 4.12 | $\dot{x}_1 = -x_1 - x_1x_2$<br>$\dot{x}_2 = -x_2 - x_1^3$   | $x_1^2 + 2x_1x_2 + 2x_2^2$   | [26, Bsp. 6.13]                                  | $c = 0.5455994$<br>$\bar{c} = 0.5455995$ |
| 4.13 | $\dot{x}_1 = -x_1 + x_1^2x_2$<br>$\dot{x}_2 = -2x_2$  | $\left(\frac{x_1}{(2.59)}\right)^2 + \left(\frac{x_1}{1.32}\right)^2$  | $c = 1$<br>$\bar{c} = [58, \text{Bsp. 1}]$       | $c = 1.0952614$<br>$\bar{c} = 1.0952616$ |
| 4.14 | $\dot{x}_1 = -\frac{13}{10}x_1 - \frac{3}{2}x_2 + \frac{1}{2}x_3 - \frac{1}{2}x_1x_2$<br>$\dot{x}_2 = \frac{4}{5}x_1 + \frac{1}{2}x_1x_2$<br>$\dot{x}_3 = \frac{1}{2}x_2 - x_3$ | $\frac{299}{475}x_1^2 + \frac{6457}{3800}x_2^2$<br>$+ \frac{286}{475}x_3^2 + \frac{378}{475}x_1x_2$<br>$+ \frac{184}{475}x_2x_3 + \frac{194}{475}x_1x_3$ | $c = 2.6094$<br>$\bar{c} = [57]$                 | $c = 2.6094420$<br>$\bar{c} = 2.6094421$ |

Tabelle 4.2: Berechnung von Einzugsgebieten für polynomiale Systeme (Fortsetzung)



| Beispiel | System   | Benchmark                                 | neue Methode  |
|----------|--|---|---|
| 4.15     | $\dot{x}_1 = -x_1 + 2x_1^2x_2$<br>$\dot{x}_2 = -x_2$                               | $\underline{c} = 0.450$<br>[35, Bsp. 8]   | $\underline{c} = 0.4647959$<br>$\overline{c} = 0.4647960$ |
| 4.16     | $\dot{x}_1 = x_2$<br>$\dot{x}_2 = -2x_1 - 3x_2 + x_1^2x_2$                         | $\underline{c} = 6.680$<br>[35, Bsp. 9]   | $\underline{c} = 7.9296479$<br>$\overline{c} = 7.9296480$ |
| 4.17     | $\dot{x}_1 = 4x_2 + x_1^3$<br>$\dot{x}_2 = -12x_1 - 16x_2 - 4x_2^3$                | $\underline{c} = 5.848$<br>[35, Bsp. 10]  | $\underline{c} = 7.5608322$<br>$\overline{c} = 7.5608323$ |
| 4.18     | $\dot{x}_1 = -2x_1 + x_2 + x_1^3 + x_2^5$<br>$\dot{x}_2 = -x_1 - x_2 - x_1^2x_2^3$ | $\underline{c} = 0.992$<br>[35, Bsp. 12]  | $\underline{c} = 1.2549328$<br>$\overline{c} = 1.2549329$ |
| 4.19     | $\dot{x}_1 = x_2$<br>$\dot{x}_2 = -x_1 - x_2(1 - x_1^2)$                           | $\underline{c} = 3.047$<br>[13, Bsp. 4.4] | $\underline{c} = 3.0470403$<br>$\overline{c} = 3.0470404$ |
| 4.20     | $\dot{x}_1 = x_2^4 - 2x_1$<br>$\dot{x}_2 = 3x_1x_2^3 - x_2$                        | [26, Bsp. 6.14]                           | $\underline{c} = 0.8211190$<br>$\overline{c} = 0.8211191$ |

Tabelle 4.3: Berechnung von Einzugsgebieten für polynomiale Systeme mit Ljapunowfunktionen höherer Ordnung

| Beispiel | Ljapunowfunktion  |
|----------|---|
| 4.15     | $0.2991x_1^2 + 0.06027x_1^3x_2 + 0.4513x_1x_2 + 0.003036x_1^4x_2^2$<br>$+0.04548x_1^2x_2^2 + 0.34079x_2^2$  |
| 4.16     | $2.333x_1^2 + 1.353x_1x_2 + 0.1152x_1^3x_2 + 0.4306x_2^2 + 0.092x_1^2x_2^2$<br>$+0.00509x_1^4x_2^2$   |
| 4.17     | $1.398x_1x_2 + 0.002173x_1^3x_2 - 0.0085276x_1x_2^3 - 0.00001326x_1^3x_2^3$<br>$3.816x_1^2 + 0.011785x_1^4 + 0.3774x_2^2 + 0.0047097x_2^4 + 0.0000090992x_1^6$<br>$+0.0000146952x_2^6$  |
| 4.18     | $1.3637x_2^2 + 0.31873x_1^2 + 0.14717x_1x_2 + 0.0989x_1^3x_2 + 0.11578x_1x_2^5$<br>$+0.40147x_1^2x_2^4 + 0.004794x_1^3x_2^3 + 0.029774x_1^4x_2^6 + 0.011599x_1^2x_2^8$<br>$+0.01159x_1^5x_2^3 + 0.02296x_1^3x_2^5 + 0.011481x_2^{10} + 0.011481x_1^6$<br>$+0.115782x_1^4 + 0.098905x_2^6$ |
| 4.19     | $x_1^2 + x_1x_2 + x_2^2 + x_1^4 + x_1^2x_2^2 + x_2^4$   |
| 4.20     | $x_1^4 + x_2^2$   |

Tabelle 4.4: Ljapunowfunktionen für die Beispiele aus Tabelle 4.3

| Beispiel | System   | Ljapunowfunktion            | Benchmark                                       | neue Methode                               |
|----------|--|-----------------------------|---|--|
| 4.21     | $\dot{x}_1 = x_2$<br>$\dot{x}_2 = -x_2 - \sin(x_1)$  | $4x_1^2 + 2x_1x_2 + 3x_2^2$ | $c = 22.94$<br>$\bar{c} = [11, \text{Bsp. 1}]$  | $c = 23.0071867$<br>$\bar{c} = 23.0071868$ |
| 4.22     | $\dot{x}_1 = -0.25x_1 + \ln(1 + x_2)$<br>$\dot{x}_2 = -0.275x_1 - 0.2x_1x_2 + (0.125x_1 - x_2) \cos x_1$           | $x_1^2 + x_2^2$             | $c = 0.2606$<br>$\bar{c} = [11, \text{Bsp. 2}]$ | $c = 0.2737075$<br>$\bar{c} = 0.2737076$   |
| 4.23     | $\dot{x}_1 = 1 + x_3 + 0.125x_3^2 - \exp(x_1)$<br>$\dot{x}_2 = -x_2 - x_3$<br>$\dot{x}_3 = -x_2 - 2x_3 - 0.5x_1^2$ | $x_1^2 + x_2^2 + x_3^2$     | $c = 2.655$<br>$\bar{c} = [11, \text{Bsp. 3}]$  | $c = 2.6613837$<br>$\bar{c} = 2.6613839$   |
| 4.24     | $\dot{x}_1 = -x_1 + x_2 + 0.5(\exp(x_1) - 1)$<br>$\dot{x}_2 = -x_1 - x_2 + x_1x_2 + x_1 \cos x_1$                  | $x_1^2 + x_2^2$             | $c = 0.3210$<br>$\bar{c} = [12, \text{Bsp. 1}]$ | $c = 0.3210740$<br>$\bar{c} = 0.3210741$   |
| 4.25     | $\dot{x}_1 = x_2$<br>$\dot{x}_2 = -0.2x_2 + 0.81 \sin x_1 \cos x_1 - \sin x_1$                                     | $x_1^2 + x_1x_2 + 4x_2^2$   | $c = 0.6990$<br>$\bar{c} = [12, \text{Bsp. 2}]$ | $c = 0.6992996$<br>$\bar{c} = 0.6992998$   |

Tabelle 4.5: Berechnung von Einzugsgebieten für nichtpolynomiale Systeme

## KAPITEL FÜNF

# Parallelisierung

### 5.1 Einleitung

Moderne Rechner verfügen bis auf wenige Ausnahmen meistens über mehrere Rechenkerne, selbst in moderne Smartphones halten Doppelkernprozessoren Einzug. Der Vorteil liegt hierbei darin, dass mehrere Aufgaben zeitgleich ausgeführt werden können, allerdings auch - sofern die Aufgabenstellung hierfür ausgelegt ist - eine Aufgabenstellung aufgeteilt auf mehreren Kernen berechnet werden kann. Hierbei teilen sich mehrere Kerne Arbeitsspeicher und Peripherie.

Eine weitere Möglichkeit, eine Aufgabe zu verteilen, ist es, Teilaufgaben auf mehreren getrennten Rechnern - im Folgenden Rechnerknoten genannt - zu berechnen. Hierbei verfügen die getrennten einzelnen Rechner über eigenen Arbeitsspeicher und kommunizieren untereinander über ein Netzwerk.

Durch die Kommunikation und die Teilung von Arbeitsspeicher und Peripherie geht Rechenzeit verloren, sodass die Ausführungsgeschwindigkeit nicht mit der Anzahl der Rechenkerne bzw. Parallelrechner linear ansteigt. Bei schlecht parallelisierbaren Aufgabenstellungen kann die Parallelisierung sogar zu einer Erhöhung der Laufzeit führen. Theoretische Betrachtungen zur Parallelisierbarkeit von Algorithmen wurden beispielsweise von Amdahl [5] und Gustafson [34] durchgeführt, allerdings werden gewisse Effekte, welche zu einer Beschleunigung oder Verlangsamung des zu parallelisierenden Codes führen können, hierbei nicht berücksichtigt, sodass allgemeine Aussagen zu dieser Thematik nicht zu treffen sind.

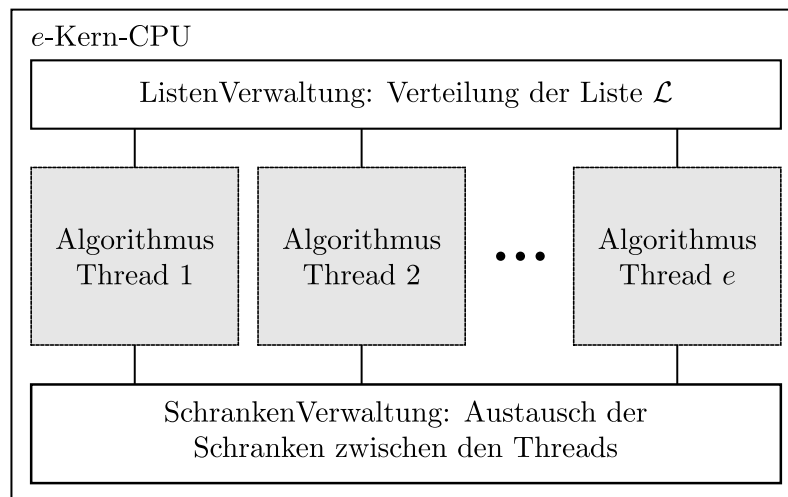


Abbildung 5.1: Architektur der Parallelisierung auf einer CPU mit  $e$  Kernen

In Zusammenarbeit mit Mayer [64, 65] wurde untersucht, inwieweit der in Kapitel 4 vorgestellte Algorithmus von der Parallelisierung profitiert.

## 5.2 Umsetzung

Da der in Kapitel 4 vorgestellte Algorithmus seriell abgearbeitet wird, müssen zur Parallelisierung Anpassungen vorgenommen werden. In [18] werden zwei Arten beschrieben, wie ein Branch-and-Bound-Algorithmus parallelisiert werden kann. Zum Einen kann die komplette Datenstruktur parallelisiert werden, sodass sämtliche Funktionen auf allen Rechnerknoten ausgeführt werden. Die andere Möglichkeit liegt darin, die Testaufgaben auf die einzelnen Rechnerknoten zu verteilen. Hier müssten nur die Grenzen und bei Bedarf Listen ausgetauscht werden.

Im Folgenden wird der zweite Ansatz weiterverfolgt. Zuerst soll die die Parallelisierung auf einem Knoten mit  $e$  Kernen betrachtet werden. Wie in Abbildung 5.1 ersichtlich, laufen auf den einzelnen Kernen Threads mit Teilaufgaben, welche untereinander über zwei separate Datenstrukturen die Schranken (Schranken-Verwaltung) sowie Listen (ListenVerwaltung) austauschen. Da diese Strukturen im Speicher liegen, auf welche alle Threads zugreifen können, muss hierfür ein Mutex [89] beziehungsweise wechselseitiger Ausschluss verwendet werden. Alle Threads sollen, wie Abbildung 5.2 zu entnehmen ist, nun eine eigene Liste

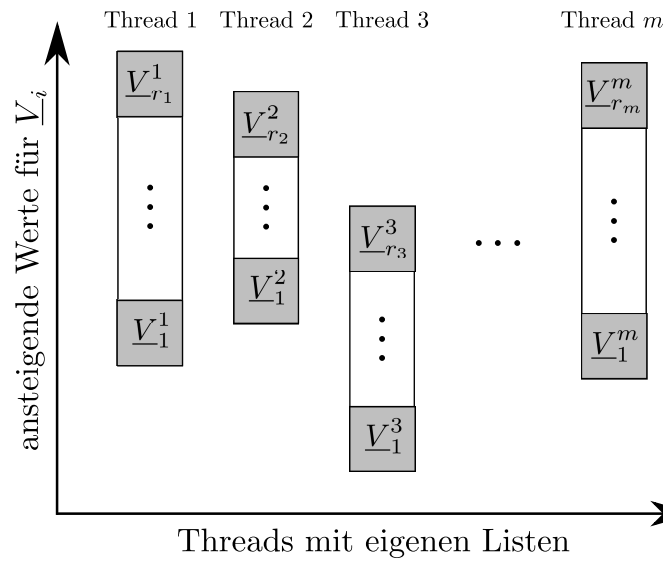


Abbildung 5.2: Verwaltung von parallelen Listen

verwalten, welche jeweils Teil der Gesamtliste  $\mathcal{L}$  sind. Läuft die Liste innerhalb eines Threads leer oder werden die Schranken aktualisiert, muss auf die entsprechende Datenstruktur zugegriffen werden, um entweder eine neue Teilliste anzufordern oder die anderen Threads über die aktualisierten Schranken zu informieren. Gleichzeitig müssen die Schranken regelmäßig ausgelesen werden, um die Liste gegebenenfalls verkleinern zu können.

Um nun mehrere Rechner über ein Kommunikationsnetzwerk zu verbinden, wird eine Struktur verwendet, wie sie in Abbildung 5.3 dargestellt wird. Der Kommunikationsthread in den jeweiligen Rechnerknoten nutzt die MPI-Schnittstelle [32], um die Schranken und Listen zwischen den Rechnerknoten auszutauschen. Bei der Ausführung ist zu unterscheiden, ob der Kommunikationsthread einem separaten Kern des Knotens zugewiesen wird oder ob er auf einem der Kerne, welche bereits eine Teilaufgabe berechnen, ausgeführt wird. Im ersten Fall stehen nur  $e - 1$  Threads zur Berechnung des Teilproblems zur Verfügung, allerdings bremst der Rechenaufwand des Kommunikationsprozesses nicht die Berechnung der Teilaufgabe aus und umgekehrt. Allerdings ist der entsprechende Kern, solange keine Kommunikation erfolgt, nicht ausgelastet. Im zweiten Fall sind alle Kerne vollständig ausgelastet, allerdings konkurriert der Kommunikationsthread mit einem der Threads zur Lösung der Teilaufgabe. Im weiteren Verlauf wird untersucht werden, inwieweit sich dies auf den Speedup auswirkt.

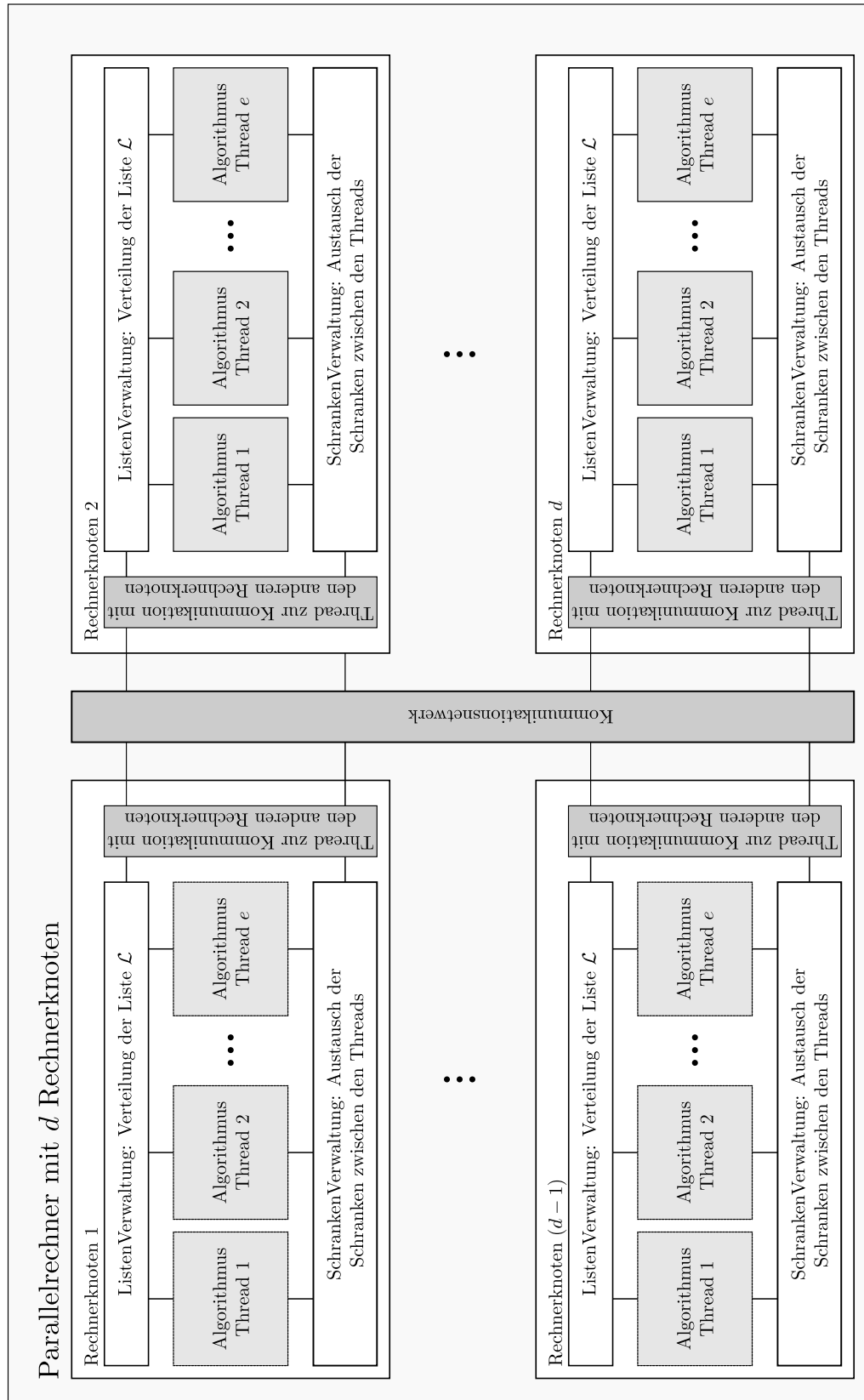


Abbildung 5.3: Architektur der Parallelisierung auf einem Mehrkernrechner-Verbund (nach [64])

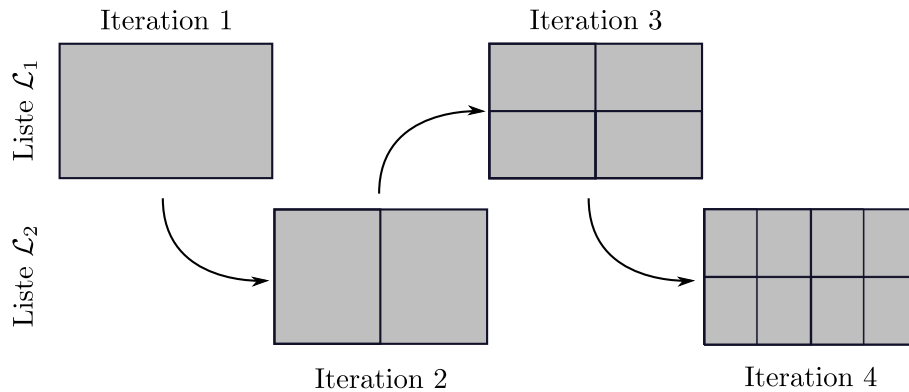


Abbildung 5.4: Zwei-Listen-Konzept

Der in dieser Arbeit präsentierte Algorithmus nimmt eine Tiefensuche bei der Suche des Optimums vor. Das heißt, es wird versucht, die Auswertung des Intervalls mit der niedrigsten unteren Schranke für  $V(x)$  zu verbessern. Durch die dadurch auftretenden häufigen Aktualisierungen der Schranken wird viel Rechenzeit für den Austausch der Schranken verwendet.

Für die Implementierung wurde nun ein Zwei-Listen-Konzept gewählt, wie es in Abbildung 5.4 dargestellt ist. Das Startintervall ist in der Liste  $\mathcal{L}_1$  gespeichert und wird dort im ersten Schritt entnommen. Alle zurückzusortierenden Listen werden nun solange in die Liste  $\mathcal{L}_2$  geschrieben, bis die Liste  $\mathcal{L}_1$  leer ist. Danach werden die Intervalle solange der Liste  $\mathcal{L}_2$  entnommen, bisektiert und in Liste  $\mathcal{L}_1$  zurückgeschrieben, bis  $\mathcal{L}_2$  leer ist. Durch Wiederholung dieser Schritte wird eine Breitensuche vorgenommen, mit dem Ziel, weniger Aktualisierungen der Schranken vornehmen zu müssen und trotzdem gleichzeitig die Listengröße in jedem Thread überschaubar zu halten.

## 5.3 Effizienzbetrachtung

Zu dem Zweck, die Effizienz der Parallelisierung in den Blick zu nehmen, wurde von Mayer das Konzept aus Abschnitt 5.2 auf einem Parallelrechner-System implementiert. Dieses bestand aus einem Server mit AMD Phenom II X4 905E Prozessor mit 4 Rechenkernen in einem Prozessor und 6 GB Arbeitsspeicher sowie 3 Clients mit dem gleichen Prozessor und 4 GB Arbeitsspeicher. Die einzelnen Rechnerknoten wurden über ein Gigabit-Ethernet-Netzwerk miteinander verbunden. Als Betriebssystem wurde OpenSuSE 11.2 [87] in der 64-Bit-Version

| Knoten | Threads pro Knoten | durchschnittliche Laufzeit [s] | Standardabweichung (rel. Fehler) | S     | S <sub>max</sub> |
|--------|--------------------|--------------------------------|----------------------------------|-------|------------------|
| 1      | 1                  | 2859.16                        | 82.86 (2,9%)                     | 1     | 1                |
| 1      | 3                  | 1038.69                        | 67.42 (6,5%)                     | 2.75  | 3                |
| 1      | 4                  | 918.41                         | 69.75 (7,6%)                     | 3.11  | 4                |
| 2      | 3                  | 491.45                         | 45.95 (9,3%)                     | 5.82  | 6                |
| 2      | 4                  | 443.24                         | 98.33 (22,2%)                    | 6.45  | 8                |
| 3      | 3                  | 343.05                         | 42.09 (12,3%)                    | 8.33  | 9                |
| 3      | 4                  | 293.26                         | 37.22 (12,7%)                    | 9.75  | 12               |
| 4      | 3                  | 282.53                         | 78.44 (27,8%)                    | 10.12 | 12               |
| 4      | 4                  | 242.65                         | 31.63 (13,0%)                    | 11.78 | 16               |

Tabelle 5.1: Laufzeiten für verschiedene Konfigurationen

ausgewählt und der Algorithmus in Qt/C++ implementiert. Für die Intervallarithmetik wurde `filib++` [55] verwendet. Um Ableitungen der Ljapunowfunktion und der zeitlichen Ableitungen schnell und effizient zu bestimmen, wurde die automatische Differenziation [31] mithilfe von `FADBAD++` [85] genutzt. Zur Verwaltung von Threads auf einem Knoten wurde mit `QThreads` aus der Qt-Klassenbibliothek [67, 74] für C++ verwendet, die MPI-Schnittstelle wurde durch `MPICH2` [6, 33] realisiert.

Der Speedup  $S$  der Parallelisierung ist definiert durch den Quotienten aus der Laufzeit des parallelen und des seriellen Algorithmus, das heißt,

$$S := \frac{t(p)}{t(1)},$$

wobei  $t(p)$  hier die Laufzeit des Algorithmus mit insgesamt  $p$  Threads beschreibt. Mit  $S_{\max}$  soll der theoretische Speedup bezeichnet werden, welcher sich aus dem Produkt aus der Anzahl der Verfügung stehenden Rechenknoten  $d$  und der Anzahl der für die Rechnung zur Verfügung stehenden Threads pro Knoten (entweder  $e$  oder  $e - 1$ ) berechnet.

Zur Analyse wurde mit Beispiel 4.7 ein möglichst komplexes Beispiel ausgewählt und insgesamt 100-fach ausgeführt. Gemittelt wurde eine Laufzeit  $t(1) = 2859.16$  Sekunden gemessen, wobei die Standardabweichung 82.86 Sekunden beträgt. Danach wurden für verschiedene Konfigurationen jeweils 100 Messungen durchgeführt; deren Ergebnisse sind Tabelle 5.1 zu entnehmen. Hierbei haben Konfigura-



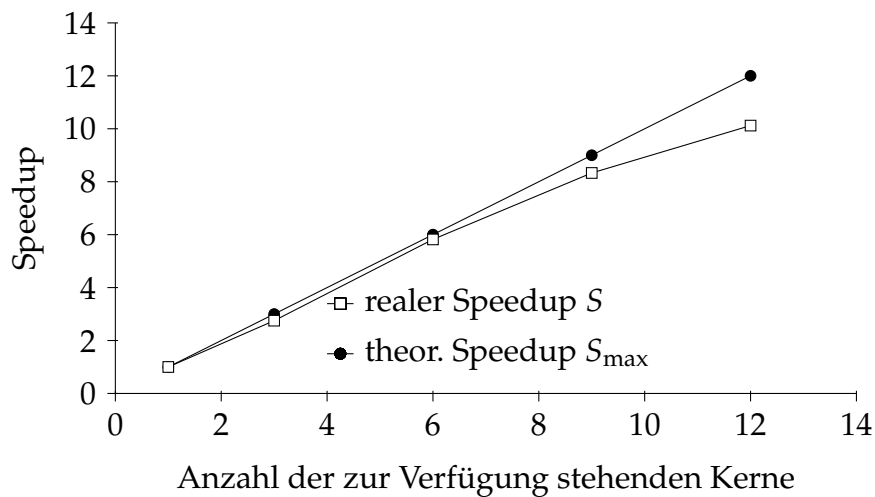


Abbildung 5.5: Konfiguration mit 3 Threads pro 4-Kern-Prozessor

tionen mit jeweils drei Threads pro Kern einen separaten Kommunikationsthread pro Knoten, während sich in Konfigurationen mit vier Threads pro Kern der Kommunikationsthread die Rechenzeit mit den Algorithmus-Threads teilt.

Es fällt auf, dass Konfigurationen mit zusätzlichen Thread pro Kern, welcher sich Ressourcen mit dem Kommunikationsthread teilt, die Aufgabenstellung schneller lösen, als dies Konfigurationen mit separatem Kommunikationsthread leisten können. Allerdings lässt sich auch erkennen, dass der reale Speedup in diesen Konfigurationen schlechter ist als der theoretische. Insgesamt ist festzuhalten, dass es bei der gegebenen Aufgabenstellung und Rechnerkonfiguration sinnvoller ist, alle vorhandenen Kerne zum Lösen der Aufgabenstellung zu nutzen. Für Konfigurationen mit deutlich größerer Anzahl von Threads muss diese Betrachtung allerdings erneut durchgeführt werden.

In den Abbildungen 5.5 und 5.6 ist der jeweilige Speedup für beide Konfigurationsklassen mit steigender Anzahl von Threads aufgezeichnet. In beiden Konfigurationen sieht man, dass erwartungsgemäß mit steigender Anzahl von Threads der Abstand des realen Speedups zum theoretischen Speedup größer wird.

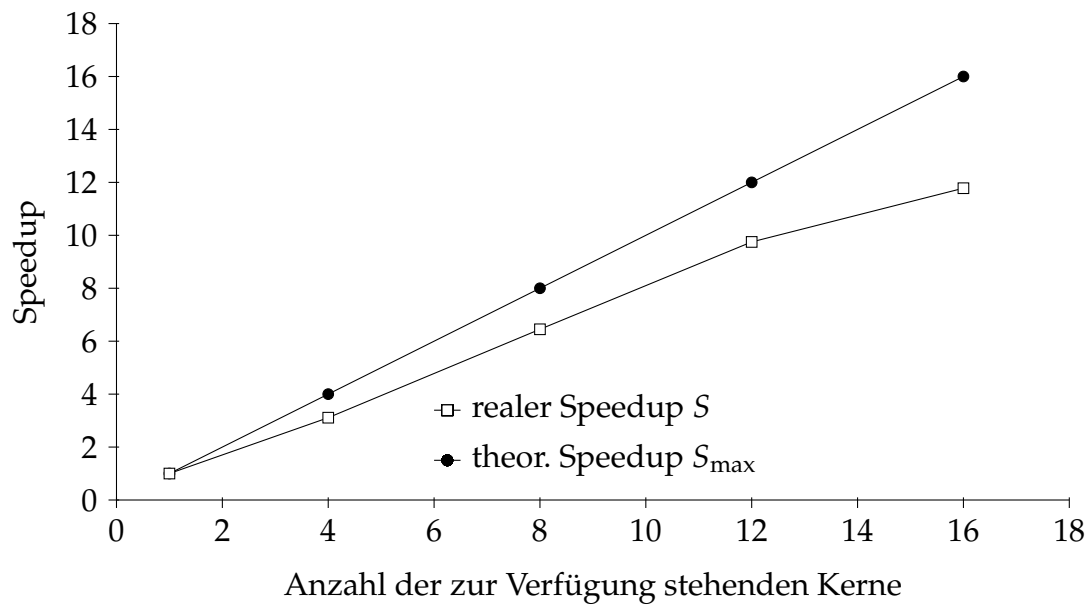


Abbildung 5.6: Konfiguration mit 4 Threads pro 4-Kern-Prozessor

# Zusammenfassung und Ausblick

Die Untersuchung der Stabilität eines dynamischen Systems stellt eine wichtige Aufgabe in der Systemanalyse dar. Bei nichtlinearen Systemen ist der erste Schritt, herauszufinden, ob das System überhaupt stabil im Sinne von Ljapunow ist. Wenn das zu untersuchende System diese Eigenschaft besitzt, interessiert man sich für das Einzugsgebiet, ein Gebiet von Zuständen, in welche das System ausgelenkt werden kann, sodass die Systemtrajektorien in die Ruhelage einlaufen.

Mit Hilfe der Niveaumengen einer sogenannten Ljapunowfunktion ist ein Teilgebiet des exakten Einzugsgebiets algebraisch angebbbar. Das Problem besteht hierbei darin, den maximalen Funktionswert  $c^*$  dieser Ljapunowfunktion anzugeben, sodass das durch die Niveaumengen beschränkte Gebiet ein Teilgebiet des Einzugsgebiets ist.

Zur Lösung dieser Problemstellung wurde in dieser Arbeit ein Algorithmus vorgestellt, der im Gegensatz zu den existierenden Methoden einen neuen Ansatz verfolgt. Anstatt die Aufgabenstellung in die Lösung einer linearen Matrixungleichung zu überführen oder Transformationen der verwendeten Funktionen vornehmen zu müssen, nutzt der Algorithmus die Intervallarithmetik, um Schranken für die Ljapunowfunktion und ihre zeitliche Ableitung zu bestimmen. Als angenehmer Nebeneffekt ist dieses Verfahren bei sorgfältiger Implementierung robust gegenüber Rundungsfehlern. Da das Verfahren neben der unteren Schranke für den gesuchten Wert  $c^*$  auch eine obere Schranke liefert, kann während der Laufzeit eine Aussage über die Qualität der unteren Schranke getroffen werden.

---

Zur Nutzung der Intervallarithmetik stehen Toolboxen bzw. Bibliotheken für zahlreiche Programmiersprachen zur Verfügung. Im Rahmen der Forschung zu dieser Arbeit wurden zwei dieser Implementierungen, nämlich `INTLAB` für `MATLAB` sowie `FILIB++` für `C++`, ausgiebig genutzt, um das neue Verfahren zu implementieren. Zur Evaluation ist eine umfangreiche Literaturrecherche durchgeführt worden, mit dem Ziel, das Verfahren an einer Vielzahl von Beispielen zu testen. Hierbei konnten mindestens die Ergebnisse der existierenden Verfahren bestätigt werden, meistens konnten diese sogar verbessert werden. In zwei Fällen wurde festgestellt, dass das kompetitive, auf der Lösung von LMI basierende Verfahren ein minimal zu großes Einzugsgebiet liefert.

Da der Rechenaufwand mit der Anzahl der Zustände exponentiell ansteigt, wurde eine Untersuchung der Parallelisierbarkeit des Algorithmus durchgeführt, um die Rechengeschwindigkeit zu erhöhen und dadurch diesen Effekt abzumildern. Die Parallelisierbarkeit wurde am Literaturbeispiel mit der höchsten Anzahl von Systemzuständen ( $n = 5$ ) verifiziert. Hierbei konnte eine deutliche Verkürzung der Programmlaufzeit festgestellt werden.

Mit dem vorgestellten Algorithmus lässt sich eine umfangreiche Klasse von nicht-linearen Systemen auf ihre Stabilität untersuchen. Allerdings sind aufgrund der Einschränkungen der Intervallarithmetik bei der Division mit Intervallen, welche die Null enthalten, keine Systeme mit rationalen Funktionstermen in der Systemdifferentialgleichung oder der Ljapunowfunktion betrachtbar. Dies betrifft nicht nur Intervalle, welche mindestens eine Singularität des rationalen Funktionsterms beinhalten, sondern aufgrund der Überschätzung auch weitere Intervalle. Trotz zahlreicher Überlegungen wurde für dieses Problem noch keine anwendbare Lösung gefunden. Vanelli und Vidysagar beweisen in [99] die Existenz einer sogenannten maximalen Ljapunowfunktion, welche allerdings nur schwer zu finden. Diese Funktionen können jedoch durch rationale Funktionen approximiert werden, welche selbst auch wieder Ljapunowfunktionen sind. Hachicho nutzt in seiner Dissertation [35] Funktionen dieser Art aus, um ein größeres Einzugsgebiet zu bestimmen. Aufgrund der dadurch entstehenden Möglichkeiten ist eine Erweiterung des hier vorgestellten Verfahrens hinsichtlich rationaler Funktionen anzustreben.

Das hier dargestellte und begründete Verfahren setzt voraus, dass es sich bei dem zu analysierenden System um ein autonomes System handelt, was voraussetzt,

---

dass der Regler des Systems schon entworfen und in der Systemdifferentialgleichung berücksichtigt wurde. Kommende Verfahren werden jedoch die Betrachtung von Einzugsgebieten nicht nur zur Stabilitätsanalyse, sondern auch zur Reglersynthese nutzen. Ziel ist es hierbei, den Regler zu ermitteln, der das größte Einzugsgebiet eines nichtlinearen Systems garantiert. Dies ist jedoch nicht einfach umzusetzen. Für Erweiterungen dieses Verfahrens sollte untersucht werden, inwieweit der Reglerentwurf mit aufgenommen werden kann.

# Algorithmen

## A.1 Stabilitätsanalyse mit quadratischen Ljapunow-funktionen

- 1: {Initialisierung von  $\underline{c}, \bar{c}$ }
- 2:  $\left[ \underline{V}, \bar{V} \right] \leftarrow V^I(x_{\text{init}}^I)$
- 3:  $\mathcal{L} \leftarrow \left( \underline{V}, x_{\text{init}}^I \right)$
- 4: **while**  $\bar{c} - \underline{c} \geq \varepsilon$  **do**
- 5:   Entferne das erste Paar  $\left( \underline{V}_{-1}, x^{I_1} \right)$  aus der Liste  $\mathcal{L}$
- 6:   Bisectiere  $x^{I_1}$  entlang der längsten Kante in zwei Intervalle  $y^{I_1}$  und  $y^{I_2}$
- 7:   **for**  $i = 1 \rightarrow 2$  **do**
- 8:      $x^I \leftarrow y^{I_i}$
- 9:      $\check{x} \leftarrow \text{mid}(x^I)$
- 10:      $\left[ \underline{V}, \bar{V} \right] \leftarrow V^I(x^I)$
- 11:      $\left[ \underline{\dot{V}}, \bar{\dot{V}} \right] \leftarrow \dot{V}^I(x^I)$
- 12:     **if**  $\underline{V} > \bar{c}$  **or**  $\bar{V} < \underline{c}$  **then**
- 13:       {Verwerfe Intervall}
- 14:     **else if**  $\bar{\dot{V}} > 0$  **then**
- 15:        $V_m \leftarrow V(\check{x})$
- 16:       **if**  $V_m < \bar{c}$  **then**
- 17:          $\bar{c} = V_m$
- 18:     **end if**
- 19:     {Intervall  $x^I$  kann verworfen werden}

```

20:   else if  $\bar{V} < 0$  then
21:     {Intervall  $x^I$  kann verworfen werden}
22:   else
23:     if  $\dot{V}(\check{x}) > 0$  then
24:        $V_m \leftarrow V(\check{x})$ 
25:       if  $V_m < \bar{c}$  then
26:          $\bar{c} = V_m$ 
27:       end if
28:     end if
29:     sortiere Paar  $(\underline{V}, x^I)$  zurück in  $\mathcal{L}$ 
30:   end if
31:   if  $V_{-1} > \max(c_{-init})$  then
32:      $\underline{c} = V_{-1}$ 
33:   end if
34: end for
35: end while

```

## A.2 Stabilitätsanalyse mit Ljapunowfunktionen höherer Ordnung

```

1: {Initialisierung von  $\underline{c}, \bar{c}$ }
2:  $[\underline{V}, \bar{V}] \leftarrow V^I(x_{init}^I)$ 
3:  $\lambda^I \leftarrow [-\infty, \infty]$ 
4:  $\mathcal{L} \leftarrow (\underline{V}, x_{init}^I, \lambda^I)$ 
5: while  $\bar{c} - \underline{c} \geq \varepsilon$  do
6:   Entferne ersten Eintrag  $(\underline{V}_1, x^{I_1}, \lambda^{I_1})$  aus der Liste  $\mathcal{L}$ 
7:   Bisectiere  $x^{I_1}$  entlang der längsten Kante in zwei Intervalle  $y^{I_1}$  und  $y^{I_2}$ 
8:   for  $i = 1 \rightarrow 2$  do
9:      $x^I \leftarrow y^{I_i}$ 
10:     $\lambda^I \leftarrow \lambda^{I_1}$ 
11:     $\check{x} \leftarrow \text{mid}(x^I)$ 
12:     $[\underline{V}, \bar{V}] \leftarrow V^I(x^I)$ 
13:     $[\underline{\dot{V}}, \bar{\dot{V}}] \leftarrow \dot{V}^I(x^I)$ 
14:    if  $\underline{V} > \bar{c}$  or  $\bar{V} < \underline{c}$  then

```

```

15:     {Verwerfe Intervall}
16:     else if  $\dot{V} > 0$  then
17:          $V_m \leftarrow V(\check{x})$ 
18:         {Intervall  $x^I$  kann verworfen werden}
19:     else if  $\bar{V} < 0$  then
20:         {Intervall  $x^I$  kann verworfen werden}
21:     else
22:          $(x^{I_{\text{neu}}}, \lambda^{I_{\text{neu}}}) \leftarrow \text{Krawzykschritt}(x^I, \lambda^I)$ 
23:         if  $x^{I_{\text{neu}}} = \emptyset$  or  $\lambda^{I_{\text{neu}}} = \emptyset$  then
24:             {Intervall  $x^I$  kann verworfen werden}
25:         else
26:             if  $x^{I_{\text{neu}}} \subset x^I$  and  $\lambda^{I_{\text{neu}}} \subset \lambda^I$  then
27:                  $[\underline{V}, \bar{V}] \leftarrow V^I(x^{I_{\text{neu}}})$ 
28:                 if  $V < \underline{c}$  then
29:                      $\underline{c} = V$ 
30:                 end if
31:             end if
32:             sortiere  $(\underline{V}, x^{I_{\text{neu}}}, \lambda^{I_{\text{neu}}})$  zurück in  $\mathcal{L}$ 
33:         end if
34:     end if
35:     if  $\underline{V}_{-1} > \max(\underline{c}_{\text{init}})$  then
36:          $\underline{c} = \underline{V}_{-1}$ 
37:     end if
38: end for
39: end while

```



# Literaturverzeichnis

- [1] ADAMY, J.: *Nichtlineare Regelungen*. Springer-Verlag Berlin Heidelberg, 2009. – ISBN 9783642007934
- [2] ALEFELD, G.: Inclusion methods for systems of nonlinear equations. In: HERZBERGER, J. (Hrsg.): *Topics in validated computations*. Elsevier, Amsterdam, 1994 – ISBN 9780444816856, S. 7–24
- [3] ALEFELD, G. ; HERZBERGER, J.: *Einführung in die Intervallrechnung*. Bibliographisches Institut, Mannheim, 1974. – ISBN 9783411014668
- [4] ALEFELD, G. ; HERZBERGER, J.: *Introduction to interval computations*. Academic Press, New York, 1983. – ISBN 9780120498208
- [5] AMDAHL, G.M.: Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities. In: *Proceedings of the AFIPS spring joint computer conference*, 1967, S. 483–485
- [6] ARGONNE NATIONAL LIBRARY: *mpich2*. – <http://www.mcs.anl.gov/research/projects/mpich2/index.php> (zuletzt überprüft am 2. Januar 2012)
- [7] BERTSEKAS, D.P.: *Nonlinear Programming*. Athena Scientific, 1995 (Optimization and neural computation series). – ISBN 9781886529144
- [8] BOCHNAK, J. ; COSTE, M. ; ROY, M.-F.: *Real algebraic geometry*. Springer-Verlag Berlin Heidelberg, 1998. – ISBN 9783642084294
- [9] BORNEMANN, F. ; LAURIE, D. ; WAGON, S. ; WALDVOGEL, J.: *Vom Lösen numerischer Probleme: Ein Streifzug entlang der SIAM 10x10-Digit Challenge*. Springer-Verlag Berlin Heidelberg, 2006. – ISBN 9783540341154

- [10] CHESI, G.: An LMI approach to constrained optimization with homogeneous forms. In: *Systems & Control Letters* 42 (2001), Nr. 1, S. 11–19. – DOI 10.1016/S0167-6911(00)00072-4
- [11] CHESI, G.: Domain of attraction: Estimates for non-polynomial systems via LMIs. In: *Proceedings of the 16th IFAC World Congress of Automatic Control, Prague, Czech Republic, 2005*
- [12] CHESI, G.: Estimating the domain of attraction for non-polynomial systems via LMI optimizations. In: *Automatica* 45 (2009), Nr. 6, S. 1536–1541
- [13] CHESI, G.: *Domain of Attraction: Analysis and Control Via SOS Programming*. Springer London, 2011. – ISBN 9780857299581
- [14] CHESI, G. ; TESI, A. ; VICINO, A.: Computing Optimal Quadratic Lyapunov Functions for Polynomial Nonlinear Systems via LMIs. In: *Proceedings of the 15th IFAC World Congress, Barcelona, Spain, 2002*. – DOI 10.3182/20020721-6-ES-1901.00257
- [15] CHIANG, H.D. ; HIRSCH, M.W. ; WU, F.F.: Stability region of nonlinear autonomous dynamical systems. In: *IEEE Transactions on Automatic Control* 33 (1988), S. 16–27. – DOI 10.1109/9.357
- [16] CHIANG, H.D. ; THORP, J.S.: Stability regions of nonlinear dynamical systems: a constructive methodology. In: *IEEE Transactions on Automatic Control* 34 (1989), S. 1229–1241. – DOI 10.1109/9.40768
- [17] CORNELIUS, H. ; LOHNER, R.: Computing the range of values of real functions with accuracy higher than second order. In: *Computing* 33 (1984), S. 331–347. – DOI 10.1007/BF02242276
- [18] CRAINIC, T.G ; LE CUN, B. ; ROUCAIROL, C.: Parallel Branch-and-Bound Algorithms. In: TALBI, E.-G. (Hrsg.): *Parallel combinatorial optimization*. John Wiley & Sons, Hoboken, 2006. – DOI 10.1002/9780470053928.ch1 – ISBN 9780471721017, S. 1–28
- [19] DAVISON, E.J. ; KURAK, E.M.: A Computational Method for Determining Quadratic Lyapunov Functions for Non-Linear Systems. In: *Automatica* 7 (1971), S. 627–636. – DOI 10.1016/0005-1098(71)90027-6

- [20] DIDRIT, O. ; JAULIN, L.M. ; E., Kieffer: *Applied interval analysis*. Springer-Verlag London New York, 2001. – ISBN 9781852332198
- [21] DILAVER, K.F.: *Analyse der asymptotischen Stabilität nichtlinearer Systeme mit Hilfe des Satzes von Ehlich und Zeller*. Wuppertal, Bergische Universität, Diss., 2008
- [22] DU, K.: *Cluster Problem in Global Optimization Using Interval Arithmetic*, University of Southwestern Louisiana, Diss., 1994
- [23] EHLICH, H. ; ZELLER, K.: Schwankungen von Polynomen zwischen Gitterpunkten. In: *Mathematische Zeitschrift* 86 (1964), S. 41–44. – DOI 10.1007/BF01111276
- [24] FAN, Y. ; TIBKEN, B.: Estimation of the domain of attraction for polynomial systems. In: GALKOWSKI, K. (Hrsg.): *Proceedings of the 4th International Workshop on Multidimensional Systems*. Wuppertal : GMT Ges. für Medizin u. Technik, 2005 – ISBN 3981029984, S. 77–82
- [25] FÖLLINGER, Otto: *Nichtlineare Regelungen*. Oldenbourg, München and Wien, 1993. – ISBN 9783486225037
- [26] FORSMAN, K.: *Constructive commutative algebra in nonlinear control theory*. Dep. of Electrical Engineering, Linköping Univ., Linköping, 1991. – ISBN 9789178708277
- [27] GAHINET, P. ; NEMIROVSKII, A. ; LAUB, A.J. ; CHILALI, M.: The LMI control toolbox. In: *Proceedings of the 41st Conference on Decision and Control, Orlando, USA*, S. 2038–2041
- [28] GÄRTEL, U.: *Fehlerabschätzungen für vektorwertige Randwertaufgaben zweiter Ordnung, insbesondere für Probleme aus der chemischen Reaktions-Diffusionstheorie*, Universität zu Köln, Diss., 1987
- [29] GENESIO, R. ; TARTAGLIA, M. ; VICINO, A.: On the estimation of asymptotic stability regions: State of the art and new proposals. In: *IEEE Transactions on Automatic Control* 30 (1985), Nr. 8, S. 747–755. – DOI 10.1109/TAC.1985.1104057

- [30] GENNAT, M.: *Simulation und Sensitivitätsanalyse Nichtlinearer Statischer und Dynamischer Systeme*, Diss., 2007. – Der Andere Verlag, Tönning - ISBN 9783899596649
- [31] GRIEWANK, A. ; WALTHER, A.: *Evaluating derivatives: Principles and techniques of algorithmic differentiation*. Society for Industrial and Applied Mathematics, Philadelphia, 2008. – ISBN 9780898716597
- [32] GROPP, W. ; LUSK, E. ; SKJELLUM, A.: *Using MPI: Portable parallel programming with the message-passing interface*. 2. MIT Press, Cambridge, 1999. – ISBN 9780262571340
- [33] GROPP, W. ; LUSK, E. ; THAKUR, R.: *Using MPI-2: Advanced features of the message-passing interface*. Cambridge and Mass : MIT Press, 1999. – ISBN 9780262571333
- [34] GUSTAFSON, John L.: Reevaluating Amdahl's law. In: *Communications of the ACM* 31 (1988), Nr. 5, S. 532–533. – DOI 10.1145/42411.42415
- [35] HACHICHO, O.: *Stability analysis of polynomial dynamical systems with semidefinite optimization*, Diss., 2003. – Der Andere Verlag, Tönning - ISBN 9783899591372
- [36] HACHICHO, O. ; TIBKEN, B.: Estimating domains of attraction of a class of nonlinear dynamical systems with LMI methods based on the theory of moments. In: *Proceedings of the 41st Conference on Decision and Control, Las Vegas, USA* Bd. 3, IEEE Customer Service, Piscataway and NJ, 2002. – DOI 10.1109/CDC.2002.1184354 – S. 3150 – 3155
- [37] HADDAD, W.M. ; KAPILA, V. ; CHELLABOINA, V.-S.: Guaranteed Domains of Attraction for Multivariable Luré systems via open Lyapunov Surfaces. In: *International Journal of Robust and Nonlinear Control* (1997), S. 935–949. – DOI 10.1002/(SICI)1099-1239(199710)7:10<935::AID-RNC255>3.0.CO;2-B
- [38] HAHN, Wolfgang: *Stability of motion*. Springer-Verlag Berlin Heidelberg New York, 1967 (Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen mit besonderer Berücksichtigung der Anwendungsgebiete)
- [39] HANSEN, E.R.: *Global optimization using interval analysis*. Marcel Dekker Inc, New York, 1992. – ISBN 9780824786960

- [40] HANSEN, E.R. ; WALSTER, G. W.: *Global optimization using interval analysis*. 2nd edition, revised and expanded. Marcel Dekker Inc, New York, 2004. – ISBN 9780824740597
- [41] HAYES, B.: A Lucid Interval. In: *American Scientist* 91 (2003), Nr. 6, S. 484–488. – DOI 10.1511/2003.6.484
- [42] HEWIT, J.R. ; STOREY, C.: Comparison of numerical methods in stability analysis. In: *International Journal of Control* 10 (1969), Nr. 6, S. 687–701. – DOI 10.1080/00207176908905870
- [43] HURWITZ, A.: Ueber die Bedingungen, unter welchen eine Gleichung nur Wurzeln mit negativen reellen Teilen besitzt. In: *Mathematische Annalen* 46 (1895), Nr. 2, S. 273–284. – DOI 10.1007/BF01446812
- [44] JACOBI, T. ; PRESTEL, A.: Distinguished representations of strictly positive polynomials. In: *Journal für die reine und angewandte Mathematik* 532 (2001), S. 223–235
- [45] JOHANSEN, T.A.: Computation of Lyapunov Functions for Smooth Non-linear Systems using Convex Optimization. In: *Automatica* 36 (1999), S. 1617–1626
- [46] KAZAKOVA-FREHSE, N. ; FRICK, K.: The Estimation of a Robust Domain of Attraction using Gershgorin Theorem. In: *International Journal of Robust and Nonlinear Control* 8 (1998), Nr. 3, S. 295–303. – DOI 10.1002/(SICI)1099-1239(199803)8:3<295::AID-RNC317>3.0.CO;2-O
- [47] KEARFOTT, R. B. ; DU, K.: The Cluster Problem In Multivariate Global Optimization. In: *Journal of Global Optimization* 5 (1994), S. 253–265
- [48] KEARFOTT, R.B: *Rigorous global search: Continuous problems*. Kluwer Academic Publishers, Dordrecht and Boston, 1996. – ISBN 9780792342380
- [49] KEARFOTT, R.B. ; DU, K.: The Cluster Problem in Global Optimization: The Univariate case. In: *Computing (Suppl.)* 9 (1992), S. 117–127
- [50] KHALIL, H.K.: *Nonlinear systems*. 3. ed. Prentice Hall, Upper Saddle River and NJ, 2002. – ISBN 0131227408

- [51] KLATTE, R. ; KULISCH, U. ; NEAGA, M. ; RATZ, D. ; ULLRICH, C.: *PASCAL-XSC: Sprachbeschreibung mit Beispielen*. Springer-Verlag Berlin, 1991. – ISBN 9783540537144
- [52] KRÄMER, W.: *intpakX - An Interval Arithmetic Package for Maple*. In: *Proceedings of the 12th GAMM SCAN 2006, Duisburg, Germany, 2006*
- [53] LASALLE, J.P. ; LEFSCHETZ, S.: *Die Stabilitätstheorie von Ljapunow: Die direkte Methode mit Anwendungen*. Bibliographisches Institut, Mannheim, 1967 (BI-Hochschultaschenbücher)
- [54] LASSERRE, J.B.: *Global Optimization with Polynomials and the Problem of Moments*. In: *SIAM Journal on Optimization* 11 (2001), S. 796–817. – DOI 10.1137/S1052623400366802
- [55] LERCH, M. ; TISCHLER, G. ; GUDENBERG, W.J. von ; HOFSCHESTER, W. ; KRÄMER, W.: *filib++, a Fast Interval Library Supporting Containment Computations*. In: *ACM Transactions on Mathematical Software* 32 (2006), Nr. 2, S. 299–324. – DOI 10.1145/1141885.1141893
- [56] LEVIN, A.: *An analytical method of estimating the domain of attraction for polynomial differential equations*. In: *IEEE Transactions on Automatic Control* 39 (1994), Nr. 12, S. 2471–2475. – DOI 10.1109/9.362845
- [57] LI, C. ; CAO, C. ; LI, N. ; CHEN, X.: *Estimating the domain of attraction of a class of SIR epidemic model*. In: *Proceedings of the Chinese Control and Decision Conference, Yantai, China, 2008*. – DOI 10.1109/CCDC.2008.4598283 – S. 5010–5013
- [58] LIAW, C.-H. D.-C .and L. D.-C .and Lee: *An Approach to Estimate Domain of Attraction for Nonlinear Control Systems*. In: *Proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC), Beijing, China, 2006*. – DOI 10.1109/ICICIC.2006.41 – S. 146–149
- [59] LJAPUNOV, A.M.: *The general problem of the stability of motion: engl. Übersetzung der russischen Dissertation aus dem Jahr 1892*. Taylor & Francis, London, 1992. – ISBN 9780748400621
- [60] LOH, E. ; WALSTER, G.W.: *Rump's Example Revisited*. In: *Reliable Computing* 8 (2002), S. 245–248. – DOI 10.1023/A:1015569431383

- [61] LUNZE, J.: *Automatisierungstechnik: Methoden für die Überwachung und Steuerung kontinuierlicher und ereignisdiskreter Systeme*. 2. Auflage. Oldenbourg, München, 2007. – ISBN 9783486580617
- [62] LUNZE, J.: *Regelungstechnik 1: Systemtheoretische Grundlagen, Analyse und Entwurf einschleifiger Regelungen*. 8., neu bearbeitete Aufl. Springer-Verlag, Berlin, 2010. – ISBN 3642138071
- [63] MAGNUS, K.: Zur Entwicklung des Stabilitätsbegriffes in der Mechanik. In: *Naturwissenschaften* 46 (1959), S. 590–595. – DOI 10.1007/BF00684205
- [64] MAYER, S.: *Aufbau und Evaluation eines Parallelrechners zur Ausführung von Optimierungsalgorithmen*, Bergische Universität Wuppertal, Diplomarbeit, 2010
- [65] MAYER, S. ; TIBKEN, B. ; WARTHENPFUHL, S.: Parallel Computation of Domains of Attraction for Nonlinear Dynamic Systems. In: *Proceedings of the 6th International Symposium on Parallel Computing in Electrical Engineering (PARELEC), Luton, UK, 2011*. – DOI 10.1109/PARELEC.2011.19 – S. 87–92
- [66] MICHEL, A.N. ; SARABUDLA, N. R. ; MILLER, R.K.: Stability analysis of complex dynamical systems. In: *Circuits, Systems, and Signal Processing* 1 (1982), Nr. 2, S. 171–202. – DOI 10.1007/BF01600051
- [67] MOLKENTIN, D. ; PÖNITZ, A.: *Qt 4: Einführung in die Applikationsentwicklung*. 2. Open Source Press, München, 2010. – ISBN 3937514996
- [68] MOON, Y.-H. ; CHOI, B.-K. ; ROH, T.-H.: Estimating the Domain of Attraction for Power systems via a Group of Damping-Reflected Energy Functions. In: *Automatica* 36 (2000), Nr. 3, S. 419–425. – DOI 10.1016/S0005-1098(99)00162-4
- [69] MOORE, R.E.: *Interval analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1966
- [70] MOORE, R.E ; KEARFOTT, R.B. ; CLOUD, M.J.: *Introduction to interval analysis*. Society for Industrial and Applied Mathematics, Philadelphia, 2009. – ISBN 0898716691
- [71] NEMIROVSKI, A. ; GAHINET, P.: *LMI Control Toolbox: The LMI Lab Version 1.0*. The MathWorks Inc., South Natick, USA, 1995
- [72] NYQUIST, H.: Regeneration theory. In: *Bell System Technical Journal* 11 (1932), S. 126–147

- [73] PARILLO, P.: *Structured Semidefinite Programs and Semialgebraic Geometry Methods in Robustness and Optimization*, California Institute of Technology, Pasadena, USA, Diss., 2000
- [74] QT-PROJECT: *Qt*. – <http://qt.nokia.com/> (zuletzt überprüft am 2. Januar 2012)
- [75] ROUTH, E.J.: *Dynamics of a System of Rigid Bodies*. Macmillan and co, 1860
- [76] ROUTH, E.J.: *A Treatise on the Stability of a Given State of Motion, Particularly Steady Motion*. Macmillan and co, 1877
- [77] RUMP, S.M.: Algebraic Computation, Numerical Computation and Verified Inclusions. In: JANSSEN, Rainer (Hrsg.): *Trends in Computer Algebra*. Springer-Verlag Berlin Heidelberg, 1988 – ISBN 3540189289, S. 177–197
- [78] RUMP, S.M.: INTLAB - INTerval LABoratory. In: CSENDES, Tibor (Hrsg.): *Developments in Reliable Computing*. Kluwer Academic Publishers Dordrecht, 1999, S. 77–104. – <http://www.ti3.tu-harburg.de/rump/> (zuletzt überprüft am 2. Januar 2012)
- [79] RUTTMANN, B.: *Untersuchungen zur Fehlerabschätzung von polynomialen Näherungslösungen bei Randwertaufgaben gewöhnlicher Differentialgleichungen*, Universität zu Köln, Diss., 1982
- [80] SALEME, A. ; TIBKEN, B. ; WARTHENPFUHL, S. ; SELBACH, C.: Estimation of the Domain of Attraction for Non-Polynomial Systems: A Novel Method. In: *Proceedings of the 18th IFAC World Congress, Milano, Italy, 2011*. – DOI 10.3182/20110828-6-IT-1002.01450 – S. 10976–10981
- [81] SCHEIDERER, C.: Distinguished Representations of Non-Negative Polynomials. In: *Journal of Algebra* 289 (2005), S. 558–573. – DOI 10.1016/j.jalgebra.2005.01.043
- [82] SCHICHL, H. ; NEUMAIER, A.: Exclusion Regions for Systems of Equations. In: *SIAM Journal on Numerical Analysis* 42 (2005), Nr. 1, S. 383–408. – DOI 10.1137/S0036142902418898
- [83] SIMON, M.: *Zur Stabilität dynamischer Systeme mit stochastischer Anregung*. Univ.-Verl. Karlsruhe, 2004. – ISBN 9783937300139



- [84] SKELBOE, S.: Computation of rational interval functions. In: *BIT* 14 (1974), Nr. 1, S. 87–95. – DOI 10.1007/BF01933121
- [85] STAUNING, O.: *Introduction to FADBAD, a C++ Program package for Automatic Differentiation*. Folien eines Vortrags des Kurses „C0202 – Topics in Numerical Analysis“ am 11.03.1997 an der Technical University of Denmark, 1997. – <http://www.imm.dtu.dk/~Ekm/FADBAD/adiffstalk0202.pdf>, zuletzt überprüft am 2. Januar 2012
- [86] STURM, J.F.: Using SeDuMi 1.02, A MATLAB toolbox for optimization over symmetric cones. In: *Optimization Methods and Software* 11 (1999), Nr. Special issue on Interior Point Methods (CD supplement with software), S. 625–653. – DOI 10.1080/10556789908805766
- [87] SUSE LINUX GMBH: *OpenSuSE*. – <http://de.opensuse.org> (zuletzt überprüft am 2. Januar 2012)
- [88] TAN, W. ; PACKARD, A.: Stability Region Analysis Using Polynomial and Composite Polynomial Lyapunov Functions and Sum-of-Squares Programming. In: *IEEE Transactions on Automatic Control* 53 (2008), Nr. 2, S. 565–571. – DOI 10.1109/TAC.2007.914221
- [89] TANENBAUM, A.S: *Moderne Betriebssysteme*. 3. Pearson Studium, München und Boston, 2009. – ISBN 3827373425
- [90] TESI, A. ; VILLORESI, F. ; GENESIO, R.: On the stability domain estimation via a quadratic Lyapunov function: convexity and optimality properties for polynomial systems. In: *IEEE Transactions on Automatic Control* 41 (1996), Nr. 11, S. 1650–1657. – DOI 10.1109/9.544002
- [91] THE MATHWORKS INC., USA Natick: *MATLAB Version 7.11.0 (R2010b)*. 2010. – <http://www.mathworks.de/products/matlab/index.html> (zuletzt überprüft am 2. Januar 2012)
- [92] TIBKEN, B.: Estimation of the domain of attraction for polynomial systems via LMIs. In: *Proceedings of the 39th Conference on Decision and Control, Sydney, Australia* Bd. 4, IEEE Customer Service, Piscataway and NJ, 2000. – DOI 10.1109/CDC.2000.912314 – S. 3860–3864

- [93] TIBKEN, B. ; DILAVER, K.F.: Computation of subsets of the domain of attraction for polynomial systems. In: *Proceedings of the 41st Conference on Decision and Control, Las Vegas, USA* Bd. 3, IEEE Customer Service, Piscataway and NJ, 2002. – DOI 10.1109/CDC.2002.1184239 – S. 2651–2656
- [94] TIBKEN, B. ; FAN, Y.: Computing the Domain of Attraction for Polynomial Systems via BMI Optimization Method. In: *Proceedings of the 25th American Control Conference, Minneapolis, USA*, IEEE Customer Service, Piscataway and NJ. – DOI 10.1109/ACC.2006.1655340 – S. 117–122
- [95] TIBKEN, B. ; HACHICHO, O.: Estimation of the domain of attraction for polynomial systems using multidimensional grids. In: *Proceedings of the 39th Conference on Decision and Control, Sydney, Australia* Bd. 4, IEEE Customer Service, Piscataway, 2000. – DOI 10.1109/CDC.2000.912316 – S. 3870–3874
- [96] TIBKEN, B. ; HOFER, E.P. ; DEMIR, C.: Guaranteed regions of attraction for dynamical polynomial systems. In: LEITMANN, George (Hrsg.) ; UDWADIA, Firdaus E. (Hrsg.): *Dynamics and control* Bd. 9. Gordon & Breach and Gordon and Breach Science Publ., Amsterdam, 1999 – ISBN 9789056991722
- [97] TOPCU, U. ; PACKARD, A. ; SEILER, P. ; BALAS, G.: Help on SOS. In: *IEEE Control Systems Magazine* 30 (2010), Nr. 4, S. 18–23. – DOI 10.1109/MCS.2010.937045
- [98] UNITED STATES GENERAL ACCOUNTING OFFICE: *Patriot Missile Defense: Software Problem Led to System Failure at Dhahran, Saudi Arabia*. 1992. – <http://archive.gao.gov/t2pbat6/145960.pdf> (zuletzt überprüft am 2. Januar 2012)
- [99] VANNELLI, A. ; VIDYASAGAR, M.: Maximal Lyapunov functions and domains of attraction for autonomous nonlinear systems. In: *Automatica* 21 (1985), Nr. 1, S. 69–80. – DOI 10.1016/0005-1098(85)90099-8
- [100] WARTHENPFUHL, S. ; TIBKEN, B. ; MAYER, S.: An interval arithmetic approach for the estimation of the domain of attraction. In: *Proceedings of the 2010 IEEE International Symposium on Computer-Aided Control System Design (CACSD), Yokohama, Japan*, 2010. – DOI 10.1109/CACSD.2010.5612692 – S. 1999–2004

- [101] WEIERSTRASS, K.: Über die analytische Darstellbarkeit sogenannter willkürlicher Functionen einer reellen Veränderlichen. In: *Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften zu Berlin*. 1885, S. 633–639
- [102] ZUBOV, V.I.: Questions of the theory of Lyapunov's second method: Construction of the general solution in the region of asymptotical stability. In: *Prikl. Mat. Meh.* 19 (1955), S. 179–210