

UNIVERSITÄT
BAYREUTH

Der Hamilton-Jacobi-Bellman-Ansatz zur Berechnung von Erreichbarkeitsmengen

Diplomarbeit

von

Matthias Rodler

FAKULTÄT FÜR MATHEMATIK UND PHYSIK
MATHEMATISCHES INSTITUT

vorgelegt am: 31. Juli 2014

Betreuung:
Prof. Dr. Lars Grüne
AOR Dr. Robert Baier

Danksagung

Die ersten Zeilen meiner Diplomarbeit möchte ich nutzen, um mich bei den Personen zu bedanken, die zum Gelingen dieser Arbeit beigetragen haben.

Zunächst möchte ich Herrn Prof. Dr. Grüne dafür danken, dass er mir die Möglichkeit gegeben hat, diese Arbeit anfertigen zu können.

Mein ganz besonderer Dank gilt Herrn Dr. Robert Baier für seine immerwährende und sehr zuvorkommende Unterstützung und herausragende Hilfsbereitschaft bei der Anfertigung dieser Arbeit. Die Diskussionen in den begleitenden Besprechungen waren immer sehr inspirierend und produktiv.

Desweiteren bedanke ich mich bei Herrn Prof. Bokanowski, Laboratoire Jacques Louis Lions, Université Paris Diderot (Paris 7) et Ensta-Paris Tech, für die Bereitstellung der ITN-SADCO-Software und für die ausgiebige Hilfe und Anleitung bei der Verwendung des „ROC-HJ-Lösers“.

Inhaltsverzeichnis

Notations- und Abkürzungsverzeichnis	IV
1 Einleitung	1
1.1 Motivation	1
1.2 Gliederung	2
2 Grundlagen dyn. Systeme und nichtglatter Analysis	3
2.1 Kontrollsysteme	3
2.2 Subdifferential und Subgradient	10
3 Wertefunktion und dynamische Programmierung	17
3.1 Einführung der Wertefunktion	17
3.2 Grundlegende Eigenschaften der Wertefunktion	20
3.3 Dynamische Programmierung	23
4 Hamilton-Jacobi-Ansatz	26
4.1 Hamilton-Jacobi-Ansatz mit glatter Wertefunktion	26
4.1.1 Existenz der Lösung der Hamilton-Jacobi-Gleichung	26
4.1.2 Eindeutigkeit der Wertefunktion als klassische Lösung der Hamilton-Jacobi-Gleichung	30
4.2 Hamilton-Jacobi-Ansatz mit nichtglatter Wertefunktion	31
4.2.1 Viskositätslösungen	33
4.2.2 Eindeutigkeit der Wertefunktion als Viskositätslösung der Hamilton-Jacobi-Gleichung	36
4.3 Zusammenhang mit Erreichbarkeitsmengen	42
5 Numerische Berechnung von Erreichbarkeitsmengen	43
5.1 Verwendetes Verfahren	43
5.2 Beispiel 1	45
5.3 Beispiel 2	52
5.4 Beispiel 3	65
6 Verwendete Software: ROC-HJ-Löser	69
6.1 Überblick	69
6.2 Problemformulierung	69
6.3 Kompilation und Ausführung	71
6.4 Vorstellung des Programms	72
7 Zusammenfassung	84
A Anhang	85
Abbildungsverzeichnis	86

<i>INHALTSVERZEICHNIS</i>	III
Tabellenverzeichnis	88
Literatur	89

Notations- und Abkürzungsverzeichnis

\mathbb{R}^n	der euklidische n -dimensionale Raum
$ x $	die euklidische Norm von $x \in \mathbb{R}^n$
$\dot{y}(t)$	Ableitung der Funktion y nach t
B	offener Einheitsball im euklidischen Raum
\bar{B}	Abgeschlossener Einheitsball im euklidischen Raum
$B(x, r)$	offener Ball mit Radius r um Punkt x im euklidischen Raum
$C(I; O)$	alle stetigen Funktionen $f : I \rightarrow O$
$C^n(I; O)$	alle n -mal stetig differenzierbaren Funktionen $f : I \rightarrow O$
$\mathcal{W}^{1,1}(I; O)$	alle absolut stetigen Funktionen $f : I \rightarrow O$
$y(t; t_0, x_0, u(\cdot))$	Lösung der Differentialgleichung $\dot{y}(t) = f(t, y(t), u(t))$ mit Startzeitpunkt t_0 und Anfangswert x_0 und Kontrolle $u(\cdot)$
$y_{t_0, x_0}^u(t)$	Kurzschreibweise der Lösung $y(t; t_0, x_0, u(\cdot))$
$y_{t_0, x_0}(t)$	Alternative Schreibweise der Lösung $y_{t_0, x_0}^u(t)$, falls keine Gefahr der Verwechslung für $u(\cdot)$ existiert.
$y_{x_0}^u(t)$	Kurzschreibweise der Lösung $y(t; t_0, x_0, u(\cdot))$ mit $t_0 = 0$
$y_{x_0}(t)$	Alternative Schreibweise der Lösung $y_{x_0}^u(t)$, falls keine Gefahr der Verwechslung für $u(\cdot)$ existiert.
$y[s]$	Alternative Kurzschreibweise der Lösung $y(s; t, x, u(\cdot))$, falls keine Gefahr der Verwechslung für Anfangszeitpunkt, Anfangswert oder Kontrolle existiert.
$s \wedge t$	$\min\{s, t\}$ für $s, t \in \mathbb{R}$
$s \vee t$	$\max\{s, t\}$ für $s, t \in \mathbb{R}$
$y_{s \wedge t, (x p)}(\cdot)$	$\begin{cases} y_{s,p}(\cdot) & \text{falls } s \wedge t = s \\ y_{t,x}(\cdot) & \text{falls } s \wedge t = t \end{cases}$ für $y_{s,p}(\cdot)$ und $y_{t,x}(\cdot)$ gegeben.
$y_{s \vee t, (x p)}(\cdot)$	$\begin{cases} y_{s,p}(\cdot) & \text{falls } s \vee t = s \\ y_{t,x}(\cdot) & \text{falls } s \vee t = t \end{cases}$ für $y_{s,p}(\cdot)$ und $y_{t,x}(\cdot)$ gegeben.

$\mathcal{R}(t; t_0, x_0)$	Erreichbare Menge zum Zeitpunkt t mit Startzeitpunkt t_0 und Anfangswert x_0 .
ω	Modulus, d.h. eine stetige, nicht fallende Funktion $\omega : [0, +\infty[\rightarrow [0, +\infty[$ mit $\omega(0) = 0$; allg: eine Funktion $\omega : [0, +\infty[^2 \rightarrow [0, +\infty[$, so dass vorhergenannte Eigenschaften für alle $R > 0$ gelten.
$D^+w(x_0)$	das <i>Fréchet-Superdifferential</i> von w im Punkt x_0
$D^-w(x_0)$	das <i>Fréchet-Subdifferential</i> von w im Punkt x_0

1 Einleitung

1.1 Motivation

Das Ziel dieser Diplomarbeit ist die Untersuchung der Berechnung von Erreichbarkeitsmengen von optimalen Steuerungsproblemen mit Hilfe des Hamilton-Jacobi-Ansatzes. Optimale Steuerungsprobleme, auch Kontrollprobleme genannt, basieren auf gewöhnlichen Differentialgleichungen (ODE = Ordinary Differential Equation) der Form

$$\frac{dy}{dt} = f(t, y(t)), \quad (1.1)$$

wobei t meist als Zeit interpretiert wird und $x(t)$ als Zustand zum Zeitpunkt t bezeichnet wird. Die gewöhnlichen Differentialgleichungen stellen ein wichtiges Werkzeug der Mathematik dar, mit dem die zeitliche Entwicklung vieler Systeme in der Natur, Technik und Gesellschaft modelliert werden kann. Differentialgleichungen der Form (1.1) können auch durch weitere Parameter bzw. Funktionen erweitert werden. In Kontrollsystemen dient hierzu die Kontrolle u bzw. die Kontrollfunktion $u(t)$. Damit lautet die zu betrachtende Differentialgleichung

$$\frac{dy}{dt} = f(t, y(t), u(t)). \quad (1.2)$$

Je nach Problemstellung lässt sich die Kontrolle sowohl als Steuerung als auch als Störung des Systems interpretieren. Kontrollsysteme und deren Lösung und Lösbarkeitsbegriff stellen ein wichtiges und weitreichendes Gebiet aktueller mathematischer Forschung dar, da sie zur Modellierung vieler Abläufe und Phänomene dienen können. Zu diesen zählen beispielsweise die Kollisionsvermeidung unter anderem im Straßen- oder auch im Flugverkehr, die Berechnung von Umlaufbahnen im Weltall und die Betrachtung von Bevölkerungs- oder auch Wetter- und Klimamodellen.

Einen wichtigen Aspekt bei der Lösung von Kontrollproblemen stellt die Berechnung erreichbarer Mengen dar. Erreichbare Mengen (engl. Reachable Sets)

$$\mathcal{R}(t; t_0, x_0) := \bigcup \{y(t; t_0, x_0) \mid x(\cdot) \text{ zulässig für (1.2)}\} \quad (1.3)$$

lassen sich über verschiedene Ansätze bestimmen, wie zum Beispiel mittels Distanzfunktionen und mengenwertiger Verfahren (siehe [Rod10b]) oder auch mit Hilfe des Hamilton-Jacobi-Bellman-Ansatzes (siehe auch [Rod10a]). Auf diesem liegt das Hauptaugenmerk dieser Arbeit. Der Hamilton-Jacobi-Ansatz verwendet die sog. Hamilton-Jacobi-Gleichung zur Lösung von Kontrollproblemen und Bestimmung der Wertefunktion. Die Hamilton-Jacobi-Gleichung ist eine partielle nichtlineare Differential-Gleichung 1. Ordnung. Diese Arbeit konzentriert sich hierbei auf Viskositätslösungen der HJB-Gleichung. Bei Viskositätslösungen handelt es sich um eine erweiterte, schwächere Form der Lösung, die dazu nötig ist, um die Voraussetzung der Differenzierbarkeit der Lösung auf eine fast-überall-Formulierung abzuschwächen. Somit muss für diese Viskositätslösungen eine eigene Lösungstheorie entwickelt werden, um Existenz und Eindeutigkeit zu gewährleisten.

1.2 Gliederung

Hier in diesem ersten Kapitel wird eine kurze Einführung in das Thema dieser Diplomarbeit „Der Hamilton-Jacobi-Bellman-Ansatz zur Berechnung von Erreichbarkeitsmengen“ und eine Gliederungsübersicht gegeben.

Im zweiten, nun folgenden, Kapitel dieser Arbeit werden einige Grundbegriffe dynamischer Systeme und Grundlagen der nicht glatten Analysis aufgegriffen und formuliert. Neben Definitionen zu Kontrollsystemen und wichtigen grundlegenden Sätzen über die Existenz von Lösungen werden auch wichtige Aussagen aus der nichtglatten Analysis eingeführt. Hierzu zählt vor allem der zentrale Begriff des Subdifferentials bzw. des Subgradienten.

Anschließend werden im dritten Kapitel die Wertefunktion eines optimalen Steuerungsproblems definiert und wichtige Eigenschaften dieser geklärt. Desweiteren wird ein wesentlicher Satz zur dynamischen Programmierung formuliert und bewiesen.

Im vierten Kapitel wird schließlich die Theorie des Hamilton-Jacobi-Ansatzes vorgestellt und differenziert, bezogen auf die Fallunterscheidung zwischen glatter und nicht glatter Wertefunktion, analysiert. Hier wird auch die Existenz und Eindeutigkeit der Wertefunktion als Viskositätslösung der Hamilton-Jacobi-Gleichung bewiesen. Außerdem wird der Zusammenhang zwischen der Wertefunktion und der Erreichbarkeitsmenge geklärt.

Basierend auf dieser Theorie werden im folgenden fünften Kapitel numerische Beispiele vorgestellt, welche mit der „ROC-HJ“-Software des ITN-SADCO-Arbeitskreises um Prof. Bokanowski berechnet und visualisiert wurden. Die Software nutzt dabei den Ansatz der Hamilton-Jacobi-Gleichung.

Abschließend wird im letzten Kapitel die Software an sich vorgestellt und beschrieben. Auf diese Weise ist gewährleistet, dass der Leser dieser Arbeit die vorgestellten Beispiele auch selbst ausführen und den grundlegenden Umgang mit der Software erlernen kann.

Im Anhang schließlich erfolgt eine Beschreibung der beigelegten DVD, welche alle vorbereiteten Beispiele, sowie die verwendete Literatur und das vorgestellte Programm enthält.

2 Grundlagen dynamischer Systeme und nicht-glatte Analysis

Dieses Kapitel gibt einen Überblick über die Kontrollsysteme und dient zur Beschreibung einiger wichtiger Annahmen, welche im weiteren Verlauf dieser Arbeit eine große Rolle spielen.

2.1 Kontrollsysteme

Kontrollsysteme entstehen aus Differentialgleichungen $\dot{y}(t) = f(t, y(t))$ durch Hinzufügen eines zusätzlichen, meist zeitabhängigen, Parameters u bzw. $u(t)$. Dieser kann als Steuerung oder Störung interpretiert werden.

Definition 2.1 (Kontrollsystem)

Kontrollsysteme, die auf kontinuierlichen Zeiten definiert sind, werden allgemein durch Differentialgleichungen der Form

$$\dot{y}(t) = f(t, y(t), u(t)) \quad (2.1)$$

beschrieben, wobei

- die Variable t meist als *Zeit* interpretiert wird,
- $y(t) \in \mathbb{R}^n$ den *Zustand* bezeichnet und
- $u(\cdot) \in \mathcal{U} := \{u : \mathbb{R} \rightarrow U, u \text{ messbar}\}$ die *Kontrollfunktion*, wobei $U \subset \mathbb{R}^m$ den *Kontrollwertebereich* angibt.

Die Abbildung $f : \mathbb{R} \times \mathbb{R}^n \times U \rightarrow \mathbb{R}^n$ wird *Vektorfeld* genannt.

Es lassen sich grundlegend zwei Arten von Kontrollsystemen unterscheiden:

1. Zeitinvariante Kontrollsysteme

Diese Art von Kontrollsystemen besitzt keine explizite Zeitabhängigkeit. Daher kann (2.1) durch Unterdrückung des 1. Arguments t vereinfacht werden:

$$\dot{y}(t) = f(y(t), u(t)) \quad (2.2)$$

2. Zeitvariante Kontrollsysteme

Vektorfelder dieser Kontrollsysteme besitzen eine explizite Zeitabhängigkeit im 1. Argument. Daher kann die allgemeine Differentialgleichung (2.1) nicht vereinfacht werden.

Bemerkung 2.2

In dieser Arbeit werden allgemeine Kontrollsysteme mit expliziter Zeitabhängigkeit (2.1) untersucht. In der Literatur werden zur Vereinfachung oft nur zeitinvariante

Kontrollsysteme betrachtet, wobei jedoch die Zeitabhängigkeit durch eine Koordinatentransformation berücksichtigt werden kann. Im Hinblick auf die Transformation bzgl. der Hamilton-Jacobi-Gleichung, welche im weiteren Verlauf der Arbeit eine wichtige Rolle spielen wird, sei hierbei auf [BCD08, S. 26] verwiesen.

Zur Lösbarkeit der betrachteten Differentialgleichung und zur Konkretisierung der Problemstellung empfiehlt es sich einige Annahmen zu treffen, welche im Folgenden erläutert werden.

Für das Kontrollsystem werden für diese Arbeit folgende Annahmen getroffen. Hierbei orientieren wir uns an [BCD08, Kap. 3, S. 97]:

Annahme 2.3

$$\begin{aligned}
 (A_0) \quad & \left\{ \begin{array}{l} U \subset \mathbb{R}^m, U \text{ kompakt} \\ f : [0, T] \times \mathbb{R}^n \times U \rightarrow \mathbb{R}^n \text{ ist stetig} \end{array} \right. \\
 (A_1) \quad & \left\{ \begin{array}{l} f \text{ ist beschränkt auf } [0, T] \times B(0, R) \times U \quad \forall R > 0, \\ \text{d.h. } |f(t, x, u)| \leq M_f(R); \\ \forall t \in [0, T], \forall x \in B(0, R), \forall u \in U, \forall R > 0 \end{array} \right. \\
 (A_2) \quad & \left\{ \begin{array}{l} |f(t, y, u) - f(s, x, u)| \leq \omega_f(|t - s| + |x - y|, R) \\ \forall t, s \in [0, T], x, y \in B(0, R), \forall R > 0 \end{array} \right. \\
 (A_3) \quad & \left\{ \begin{array}{l} |f(t, x, u) - f(t, y, u)| \leq L_f(R) |x - y| \\ \forall t \in [0, T], x, y \in B(0, R), \forall R > 0, u \in U \end{array} \right.
 \end{aligned}$$

Bemerkung 2.4

Betrachtet man zeitinvariante Kontrollsysteme, lässt sich aus der häufig verwendeten Bedingung der lokalen Lipschitzstetigkeit des Vektorfelds in der Zustandsvariable (A_3) direkt die Bedingung (A_2) unter Verwendung des Modulus $w_f(r, R) = Lr$ herleiten.

Die Annahmen 2.3 stellen sicher, dass die Differentialgleichung

$$\dot{y}(t) = f(t, y(t), u(t))$$

mit Anfangswert $y(t_0) = x_0$ für jede beliebige Kontrollfunktion $u(\cdot) \in \mathcal{U}$ eine eindeutige Lösung $y(\cdot)$ auf dem betrachteten Zeitintervall $[t_0, T]$ mit Anfangszeitpunkt t_0 und Anfangswert $x_0 \in \mathbb{R}^n$ besitzt. Der Kontrollraum \mathcal{U} wird hier als die Menge der messbaren Funktionen $u(\cdot)$ angenommen.

Dies bedeutet, dass $y(\cdot)$ die Lösung der Integralgleichung

$$y(t) = x_0 + \int_{t_0}^t f(s, y(s), u(s)) ds, \quad t > t_0, \quad (2.3)$$

darstellt, in dem Sinne, dass $y(\cdot)$ im Besonderen absolut stetig auf kompakten Teilintervallen von $[t_0, \infty[$ ist und es (2.1) fast überall löst.

Die Lösung $y(t; t_0, x_0, u(\cdot))$ zum Anfangszeitpunkt t_0 , mit Startwert x_0 und Kontrolle $u(\cdot)$ wird im Weiteren kurz als $y_{t_0, x_0}^u(t)$ oder bei Verwendung des Startzeitpunktes $t_0 = 0$ als $y_{x_0}^u(t)$ geschrieben. Wenn Missverständnisse ausgeschlossen sind, beispielsweise wenn lediglich eine eindeutige Kontrolle verwendet wird, wird auch $y_{t_0, x_0}(t)$ oder $y_{x_0}(t)$ geschrieben.

Im Folgenden werden einige wichtige Resultate erläutert, welche im Verlauf dieser Arbeit mehrmals Verwendung finden und daher im Vorgriff zusammengefasst werden.

Begonnen wird an dieser Stelle mit dem Gronwall-Lemma, welches es ermöglicht, aus einer Integralungleichung explizite Schranken zu erhalten [BCD08, Kap. III, 5. Appendix, S. 218ff.].

Satz 2.5 Gronwall-Lemma

Seien $\omega \in L^1([t_0, T])$ und $h \in L^\infty([t_0, T])$ mit

$$\omega(t) \leq h(t) + L \int_{t_0}^t \omega(s) ds \quad \text{für fast alle } t \in [t_0, T] \text{ und ein } L \geq 0. \quad (2.4)$$

Dann gilt:

$$\omega(t) \leq h(t) + Le^{Lt} \int_{t_0}^t h(s)e^{-Ls} ds \quad \text{für fast alle } t \in [t_0, T]. \quad (2.5)$$

Falls zusätzlich h nicht fällt ($h'(t) \geq 0$) gilt:

$$\omega(t) \leq h(t)e^{L(t-t_0)} \quad \text{für fast alle } t \in [t_0, T]. \quad (2.6)$$

Bemerkung 2.6

Die lokale Beschränktheit von f (A_1), gleichmäßig in der Kontrollvariable u , muss zum Erhalt eines Vergleichsprinzips und des Eindeutigkeitssatzes durch den Begriff des linearen Wachstums (A'_1) verschärft werden. Es muss gelten:

$$(A'_1) \quad \left\{ \begin{array}{l} f \text{ besitzt lineares Wachstum auf } [0, T] \times \mathbb{R}^n \times U, \\ \text{d.h. } \exists C > 0, \text{ sd. } |f(t, x, u)| \leq C(1 + |x|) \\ \forall t \in [0, T], \forall x \in \mathbb{R}^n, \forall u \in U. \end{array} \right.$$

In zeitinvarianten Betrachtungen ist diese Verschärfung von (A_1) nicht notwendig, daher wurde am Anfang zunächst (A_1) anlehnend an [BCD08, Kap. 3, S. 97] explizit eingeführt. Für dieses Thema ist (A_1) jedoch zu schwach und wir werden daher ausschließlich (A'_1) verwenden.

Mit Hilfe des Gronwall-Lemmas und der verschärften Annahme (A'_1) lassen sich im Folgenden Aussagen über die Beschränktheit der Lösungen in Abhängigkeit von Anfangswert und betrachteter Intervalllänge herleiten.

Folgerung 2.7

Sei $y_{\tau,x}(\cdot) := y(\cdot; \tau, x, u(\cdot))$ die Lösung von

$$\begin{aligned} y'(s) &= f(s, y(s), u(s)) \\ y(\tau) &= x \end{aligned}$$

und es gelte (A_0) und (A'_1) .

Dann gilt für alle $\tau \in [0, T]$:

$$|y_{\tau,x}(t) - x| \leq \tilde{R}(t; \tau, x), \quad \forall t \in [\tau; T] \quad (2.7)$$

$$|y_{\tau,x}(t)| \leq R(T, x), \quad \forall t \in [\tau; T] \quad (2.8)$$

wobei

$$\begin{aligned} \tilde{R}(s; t_0, x_0) &:= C(s - t_0)(1 + |x_0|)e^{C(s-t_0)} \text{ und} \\ R(s, x_0) &:= Cse^{Cs} + (Cse^{Cs} + 1)|x_0|. \end{aligned}$$

Beweis:

Betrachte die Differenz zwischen der Lösungstrajektorie $y_{\tau,x}(t)$ zum Zeitpunkt t und dem Startwert x (OBdA sei $t > \tau$):

$$\begin{aligned} |y_{\tau,x}(t) - x| &= \left| x + \int_{\tau}^t f(s, y_{\tau,x}(s), u(s)) ds - x \right| = \left| \int_{\tau}^t f(s, y_{\tau,x}(s), u(s)) ds \right| \\ &\leq \int_{\tau}^t \underbrace{|f(s, y_{\tau,x}(s), u(s))|}_{\substack{(A'_1) \\ \leq C(1+|y_{\tau,x}(s)|)}} ds \\ &\leq \int_{\tau}^t C(1 + |y_{\tau,x}(s)|) ds \\ &= \int_{\tau}^t C(1 + |y_{\tau,x}(s) - x + x|) ds \\ &\leq \int_{\tau}^t C(1 + |y_{\tau,x}(s) - x| + |x|) ds \\ &= \int_{\tau}^t C(1 + |x|) ds + \int_{\tau}^t C |y_{\tau,x}(s) - x| ds \end{aligned}$$

$$= C(t - \tau)(1 + |x|) + C \int_{\tau}^t |y_{\tau,x}(s) - x| ds.$$

Somit gilt also

$$\underbrace{|y_{\tau,x}(t) - x|}_{=:w(t)} \leq \underbrace{C(t - \tau)(1 + |x|)}_{=:h(t) \text{ (nicht fallend)}} + C \int_{\tau}^t \underbrace{|y_{\tau,x}(s) - x|}_{w(s)} ds.$$

Mit dem Gronwall-Lemma (Satz 2.5) folgt

$$|y_{\tau,x}(t) - x| \leq C(t - \tau)(1 + |x|)e^{C(t-\tau)} =: \tilde{R}(t; \tau, x).$$

Zudem gilt:

$$\begin{aligned} |y_{\tau,x}(t)| &= |y_{\tau,x}(t) - x + x| \leq |y_{\tau,x}(t) - x| + |x| \\ &\leq C(t - \tau)(1 + |x|)e^{C(t-\tau)} + |x| &&= \tilde{R}(t; \tau, x) + |x| \\ &= C(t - \tau)e^{C(t-\tau)} + (C(t - \tau)e^{C(t-\tau)} + 1) |x| \\ &\leq CT e^{CT} + (CT e^{CT} + 1) |x| &&=: R(T, x). \end{aligned}$$

□

Mit der Aussage der Folgerung 2.7 erhalten wir nun unter Anwendung des Gronwall-Lemmas (Satz 2.5) Abschätzungen für die Lösungstrajektorien, falls jeweils nur der Anfangszeitpunkt bzw. der Anfangswert variiert wird.

Lemma 2.8

- (i) Seien (A_0) , (A'_1) und (A_3) erfüllt. Dann gilt für je zwei Lösungen mit Anfangswert x und Anfangszeitpunkten t_0 und τ die folgende Aussage:
Es existiert L_y , so dass für alle $t \in [0, T]$ gilt:

$$|y_{t_0,x}(t) - y_{\tau,x}(t)| \leq L_y(1 + |x|) |t_0 - \tau|. \quad (2.9)$$

- (ii) Seien (A_0) , (A'_1) und (A_3) erfüllt. Dann gilt für je zwei Lösungen mit Anfangszeitpunkt t_0 und Anfangswerten x und z :

$$|y_{t_0,x}(t) - y_{t_0,z}(t)| \leq e^{L_f(R)(t-t_0)} |x - z|, \quad (2.10)$$

für alle $t \in [t_0, T]$ und alle $u \in \mathcal{U}$.

Beweis:

- (i) Seien $y_{t_0,x}(t)$ und $y_{\tau,x}(t)$ zwei Lösungstrajektorien mit unterschiedlichen Anfangszeitpunkten t_0 und τ , jedoch mit gleichem Anfangswert x , ausgewertet zum Zeitpunkt t (OBdA sei $t \geq t_0 \geq \tau$):

$$\begin{aligned}
 y_{t_0,x}(t) &:= y(t; t_0, x, u(\cdot)) \stackrel{(2.3)}{=} x + \int_{t_0}^t f(s, y_{t_0,x}(s), u(s)) ds, \\
 y_{\tau,x}(t) &:= y(t; \tau, x, u(\cdot)) \stackrel{(2.3)}{=} x + \int_{\tau}^t f(s, y_{\tau,x}(s), u(s)) ds \\
 &= x + \underbrace{\int_{\tau}^{t_0} f(s, y_{\tau,x}(s), u(s)) ds}_{=y_{\tau,x}(t_0)} + \int_{t_0}^t f(s, y_{\tau,x}(s), u(s)) ds \\
 &= y_{\tau,x}(t_0) + \int_{t_0}^t f(s, y_{\tau,x}(s), u(s)) ds.
 \end{aligned}$$

Betrachten wir nun die Differenz dieser Trajektorien:

$$\begin{aligned}
 |y_{t_0,x}(t) - y_{\tau,x}(t)| &= \left| x + \int_{t_0}^t f(s, y_{t_0,x}(s), u(s)) ds - y_{\tau,x}(t_0) - \int_{t_0}^t f(s, y_{\tau,x}(s), u(s)) ds \right| \\
 &= \left| x - y_{\tau,x}(t_0) + \int_{t_0}^t f(s, y_{t_0,x}(s), u(s)) ds - \int_{t_0}^t f(s, y_{\tau,x}(s), u(s)) ds \right| \\
 &\leq |x - y_{\tau,x}(t_0)| + \left| \int_{t_0}^t f(s, y_{t_0,x}(s), u(s)) ds - \int_{t_0}^t f(s, y_{\tau,x}(s), u(s)) ds \right| \\
 &\leq \underbrace{|x - y_{\tau,x}(t_0)|}_{\stackrel{(2.7)}{\leq} \tilde{R}(t_0; \tau, x)} + \int_{t_0}^t |f(s, y_{t_0,x}(s), u(s)) - f(s, y_{\tau,x}(s), u(s))| ds.
 \end{aligned}$$

Aufgrund der Beschränkung der Trajektorien (siehe (2.8)) existiert ein $R = R(T, x)$ mit $\max\{|y_{t_0,x}(s)|, |y_{\tau,x}(s)|\} \leq R \forall s \in [t_0, t]$. Mit (A_3) folgt:

$$|y_{t_0,x}(t) - y_{\tau,x}(t)| \leq \tilde{R}(t_0; \tau, x) + \int_{t_0}^t L_f(R) |y_{t_0,x}(s) - y_{\tau,x}(s)| ds.$$

Da $h(t) := \tilde{R}(t_0; \tau, x)$ unabhängig von t und somit konstant und nicht fallend ist, folgt mit dem Gronwall-Lemma (Satz 2.5):

$$|y_{t_0,x}(t) - y_{\tau,x}(t)| \leq \tilde{R}(t_0; \tau, x) e^{L_f(R)(t-t_0)}.$$

Somit kann man die Lipschitzstetigkeit der Trajektorien bezüglich des Startzeitpunkts t_0 in Abhängigkeit des Startwertes x mit Hilfe der Definition von $\tilde{R}(t_0; \tau, x)$ folgern (siehe (2.7)):

$$\begin{aligned} |y_{t_0,x}(t) - y_{\tau,x}(t)| &\leq \overbrace{C(t_0 - \tau) e^{C(t_0 - \tau)} (1 + |x|)}^{\tilde{R}(t_0; \tau, x)} e^{L_f(R)(t-t_0)} \\ &\leq C(t_0 - \tau) e^{CT} (1 + |x|) e^{L_f(R)T} \\ &\leq L_y |t_0 - \tau| (1 + |x|), \quad \text{mit } L_y := C e^{CT} e^{L_f(R)T}. \end{aligned}$$

(ii) Betrachte die Funktion

$$\phi(t) := \underbrace{|y(t; t_0, x, u(\cdot))|}_{=: y_{t_0,x}(t)} - \underbrace{|y(t; t_0, z, u(\cdot))|}_{=: y_{t_0,z}(t)} \quad (2.11)$$

Da $y_{t_0,x}(t)$ als Lösung der Differentialgleichung stetig differenzierbar ist, lässt sich die Ableitung $\frac{d}{dt}\phi(t)$ folgendermaßen bilden:

$$\begin{aligned} \frac{d}{dt}\phi(t) &= \frac{d}{dt} |y_{t_0,x}(t) - y_{t_0,z}(t)|^2 \\ &= 2 |y_{t_0,x}(t) - y_{t_0,z}(t)| |\dot{y}_{t_0,x}(t) - \dot{y}_{t_0,z}(t)| \\ &= 2 |y_{t_0,x}(t) - y_{t_0,z}(t)| \underbrace{|f(t, y_{t_0,x}(t), u(t)) - f(t, y_{t_0,z}(t), u(t))|}_{\stackrel{(A_3)}{\leq} L_f(R) |y_{t_0,x}(t) - y_{t_0,z}(t)|}. \end{aligned}$$

Ein solches R existiert wegen der Beschränktheit der Lösungen aufgrund des linearen Wachstums (siehe (2.8)) analog zu Fall (i). Somit lässt sich zusammenfassen:

$$\frac{d}{dt}\phi(t) \leq 2 L_f(R) |y_{t_0,x}(t) - y_{t_0,z}(t)|^2$$

Also gilt durch Verwendung von (2.11):

$$\frac{d}{dt}\phi(t) \leq 2 L_f(R) \phi(t) \quad \text{fast überall auf } t \in [t_0, T]. \quad (2.12)$$

Da $\phi(t) = |y_{t_0,x}(t) - y_{t_0,z}(t)|^2$ absolut stetig ist, erhält man durch Integration der Ungleichung (2.12):

$$\int_{t_0}^t \dot{\phi}(s) ds \leq \int_{t_0}^t 2 L_f(R) \phi(s) ds$$

$$\begin{aligned} \Leftrightarrow \quad \phi(t) - \phi(t_0) &\leq 2 L_f(R) \int_{t_0}^t \phi(s) ds \\ \Leftrightarrow \quad \phi(t) &\leq \phi(t_0) + 2 L_f(R) \int_{t_0}^t \phi(s) ds. \end{aligned}$$

Die Aussage (2.6) des Gronwall-Lemma (Satz 2.5) lässt sich auf $\phi(t)$ anwenden, da $\phi(t_0)$ konstant und somit nicht fallend ist. Damit folgt die Abschätzung:

$$\begin{aligned} \phi(t) &\leq \phi(t_0) e^{2L_f(R)(t-t_0)} \\ \Leftrightarrow \quad |y_{t_0,x}(t) - y_{t_0,z}(t)|^2 &\leq |y_{t_0,x}(t_0) - y_{t_0,z}(t_0)|^2 e^{2L_f(R)(t-t_0)} \\ \Rightarrow \quad |y_{t_0,x}(t) - y_{t_0,z}(t)| &\leq |x - z| e^{L_f(R)(t-t_0)}. \end{aligned}$$

□

Zusätzlich zu diesen Aussagen benötigen wir im weiteren Verlauf dieser Arbeit noch den den im Zusammenhang mit Kontrollsystemen geläufigen Begriff der Konkatenation, welcher die Verkettung zweier Kontrollen beschreibt:

Definition 2.9 Konkatenation

Für zwei Funktionen $u_1(\cdot), u_2(\cdot) \in \mathcal{U}$ und $s \in [t_0, T]$ definiert man die *Konkatenation zur Zeit s* als

$$u_1 \&_s u_2(t) := \begin{cases} u_1(t), & t < s \\ u_2(t), & t \geq s. \end{cases} \quad (2.13)$$

Betrachtet man als Kontrollfunktionsmenge $\mathcal{U} := \{u: [t_0, T] \rightarrow U, u \text{ messbar}\}$, $U \subset \mathbb{R}^m$, befindet sich die Konkatenation wieder in \mathcal{U} :

$$u_1 \&_s u_2(\cdot) \in \mathcal{U}.$$

2.2 Subdifferential und Subgradient

In diesem Kapitel wird nun der Grundstein zur Definition der Viskositätslösungen gelegt, indem der Begriff des Subdifferentials eingeführt wird. Hier sei an die klassische Definition der mehrdimensionalen Ableitung erinnert. Das Subdifferential lässt sich damit intuitiv herleiten. Man beachte, dass an dieser Stelle auf die explizite Zeitabhängigkeit in der Formulierung der Aussagen und in den Beweisen verzichtet wird, allerdings lassen sich diese Aussagen vollständig analog auch auf eine zeitvariante Betrachtung anwenden. Dieser Abschnitt folgt dabei [Grü04].

Definition 2.10

Sei $\Omega \subset \mathbb{R}^n$ offen. Eine Funktion $w: \Omega \rightarrow \mathbb{R}$ heißt *differenzierbar* in $x_0 \in \Omega$, wenn ein $p_0 \in \mathbb{R}^n$ existiert, so dass gilt:

$$\lim_{\substack{x \rightarrow x_0 \\ x \neq x_0}} \frac{w(x) - w(x_0) - p_0 \cdot (x - x_0)}{|x - x_0|} = 0. \quad (2.14)$$

Hierbei bezeichnet „ \cdot “ das euklidische Skalarprodukt im \mathbb{R}^n .

Wenn ein solches $p_0 \in \mathbb{R}^n$ existiert, ist dies der eindeutig bestimmte Gradient $Dw(x_0)$ von w an der Stelle x_0 .

Die obige Schreibweise der Differenzierbarkeitsbedingung als einzelne Gleichung (2.14) ist für uns hier jedoch unpraktisch und wir verwenden daher die Zerlegung der Gleichung in zwei Ungleichungen:

$$\limsup_{\substack{x \rightarrow x_0 \\ x \neq x_0}} \frac{w(x) - w(x_0) - p_0 \cdot (x - x_0)}{|x - x_0|} \leq 0, \quad (2.15)$$

$$\liminf_{\substack{x \rightarrow x_0 \\ x \neq x_0}} \frac{w(x) - w(x_0) - p_0 \cdot (x - x_0)}{|x - x_0|} \geq 0. \quad (2.16)$$

Bemerkung 2.11

Diese Darstellung ermöglicht eine Verallgemeinerung des klassischen Ableitungsbegriffs. So gibt es für eine reellwertige Funktion w , welche stetig, aber nicht notwendigerweise differenzierbar ist, Punkte, in denen (2.14) für kein $p_0 \in \mathbb{R}^n$ erfüllt ist. Dennoch kann (2.15) oder (2.16) für bestimmte Vektoren $p_0 \in \mathbb{R}^n$ erfüllt sein. Falls aber ein Punkt $p_0 \in \mathbb{R}^n$ existiert, welcher sowohl (2.16) und gleichzeitig (2.15) erfüllt, so gilt für diesen auch (2.14).

Beispiel 2.12

Betrachte die Funktion $w: \mathbb{R} \rightarrow \mathbb{R}$, $w(x) = |x|$, die im Punkt $x_0 = 0$ stetig, aber nicht differenzierbar ist.

Für diese Funktion $w(x) = |x|$ gilt nach Definition 2.10:

$$\lim_{\substack{x \rightarrow 0 \\ x \neq 0}} \frac{|x| - |0| - p_0 \cdot (x - 0)}{|x - 0|} = \lim_{\substack{x \rightarrow 0 \\ x \neq 0}} \frac{|x| - p_0 \cdot x}{|x|} = \lim_{\substack{x \rightarrow 0 \\ x \neq 0}} \left(1 - p_0 \cdot \frac{x}{|x|} \right) \stackrel{?}{=} 0,$$

$\stackrel{?}{=}$: es existiert kein p_0 , so dass dieser Grenzwert 0 erreicht.

Dieser Grenzwert kann nun für positive bzw. negative Werte x separat betrachtet werden und mit geeigneter Wahl von p_0 gegen 0 konvergieren. Allerdings existiert kein p_0 , welches diese Aussage für beliebige x erfüllt. Dennoch ist Ungleichung (2.16) für alle $p_0 \in [-1, 1]$ erfüllt:

$$\liminf_{\substack{x \rightarrow 0 \\ x \neq 0}} \frac{|x| - |0| - p_0 \cdot (x - 0)}{|x - 0|} = \liminf_{\substack{x \rightarrow 0 \\ x \neq 0}} \frac{|x| - p_0 \cdot x}{|x|} = \liminf_{\substack{x \rightarrow 0 \\ x \neq 0}} \left(1 - p_0 \cdot \frac{x}{|x|} \right) \geq 0.$$

Diese informelle Aufteilung des klassischen Ableitungsbegriffs führt zur Definition des Fréchet-Differentials.

Definition 2.13

Seien $w(\cdot) \in C(\Omega; \mathbb{R})$ und $x_0 \in \Omega$.

Dann ist das *Fréchet-Superdifferential* von w im Punkt x_0 definiert durch

$$\begin{aligned} D^+w(x_0) &:= \{p_0 \in \mathbb{R}^n \mid p_0 \text{ erfüllt (2.15)}\} \\ &= \{p_0 \in \mathbb{R}^n \mid \limsup_{\substack{x \rightarrow x_0, x \in \Omega, \\ x \neq x_0}} \frac{w(x) - w(x_0) - p_0 \cdot (x - x_0)}{|x - x_0|} \leq 0\}. \end{aligned} \quad (2.17)$$

Analog ist das *Fréchet-Subdifferential* von w im Punkt x_0 definiert durch

$$\begin{aligned} D^-w(x_0) &:= \{p_0 \in \mathbb{R}^n \mid p_0 \text{ erfüllt (2.16)}\} \\ &= \{p_0 \in \mathbb{R}^n \mid \liminf_{\substack{x \rightarrow x_0, x \in \Omega, \\ x \neq x_0}} \frac{w(x) - w(x_0) - p_0 \cdot (x - x_0)}{|x - x_0|} \geq 0\}. \end{aligned} \quad (2.18)$$

Die Elemente dieser Mengen werden manchmal als *Fréchet-Supergradienten*, respektive *Fréchet-Subgradienten*, bezeichnet.

Falls $D^+w(x_0) \neq \emptyset$ ist, nennt man w *Fréchet-superdifferenzierbar*.

Analog spricht man von der *Fréchet-Subdifferenzierbarkeit* von w , falls $D^-w(x_0) \neq \emptyset$ gilt.

Lemma 2.14

$D^+w(x_0)$ und $D^-w(x_0)$ sind konvex und abgeschlossen.

Beweis:

a) zz.: $D^+w(x_0)$ und $D^-w(x_0)$ sind konvex.

Die Konvexität von $D^+w(x_0)$ und $D^-w(x_0)$ folgt direkt aus der Definition des \limsup und \liminf .

b) zz.: $D^+w(x_0)$ und $D^-w(x_0)$ sind abgeschlossen.

Die Abgeschlossenheit von $D^+w(x_0)$ wird durch einen Widerspruchsbeweis gezeigt, die Abgeschlossenheit von $D^-w(x_0)$ kann man analog folgern.

Betrachte eine Folge $p_n \rightarrow p_0$, so dass $p_n \in D^+w(x_0)$ für alle $n \in \mathbb{N}$ und

$$\lim_{n \rightarrow \infty} \frac{w(x_n) - w(x_0) - p_0 \cdot (x_n - x_0)}{|x_n - x_0|} = \alpha > 0 \quad (2.19)$$

für eine Folge $x_n \rightarrow x$. Für k groß genug ist $|p_k - p_0| \leq \alpha/2$. Durch Addition und Subtraktion von $p_k \cdot \frac{(y_n - x)}{|y_n - x|}$ zu (2.19) erhält man

$$\limsup_{n \rightarrow \infty} \frac{w(x_n) - w(x_0) - p_k \cdot (x_n - x_0)}{|x_n - x_0|} \geq \frac{\alpha}{2},$$

was einen Widerspruch zu $p_k \in D^+w(x_0)$ darstellt.

□

Bemerkung 2.15

Im Allgemeinen gilt der Zusammenhang

$$D^-(-w)(x_0) = -D^+w(x_0).$$

zwischen dem Fréchet-Subdifferential und dem Fréchet-Superdifferential.

Dies folgt direkt aus Definition 2.13.

Lemma 2.16

Seien $w \in C(\Omega; \mathbb{R})$, $x_0 \in \Omega$ und $p_0 \in \mathbb{R}^n$, dann sind folgende Aussagen äquivalent:

1. $p_0 \in D^+w(x_0)$ (bzw. $p_0 \in D^-w(x_0)$).
2. Es existiert ein $\phi \in C^1(\Omega; \mathbb{R})$ mit $D\phi(x_0) = p_0$, so dass $w - \phi$ ein lokales Maximum (bzw. Minimum) bei x_0 besitzt.

Beweis:

Wir zeigen die Äquivalenz der beiden Aussagen, indem wir zuerst die notwendige und anschließend die hinreichende Bedingung beweisen.

- 1. \Rightarrow 2.:

Sei $p_0 \in D^+w(x_0)$. Damit gilt nach Definition 2.13:

$$\limsup_{\substack{x \rightarrow x_0 \\ x \neq x_0}} \frac{w(x) - w(x_0) - p_0 \cdot (x - x_0)}{|x - x_0|} \leq 0.$$

Dann gibt es - wie nachfolgend noch bewiesen wird - eine auf einer Umgebung $N(x_0)$ von x_0 definierte stetige Funktion $\varepsilon(\cdot)$ mit $\varepsilon(x_0) = 0$, welche auf $N(x_0) \setminus \{x_0\}$ sogar stetig differenzierbar ist. Diese Funktion besitzt die Eigenschaft, dass sie die obere Schranke für das Argument des lim sup darstellt:

$$\frac{w(x) - w(x_0) - p_0 \cdot (x - x_0)}{|x - x_0|} \leq \varepsilon(x). \quad (2.20)$$

Um die Existenz dieser Funktion $\varepsilon(\cdot)$ zu beweisen, betrachten wir Ringe $R_n := B(x_0, \frac{1}{2^n}) \setminus B(x_0, \frac{1}{2^{n+1}})$ um x_0 (siehe Abbildung 2.1).

Für alle derartigen Ringe R_n gilt:

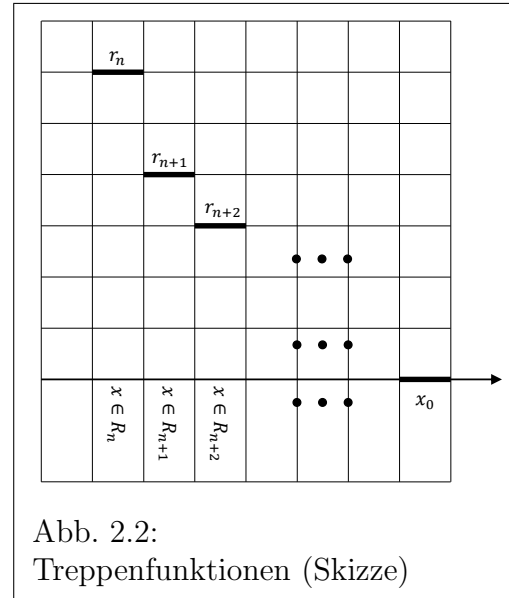
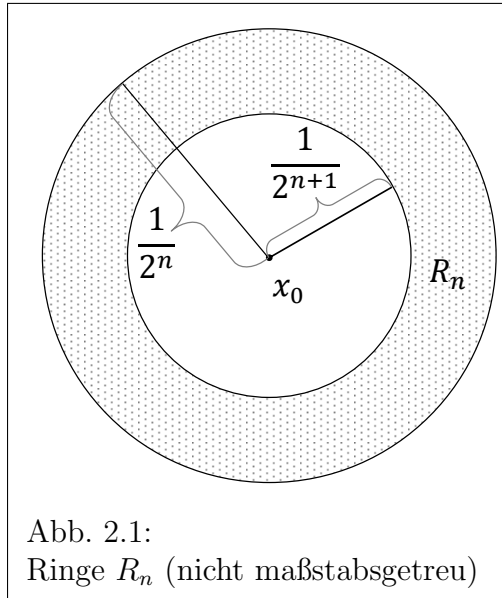
$\exists r_n$, so dass $\frac{w(x) - w(x_0) - p_0 \cdot (x - x_0)}{|x - x_0|} \leq r_n \quad \forall x \in R_n$.

Weil $\limsup_{\substack{x \rightarrow x_0 \\ x \neq x_0}} \frac{w(x) - w(x_0) - p_0 \cdot (x - x_0)}{|x - x_0|} \leq 0$, können die Schranken r_n so gewählt werden,

dass gilt: $\lim_{n \rightarrow \infty} r_n = 0$ (siehe Abbildung 2.2). Wir können also für $x \in R_n$ eine Treppenfunktion $x \mapsto r_n$ definieren, die in x_0 durch 0 stetig fortsetzbar ist. Mit dieser Treppenfunktion ist es nun möglich, durch differenzierbares Abschneiden zwischen den einzelnen Treppenstufen die gesuchte Funktion $\varepsilon(\cdot)$ zu konstruieren.

Durch Multiplikation von (2.20) mit $|x - x_0| \geq 0$, erhält man

$$w(x) - w(x_0) - p_0 \cdot (x - x_0) \leq \varepsilon(x) |x - x_0|$$



$$\implies w(x) - w(x_0) - p_0 \cdot x - \varepsilon(x) |x - x_0| \leq -p_0 \cdot x_0. \quad (2.21)$$

Sei $\phi(x) := w(x_0) + p_0 \cdot x + \varepsilon(x) |x - x_0|$ eine Testfunktion. Dann ist $\phi(x)$ differenzierbar mit $D\phi(x) = p_0$ und es gilt:

$$\begin{aligned} w(x) - \phi(x) &= w(x) - \overbrace{(w(x_0) + p_0 \cdot x + \varepsilon(x) |x - x_0|)}^{\phi(x)} \\ &= w(x) - w(x_0) - p_0 \cdot x - \varepsilon(x) |x - x_0| \\ &\stackrel{(2.21)}{\leq} -p_0 \cdot x_0 \\ &= w(x_0) - \underbrace{w(x_0) - p_0 \cdot x_0 - \varepsilon(x) |x_0 - x_0|}_{=\phi(x_0)} \\ &= w(x_0) - \phi(x_0). \end{aligned}$$

Somit gilt also

$$w(x) - \phi(x) \leq w(x_0) - \phi(x_0) \quad \forall x \in N(x_0)$$

und $w - \phi$ besitzt ein lokales Maximum in x_0 . Da ϕ differenzierbar auf \mathbb{R}^n fortgesetzt werden kann, folgt:

Es existiert ein $\phi \in C^1(\mathbb{R}^n; \mathbb{R})$ mit $D\phi(x_0) = p_0$, so dass $w - \phi$ ein lokales Maximum in x_0 annimmt.

- 2. \implies 1.:

Sei $\phi(x) \in C^1(\Omega; \mathbb{R})$ mit $D\phi(x_0) = p_0$ und $w - \phi$ besitze ein lokales Maximum bei x_0 . Dann gilt auf einer Umgebung $N = N(x_0)$ von x_0 :

$$w(x) - \phi(x) \leq w(x_0) - \phi(x_0)$$

$$\Leftrightarrow w(x) - \omega(x_0) \leq \phi(x) - \phi(x_0).$$

Somit gilt:

$$\limsup_{\substack{x \rightarrow x_0 \\ x \neq x_0}} \frac{w(x) - w(x_0) - p_0 \cdot (x - x_0)}{|x - x_0|} \leq \limsup_{\substack{x \rightarrow x_0 \\ x \neq x_0}} \frac{\phi(x) - \phi(x_0) - p_0 \cdot (x - x_0)}{|x - x_0|} \stackrel{(2.14)}{=} 0.$$

Also enthält das Fréchet-Superdifferential $D^+w(x_0)$ den Punkt p_0 .

Diese Argumentation lässt sich auf $p_0 \in D^-w(x_0)$ analog übertragen. □

Bemerkung 2.17

Durch Ersetzen der Funktion $\phi(x)$ durch $\hat{\phi}(x) = \phi(x) \pm |x - x_0|^2$ kann man in Lemma 2.16 die Bedingung „lokales Maximum (Minimum)“ durch „streng lokales Maximum (Minimum)“ ersetzen.

Das folgende Lemma zeigt eine wichtige Eigenschaft des Subdifferentials, welche im Verlauf dieser Arbeit eine Identifizierung der klassischen Lösungen einer partiellen Differentialgleichung mit den sogenannten Viskositätslösungen und eine damit verbundene Eindeutigkeitsaussage ermöglicht.

Lemma 2.18

Sei $w(\cdot) \in C(\Omega; R)$.

Falls w in x_0 differenzierbar ist, gilt

$$D^+w(x_0) = D^-w(x_0) = \{Dw(x_0)\}. \quad (2.22)$$

Beweis:

Zum Beweis dieser Aussage zeigen wir beide Inklusionen:

Sei w in x_0 differenzierbar, so folgt aus Definition 2.13 sofort $Dw(x_0) \in D^\pm(x_0)$.

Sei umgekehrt $p_0 \in D^+w(x_0)$ und w in x_0 differenzierbar. Dann folgt mit Lemma 2.16 die Existenz einer Funktion $\phi \in C^1(\Omega; R)$ mit $D\phi(x_0) = p_0$, so dass $w - \phi$ bei x_0 ein lokales Maximum annimmt. Da sowohl w als auch ϕ differenzierbar in x_0 sind, ist auch $w - \phi$ an dieser Stelle differenzierbar und es gilt $D(w - \phi)(x_0) = 0$, also $Dw(x_0) = D\phi(x_0)$. Da $D\phi(x_0) = p_0$ ist, gilt $p_0 = Dw(x_0)$. Aus der Eindeutigkeit des Gradienten folgt nun die Behauptung.

Die Aussage für $D^-w(x_0)$ lässt sich analog beweisen. □

Bemerkung 2.19

Beide Mengen $D^+w(x_0)$ und $D^-w(x_0)$ können gleichzeitig leer sein.

Beispiel 2.20

Sei $w: \mathbb{R} \rightarrow \mathbb{R}$ mit

$$w(x) := \begin{cases} x \sin\left(\frac{1}{x}\right), & x \neq 0 \\ 0, & x = 0. \end{cases}$$

Für diese Funktion w lässt sich das Fréchet-Subdifferential folgendermaßen angeben

$$\begin{aligned} D^-w(0) &= \{p_0 \in \mathbb{R}^n \mid \liminf_{\substack{x \rightarrow 0, x \in \mathbb{R}, \\ x \neq 0}} \frac{w(x) - w(0) - p_0 \cdot (x - 0)}{|x - 0|} \geq 0\} \\ &= \{p_0 \in \mathbb{R}^n \mid \liminf_{\substack{x \rightarrow 0, x \in \mathbb{R}, \\ x \neq 0}} \frac{w(x) - p_0 \cdot x}{|x|} \geq 0\} \\ &= \{p_0 \in \mathbb{R}^n \mid \liminf_{\substack{x \rightarrow 0, x \in \mathbb{R}, \\ x \neq 0}} \frac{x \sin\left(\frac{1}{x}\right) - p_0 \cdot x}{|x|} \geq 0\}. \end{aligned}$$

Da $\sin\left(\frac{1}{x}\right)$ für x gegen 0 nicht konvergiert, sondern im Bereich der Funktionswerte zwischen -1 und $+1$ alterniert, lässt sich für das Produkt $(\sin\left(\frac{1}{x}\right) - p_0) \cdot x$ im Grenzwert $x \rightarrow 0$ kein p_0 finden, für welches das Produkt für alle x und somit der \liminf positiv ist. Aus diesem Grund besteht das Subdifferential lediglich aus der leeren Menge.

Diese Argumentation lässt sich analog auf das Superdifferential übertragen.

Damit gilt die Gleichheit $D^+w(0) = D^-w(0) = \emptyset$.

Bemerkung 2.21

Die Tatsache, dass $D^-w(x_0)$ bzw. $D^+w(x_0)$ nur einelementig ist, impliziert nicht die Differenzierbarkeit von w .

Beispiel 2.22

Betrachte $w: \mathbb{R} \rightarrow \mathbb{R}$ mit

$$w(x) := \begin{cases} \max\left(x \sin\left(\frac{1}{x}\right), 0\right), & x \neq 0 \\ 0, & x = 0. \end{cases}$$

Dann ist w nicht differenzierbar bei 0, obwohl

$$\begin{aligned} D^-w(0) &= \{p_0 \in \mathbb{R}^n \mid \liminf_{\substack{x \rightarrow 0, x \in \mathbb{R}, \\ x \neq 0}} \frac{w(x) - w(0) - p_0 \cdot (x - 0)}{|x - 0|} \geq 0\} \\ &= \{p_0 \in \mathbb{R}^n \mid \liminf_{\substack{x \rightarrow 0, x \in \mathbb{R}, \\ x \neq 0}} \frac{w(x) - p_0 \cdot x}{|x|} \geq 0\} = \emptyset. \end{aligned}$$

3 Wertefunktion und dynamische Programmierung

In diesem neuen Abschnitt verlassen wir die Grundlagen und führen die Wertefunktion mit ihren wichtigsten Eigenschaften ein. Ebenso zeigen eine sehr wichtige Aussage für die Verwendung der Wertefunktion: Das „Prinzip der dynamischen Programmierung“. Zunächst beginnen wir mit der Definition des optimalen Steuerungsproblems.

3.1 Einführung der Wertefunktion

Definition 3.1 (Optimales Steuerungsproblem)

Ein optimales Steuerungsproblem zu einem gegebenen zeitvarianten Kontrollsystem (2.1) lässt sich auf einem endlichen Zeitintervall $[t_0, T]$ durch folgendes Optimierungsproblem beschreiben:

$$\mathcal{I} \left\{ \begin{array}{l} \text{Minimiere } \int_{t_0}^T L(t, x(t; t_0, x_0, u(\cdot)), u(t)) dt + g(x(T; t_0, x_0, u(\cdot))) \\ \text{über den messbaren Funktionen } u \in \mathcal{U} \\ \text{und } x \in \mathcal{W}^{1,1}([t_0, T]; \mathbb{R}^n) \\ \text{s.d. } \dot{x}(t) = f(t, x(t), u(t)) \quad \text{f.ü. auf } [t_0, T] \\ u(t) \in U \quad \text{f.ü. auf } [t_0, T] \\ x(t_0) = x_0 \end{array} \right.$$

mit

- betrachtetem Zeitintervall $[t_0, T] \subset \mathbb{R}$,
- Kontrollfunktionsmenge $\mathcal{U} := \{u: [t_0, T] \rightarrow U, \text{ messbar}\}$,
- $f: [t_0, T] \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ wie in Annahme 2.3, zusammen mit (A'_1) statt (A_1) ,
- Momentankostenfunktional $L: [t_0, T] \times \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$,
- Endkostenfunktional $g: \mathbb{R}^n \rightarrow \mathbb{R}$.

Durch Variation der Anfangsbedingung $x(t_0) = x_0$ zu $y(t) = x$ lässt sich ein Steuerungsproblem $\mathcal{I}_{t,x}$ abhängig von variabler Anfangszeit $t \in [t_0, T]$ und variablem

Anfangswert x einführen:

$$\mathcal{I}_{t,x} \left\{ \begin{array}{l} \text{Minimiere } J(u(\cdot); t, x) := \int_t^T L(s, y(s; t, x, u(\cdot)), u(s)) ds + g(y(T; t, x, u(\cdot))) \\ \text{über den messbaren Funktionen } u : [t, T] \rightarrow \mathbb{R}^m \\ \text{und } y \in \mathcal{W}^{1,1}([t, T]; \mathbb{R}^n) \\ \text{s.d. } \dot{y}(s) = f(s, y(s), u(s)) \quad \text{f.ü. auf } [t, T] \\ u(s) \in U \quad \text{f.ü. auf } [t, T] \\ y(t) = x. \end{array} \right.$$

Die Funktion

$$J(u(\cdot); t, x) := \int_t^T L(s, y(s; t, x, u(\cdot)), u(s)) ds + g(y(T; t, x, u(\cdot))), \quad (3.1)$$

welche in $\mathcal{I}_{t,x}$ minimiert wird, nennt man *Zielfunktion*. Mit Hilfe dieses Steuerungsproblems $\mathcal{I}_{t,x}$ und der *Zielfunktion* lässt sich die Wertefunktion definieren:

Definition 3.2 (Wertefunktion)

Die Funktion $V(t, x) := \inf \mathcal{I}_{t,x}$ ist die optimale Wertefunktion (*Optimalwertfunktion*) dieses optimalen Steuerungsproblems $\mathcal{I}_{t,x}$.

Als explizite Formulierung erhält man

$$V(t, x) := \inf_{u(\cdot) \in \mathcal{U}} J(u(\cdot); t, x). \quad (3.2)$$

Die Wertefunktion stellt somit das Infimum der Gesamtkosten entlang der Trajektorien über alle Kontrollen bzw. Steuerungen dar. Das Ziel eines optimalen Steuerungsproblems ist es also eine Kontrolle $u(\cdot)$ zu finden, die die Summe aus Trajektorienkosten (Kosten, welche entlang der Trajektorie entstehen)

$$\int_t^T L(s, y(s; t, x, u(\cdot)), u(s)) ds$$

und Endkosten

$$g(y(T; t, x, u(\cdot)))$$

minimiert und dabei alle Nebenbedingungen erfüllt.

Abhängig davon, welche Terme im optimalen Steuerungsproblem tatsächlich auftreten, unterscheidet man folgende Problembeschreibungen, welche ineinander umwandelbar sind:

- $g = 0$ & $L \neq 0$: Lagrange-Problem,

- $g \neq 0$ & $L = 0$: Mayer-Problem,
- $g \neq 0$ & $L \neq 0$: Bolza-Problem.

Die Formulierung als Bolza-Problem stellt die allgemeine Formulierung des optimalen Steuerungsproblems dar und soll daher im Folgenden verwendet werden.

Annahme 3.3

Zusätzlich zu den bereits bekannten Annahmen werden nun auch Bedingungen an die einzelnen Terme der Zielfunktion, also Momentankostenfunktional und Endkostenfunktional, gestellt. Zur besseren Übersichtlichkeit werden die neuen Annahmen (A_4) und (A_5) und die bereits in Kap. 2.1 auf den Seiten 4 und 5 eingeführten Annahmen hier noch einmal kompakt angegeben:

$$\begin{aligned}
 (A_0) \quad & \left\{ \begin{array}{l} U \subset \mathbb{R}^m, U \text{ kompakt} \\ f : [0, T] \times \mathbb{R}^n \times U \rightarrow \mathbb{R}^n \text{ ist stetig} \end{array} \right. \\
 (A_1) \quad & \left\{ \begin{array}{l} f \text{ besitzt lineares Wachstum auf } [0, T] \times \mathbb{R}^n \times U, \\ \text{d.h. } \exists C > 0, \text{ sd. } |f(t, x, u)| \leq C(1 + |x|) \\ \forall t \in [0, T], \forall x \in \mathbb{R}^n, \forall u \in U \end{array} \right. \\
 (A_2) \quad & \left\{ \begin{array}{l} |f(t, y, u) - f(s, x, u)| \leq \omega_f(|t - s| + |x - y|, R) \\ \forall t, s \in [0, T], x, y \in B(0, R), \forall R > 0 \end{array} \right. \\
 (A_3) \quad & \left\{ \begin{array}{l} |f(t, x, u) - f(t, y, u)| \leq L_f(R) |x - y| \\ \forall t \in [0, T], x, y \in B(0, R), \forall R > 0, u \in U \end{array} \right. \\
 (A_4) \quad & \left\{ \begin{array}{l} L : [0, T] \times \mathbb{R}^n \times U \rightarrow \mathbb{R} \text{ ist stetig} \\ |L(t, x, u) - L(s, y, u)| \leq \omega_L(|t - s| + |x - y|) \\ |L(t, x, u)| \leq M_L \\ \forall t, s \in [0, T], x, y \in \mathbb{R}^n, \forall u \in U \end{array} \right. \\
 (A_5) \quad & \left\{ \begin{array}{l} |g(x) - g(y)| \leq \omega_g(|x - y|) \\ |g(x)| \leq M_g \\ \forall x, y \in \mathbb{R}^n \end{array} \right.
 \end{aligned}$$

Mit Hilfe dieser Annahmen lassen sich einige grundlegende Eigenschaften der Wertefunktion herleiten.

3.2 Grundlegende Eigenschaften der Wertefunktion

Satz 3.4 Stetigkeit und Beschränktheit der Wertefunktion

Seien die Annahmen aus Annahme 3.3 erfüllt. Dann ist die Wertefunktion $V(t, x)$ stetig und beschränkt auf $[0, T] \times \mathbb{R}^n$.

Beweis:

Seien (τ, z) und $\varepsilon > 0$ gegeben. Wähle eine messbare Kontrolle u_ε , welche für das Optimierungsproblem mit Anfangsbedingung $y(\tau) = z$ ε -optimal ist. Also gilt nach Definition 3.2 für die zugehörige ε -optimale Trajektorie $s \mapsto y(s; \tau, z, u_\varepsilon(\cdot)) =: y_{\tau, z}(s)$:

$$\begin{aligned} V(\tau, z) &\geq J(u_\varepsilon; \tau, z) - \varepsilon \\ &= \int_{\tau}^T L(s, y_{\tau, z}(s), u_\varepsilon(s)) ds + g(y_{\tau, z}(T)) - \varepsilon. \end{aligned} \quad (3.3)$$

Verwendet man diese Kontrolle u_ε in Verbindung mit einer anderen Anfangsbedingung $y(t) = x$, erhält man eine neue Trajektorie $s \mapsto y(s; t, x, u_\varepsilon(\cdot)) =: y_{t, x}(s)$. Mit dieser Kontrolle u_ε erfüllt die Wertefunktion, als Infimum über alle möglichen Kontrollen $u(\cdot) \in \mathcal{U}$ (Definition 3.2), folgende Ungleichung:

$$V(t, x) \leq \int_t^T L(s, y_{t, x}(s), u_\varepsilon(s)) ds + g(y_{t, x}(T)).$$

Somit gilt:

$$\begin{aligned} V(t, x) - V(\tau, z) &\leq \int_t^T L(s, y_{t, x}(s), u_\varepsilon(s)) ds + g(y_{t, x}(T)) \\ &\quad - \left[\int_{\tau}^T L(s, y_{\tau, z}(s), u_\varepsilon(s)) ds + g(y_{\tau, z}(T)) - \varepsilon \right] \\ &= \underbrace{\int_t^T L(s, y_{t, x}(s), u_\varepsilon(s)) ds - \int_{\tau}^T L(s, y_{\tau, z}(s), u_\varepsilon(s)) ds}_{\textcircled{1}} \\ &\quad + \underbrace{g(y_{t, x}(T)) - g(y_{\tau, z}(T))}_{\textcircled{2}} + \varepsilon. \end{aligned}$$

Eine separate Betrachtung der einzelnen Summanden liefert:

- Summand $\textcircled{1}$:

$$\int_t^T L(s, y_{t, x}(s), u_\varepsilon(s)) ds - \int_{\tau}^T L(s, y_{\tau, z}(s), u_\varepsilon(s)) ds$$

$$(3.4) \leq \left| \int_t^T L(s, y_{t,x}(s), u_\varepsilon(s)) ds - \int_\tau^T L(s, y_{\tau,z}(s), u_\varepsilon(s)) ds \right| \quad (3.4)$$

Eine einfach durchzuführende, aber sehr technische Fallunterscheidung mit $t > \tau$ und $\tau > t$ und die Additivität der Integrale liefert¹:

$$\begin{aligned} (3.4) &\leq \left| \int_{t \vee \tau}^T L(s, y_{t,x}(s), u_\varepsilon(s)) ds - \int_{t \vee \tau}^T L(s, y_{\tau,z}(s), u_\varepsilon(s)) ds \right| \\ &\quad + \left| \int_{t \wedge \tau}^{t \vee \tau} L(s, y_{t \wedge \tau, (x|z)}(s), u_\varepsilon(s)) ds \right| \\ &\leq \int_{t \vee \tau}^T |L(s, y_{t,x}(s), u_\varepsilon(s)) - L(s, y_{\tau,z}(s), u_\varepsilon(s))| ds \\ &\quad + \int_{t \wedge \tau}^{t \vee \tau} |L(s, y_{t \wedge \tau, (x|z)}(s), u_\varepsilon(s))| ds. \end{aligned}$$

Unter Verwendung der Eigenschaften von L in (A_4) folgt:

$$\begin{aligned} &\leq \int_{t \vee \tau}^T \omega_L (|s - \tau| + |y_{t,x}(s) - y_{\tau,z}(s)|) ds + \int_{t \wedge \tau}^{t \vee \tau} M_L ds \\ &= \int_{t \vee \tau}^T \omega_L (|y_{t,x}(s) - y_{\tau,z}(s)|) ds + \underbrace{((t \vee \tau) - (t \wedge \tau))}_{=|t-\tau|} M_L \\ &\leq \int_{t \vee \tau}^T \omega_L \left(\underbrace{|y_{t,x}(s) - y_{t,z}(s)|}_{\stackrel{(2.10)}{\leq} e^{L_f(R)T} |x-z|} + \underbrace{|y_{t,z}(s) - y_{\tau,z}(s)|}_{\stackrel{(2.9)}{\leq} L_y |t-\tau| (1+|z|)} \right) ds + |t - \tau| M_L \\ &\leq \int_{t \vee \tau}^T \omega_L (e^{L_f(R)T} |x - z| + L_y |t - \tau| (1 + |z|)) ds + |t - \tau| M_L \\ &\leq (T - (t \vee \tau)) \omega_L (e^{L_f(R)T} |x - z| + L_y |t - \tau| (1 + |z|)) + |t - \tau| M_L \\ &\leq T \omega_L (e^{L_f(R)T} |x - z| + L_y |t - \tau| (1 + |z|)) + |t - \tau| M_L. \end{aligned}$$

- Summand ②:

$$g(y_{t,x}(T)) - g(y_{\tau,z}(T)) \leq |g(y_{t,x}(T)) - g(y_{\tau,z}(T))|$$

¹Für die verwendeten Notationen siehe „Notations- und Abkürzungsverzeichnis“ auf Seite IV

$$\begin{aligned}
& \stackrel{(A_5)}{\leq} \omega_g (|y_{t,x}(T) - y_{t,z}(T)|) \\
& \leq \omega_g \left(\underbrace{|y_{t,x}(T) - y_{t,z}(T)|}_{\stackrel{(2.10)}{\leq} e^{L_f(R)(T-t)}|x-z|} + \underbrace{|y_{t,z}(T) - y_{\tau,z}(T)|}_{\stackrel{(2.9)}{\leq} L_y|t-\tau|(1+|z|)} \right) \\
& \leq \omega_g \left(e^{L_f(R)(T-t)} |x - z| + L_y |t - \tau| (1 + |z|) \right) \\
& \leq \omega_g \left(e^{L_f(R)T} |x - z| + L_y |t - \tau| (1 + |z|) \right).
\end{aligned}$$

Somit ergibt sich insgesamt:

$$\begin{aligned}
V(t, x) - V(\tau, z) & \leq T\omega_L \left(e^{L_f(R)T} |x - z| + L_y |t - \tau| (1 + |z|) \right) + |t - \tau| M_L \\
& \quad + \omega_g \left(e^{L_f(R)T} |x - z| + L_y |t - \tau| (1 + |z|) \right) + \varepsilon.
\end{aligned}$$

Da $\varepsilon > 0$ beliebig war und auf der rechten Seite nur noch explizit und nicht mehr implizit auftritt (die direkte Abhängigkeit von u_ε ist weggefallen), folgt aus dem Grenzübergang für $\varepsilon \rightarrow 0$ und dem Vertauschen der Rollen von (t, x) und (τ, z) :

$$\begin{aligned}
|V(t, x) - V(\tau, z)| & \leq T\omega_L \left(e^{L_f(R)T} |x - z| + L_y |t - \tau| (1 + |z|) \right) + |t - \tau| M_L \\
& \quad + \omega_g \left(e^{L_f(R)T} |x - z| + L_y |t - \tau| (1 + |z|) \right)
\end{aligned}$$

und somit die Stetigkeit der Wertefunktion. ◁

Die Beschränktheit von $V(t, x)$ folgt aus der Beschränktheit von $J(u(\cdot); t, x)$:

$$\begin{aligned}
|J(u(\cdot); t, x)| & = \left| \int_t^T L(s, y_{t,x}(s), u(s)) ds + g(y_{t,x}(T)) \right| \\
& \leq \int_t^T \underbrace{|L(s, y_{t,x}(s), u(s))|}_{\stackrel{(A_4)}{\leq} M_L} ds + \underbrace{|g(y_{t,x}(T))|}_{\stackrel{(A_5)}{\leq} M_g} \\
& \leq (T - t)M_L + M_g \\
& \leq TM_L + M_g
\end{aligned}$$

für alle $u \in \mathcal{U}$, $t \in [0, T]$ und $x \in \mathbb{R}^n$. ◻

Folgerung 3.5

Unter den verschärften Annahmen, dass g Lipschitz-stetig und $|L(t, x, u) - L(s, y, u)| \leq L_L(|t - s| + |x - y|)$, für alle $t, s \in [0, T]$, $x, y \in \mathbb{R}^n$ und $u \in U$ ist (Verschärfung von (A_3)), lässt sich die Lipschitzstetigkeit von V folgern.

3.3 Dynamische Programmierung

Im Folgenden werden wir nun eine zentrale Eigenschaft der Wertefunktion nennen und beweisen: Das „Bellman’sche Optimalitätsprinzip“. Dieses sagt vereinfacht gesprochen aus, dass Optimalitätsprobleme auf beliebige Intervalle zerlegt und einzeln betrachtet werden kann.²

Satz 3.6 „Bellman’sches Optimalitätsprinzip“

(i) Für die optimale Wertefunktion gilt für alle $(t, x) \in [t_0, T) \times \mathbb{R}^n$:

$$V(t, x) = \inf_{u(\cdot) \in \mathcal{U}} \left\{ \int_t^\tau L(s, y[s], u(s)) ds + V(\tau, y[\tau]) \right\} \quad \forall \tau \in [t, T), \quad (3.5)$$

wobei $y[\cdot] := y(\cdot; t, x, u(\cdot))$ durch $\dot{y}(s) = f(s, y(s), u(s))$, $y(t) = x$ festgelegt ist und $V(T, x) = g(x)$ gilt.

(ii) Für ein optimales Funktionenpaar $(\bar{y}(\cdot), \bar{u}(\cdot))$ ($\bar{u}(\cdot)$ stetige Kontrollfunktion) gilt für alle $(t, x) \in [t_0, T) \times \mathbb{R}$:

$$V(t, x) = \int_t^\tau L(s, \bar{y}[s], \bar{u}(s)) ds + V(\tau, \bar{y}[\tau]) \quad \forall \tau \in [t, T). \quad (3.6)$$

Beweis:

(i) Wir zeigen die Gleichheit durch Beweis der entsprechenden Ungleichungen in beiden Richtungen.

„ \leq “: Sei $\tau \in [t, T)$, $u: [t, T] \rightarrow U$ und $\tilde{u}: [\tau, T] \rightarrow U$ beliebig. Definiere

$$\hat{u}(s) = u \&_\tau \tilde{u}(s) \text{ für } s \in [t, T).$$

Somit gilt $\hat{u}(\cdot) \in \mathcal{U}$. Die zugehörige Lösungstrajektorie $\hat{y}(\cdot)$ von $\dot{\hat{y}}(s) = f(s, \hat{y}(s), \hat{u}(s))$ mit $\hat{y}(t) = x$ erfüllt mit Anfangswert $\tilde{y}(\cdot)$, Lösungstrajektorie von $\dot{\tilde{y}}(s) = f(s, \tilde{y}(s), \tilde{u}(s))$ mit Anfangswert $\tilde{y}(\tau) = y[\tau]$, folgende Eigenschaft:

$$\hat{y}[s] = y \&_\tau \tilde{y}[s] \text{ für } s \in [t, T).$$

Damit gilt explizit ausformuliert:

$$\hat{y}[s] := \begin{cases} y[s] = y(s; t, x, u(\cdot)), & \text{falls } s \in [t, \tau) \\ \tilde{y}[s] = \tilde{y}(s; \tau, y[\tau], \tilde{u}(\cdot)), & \text{falls } s \in [\tau, T). \end{cases} \quad (3.7)$$

²siehe [Vin00, S. 30ff.], [Bre01, S. 42ff.]

Also ist $\hat{y}[\cdot]$ stetig, aber nicht notwendigerweise differenzierbar.
Nach Definition 3.2 gilt für $V(t, x)$ die folgende Ungleichung:

$$\begin{aligned}
V(t, x) &\leq \int_t^T L(s, \hat{y}[s], \hat{u}(s)) ds + g(\hat{y}[T]) \\
&= \overbrace{\int_t^T L(s, \hat{y}[s], \hat{u}(s)) ds}^{=J(\hat{u}(\cdot); t, x)} + g(\hat{y}[T]) \\
&\stackrel{(3.7)}{=} \int_t^\tau L(s, y[s], u(s)) ds + \underbrace{\int_\tau^T L(s, \tilde{y}[s], \tilde{u}(s)) ds + g(\tilde{y}[T])}_{=J(\tilde{u}(\cdot); \tau, y[\tau])} \\
&= \int_t^\tau L(s, y[s], u(s)) ds + J(\tilde{u}(\cdot); \tau, y[\tau]).
\end{aligned}$$

Der Übergang zum Infimum über alle $\tilde{u}: [\tau, T] \rightarrow U$ und alle $u: [t, T] \rightarrow U$ liefert:

$$V(t, x) \leq \inf_{u(\cdot) \in \mathcal{U}} \left\{ \int_t^\tau L(s, y[s], u(s)) ds + V(\tau, y[\tau]) \right\} \quad \forall \tau \in [t, T]. \quad (3.8)$$

„ \geq “: Sei $\varepsilon > 0$ und seien $\hat{y}(\cdot)$ und $\hat{u}(\cdot)$ Konkatenationen (wie oben), so dass $J(\hat{u}(\cdot); t, x) \leq V(t, x) + \varepsilon$. Dann gilt:

$$\begin{aligned}
V(t, x) + \varepsilon &\geq J(\hat{u}(\cdot); t, x) \stackrel{(3.1)}{=} \int_t^T L(s, \hat{y}[s], \hat{u}(s)) ds + g(\hat{y}[T]) \\
&= \int_t^\tau L(s, \underbrace{\hat{y}[s]}_{=y[s]}, \underbrace{\hat{u}(s)}_{=u(s)}) ds + \int_\tau^T L(s, \underbrace{\hat{y}[s]}_{=\tilde{y}[s]}, \underbrace{\hat{u}(s)}_{=\tilde{u}(s)}) ds + g(\underbrace{\hat{y}[T]}_{=\tilde{y}[T]}) \\
&\stackrel{(3.7)}{=} \int_t^\tau L(s, y[s], u(s)) ds + \underbrace{\int_\tau^T L(s, \tilde{y}[s], \tilde{u}(s)) ds + g(\tilde{y}[T])}_{\stackrel{(3.1)}{=} J(\tilde{u}(\cdot); \tau, y[\tau])} \\
&= \int_t^\tau L(s, y[s], u(s)) ds + J(\tilde{u}(\cdot); \tau, y[\tau])
\end{aligned}$$

$$\stackrel{(3.2)}{\geq} \int_t^\tau L(s, y[s], u(s)) ds + V(\tau, y[\tau]).$$

Der Übergang zum Infimum liefert dann

$$V(t, x) \geq \inf_{u(\cdot) \in \mathcal{U}} \left\{ \int_t^\tau L(s, y[s], u(s)) ds + V(\tau, y[\tau]) \right\}.$$

(ii) Nach Teil „ \geq “ aus (i) gilt für $\varepsilon > 0$ und die Konkatenationen $\hat{y}(\cdot)$ und $\hat{u}(\cdot)$ (wie oben), dass $J(\hat{u}(\cdot); t, x) \leq V(t, x) + \varepsilon$:

$$\begin{aligned} V(t, x) + \varepsilon \geq J(\hat{u}(\cdot); t, x) &\geq \int_t^\tau L(s, y[s], \hat{u}(s)) ds + V(\tau, y[\tau]) \\ &\stackrel{(3.8)}{\geq} V(t, x). \end{aligned}$$

Für optimales $(\bar{y}(\cdot), \bar{u}(\cdot))$ gilt diese Ungleichung mit $\varepsilon = 0$, so dass

$$V(t, x) \geq J(\bar{u}(\cdot); t, x) \geq \int_t^\tau L(s, \bar{y}[s], \bar{u}(s)) ds + V(\tau, \bar{y}[\tau]) \stackrel{(3.8)}{\geq} V(t, x).$$

Also gilt

$$V(t, x) = \int_t^\tau L(s, \bar{y}[s], \bar{u}(s)) ds + V(\tau, \bar{y}[\tau])$$

für alle $\tau \in [t, T)$.

□

Mit der Wertefunktion und dem dynamischen Prinzip können wir nun den Hamilton-Jacobii-Ansatz einführen. Hierbei gehen wir zweistufig vor. Wir differenzieren zwischen der Betrachtung des sogenannten glatten und unglatten Falls.

4 Hamilton-Jacobi-Ansatz

4.1 Hamilton-Jacobi-Ansatz mit glatter Wertefunktion

In diesem Kapitel beschäftigen wir uns mit dem Hamilton-Jacobi-Ansatz im sogenannten glatten Fall. Dieser lässt sich mit Hilfe der Definitionen des optimalen Steuerungsproblems und der Wertefunktion definieren. Hierbei spricht man unter bestimmten Voraussetzungen von einer glatten Lösung eines optimalen Steuerungsproblems:

Definition 4.1

Sei $\mathcal{I}_{t,x}$ ein optimales Steuerungsproblem für ein kontinuierliches System. Es existiert eine glatte Lösung dieses optimalen Steuerungsproblems, falls folgende Eigenschaften erfüllt sind:

- Die Wertefunktion $V(t, x)$ des optimalen Steuerungsproblems $\mathcal{I}_{t,x}$ ist eine C^1 -Funktion für $(t, x) \in [t_0, T] \times \mathbb{R}^n$.
- Für jedes Ereignis $(t, x) \in [t_0, T] \times \mathbb{R}^n$ existiert ein optimales Funktionenpaar $(\bar{y}(\cdot), \bar{u}(\cdot))$ ($\bar{u}(\cdot)$ stetige Kontrollfunktion), so dass

$$V(t, x) = \int_t^T L(s, \bar{y}(s; t, x, \bar{u}(\cdot)), \bar{u}(s)) ds + g(\bar{y}(T; t, x, \bar{u}(\cdot))). \quad (4.1)$$

Innerhalb dieses Kapitels gehen wir im Folgenden davon aus, dass das optimale Steuerungsproblem diese Eigenschaften besitzt.

Auch wird hier zur besseren Übersicht die Notation

$$y[s] := y(s; t, x, u(\cdot))$$

vermehrt verwendet, wenn die zusätzlichen Informationen im Kontext eindeutig sind und somit redundant wären.

4.1.1 Existenz der Lösung der Hamilton-Jacobi-Gleichung

Ausgehend vom Satz „Bellman’sches Optimalitätsprinzip“ (Satz 3.6 auf Seite 23) lassen sich Schlüsse über die Variation der optimalen Wertefunktion im infinitesimalen Sinne ziehen. Gleichung (3.6) liefert folgende Aussage:

Für jedes optimale Funktionenpaar $(\bar{y}(\cdot), \bar{u}(\cdot))$ mit dem Anfangswertepaar (t_0, x_0) gilt

$$V(t_0, x_0) = \int_{t_0}^t L(s, \bar{y}(s), \bar{u}(s)) ds + V(t, \bar{y}(t))$$

für alle $t \in [t_0, T]$. Somit erfüllt die optimale Wertefunktion ausgewertet entlang einer optimalen Trajektorie

$$V(t, \bar{y}(t)) - V(t_0, x_0) = \int_{t_0}^t L(s, \bar{y}(s), \bar{u}(s)) ds$$

die Bedingungen für absolute Stetigkeit und besitzt daher fast überall auf $[t_0, T]$ die Ableitung

$$\frac{dV(t, \bar{y}(t))}{dt} = -L(t, \bar{y}(t), \bar{u}(t)). \quad (4.2)$$

Entlang einer optimalen Trajektorie stimmen also die Momentankosten und die Steigung der Wertefunktion betragsmäßig überein.

Alternativ lässt sich, da $V(\cdot, \cdot)$ glatt ist, die Ableitung der Wertefunktion auch mittels partieller Ableitungen formulieren:

$$\frac{dV(t, \bar{y}(t))}{dt} = V_t + V_x \dot{\bar{y}}.$$

Somit gilt für die partielle Ableitung V_t

$$V_t = \frac{dV(t, \bar{y}(t))}{dt} - V_x \dot{\bar{y}}$$

und nach Einsetzen der Ableitung $dV(t, \bar{y}(t))/dt$ aus (4.2) die partielle Differentialgleichung

$$V_t(t_0, x_0) = -L(t, \bar{y}(t), \bar{u}(t)) - V_x(t_0, x_0) f(t, x, \bar{u}(t)), \quad (4.3)$$

vorausgesetzt, dass die partiellen Ableitungen von $V(\tau, \bar{y}(t))$ existieren.

Mit (3.5) erhält man nun für beliebige Trajektorien eine Abnahmerate der Wertefunktion, welche geringer ist als die Momentankosten. Es gilt also

$$V_t(t_0, x_0) \geq -L(t, y(t), u(t)) - V_x(t_0, x_0) f(t, x, u(t)) \quad (4.4)$$

für beliebige nicht notwendigerweise optimale Funktionspaare $(y(\cdot), u(\cdot))$.

Für $\tau = t_0$ ist die Abschätzung (3.5) mit Gleichheit erfüllt, da

$$\begin{aligned} V(t_0, x_0) &\leq \underbrace{\int_{t_0}^{t_0} L(s, y(s), u(s)) ds}_{=0} + \underbrace{V(t_0, y(t_0))}_{x_0} \\ &= V(t_0, x_0), \end{aligned}$$

was die Vermutung nahelegt, dass die Ableitungen bzgl. τ dieselbe Abschätzung erfüllen. Dies impliziert (4.4). Man beachte, dass in (4.3) und (4.4) nur die Kontrolle zum Zeitpunkt t eine Rolle spielt und nicht mehr die komplette Kontrollfunktion $u(\cdot)$. Falls also das Infimum, über welches die optimale Wertefunktion definiert ist, durch eine optimale Kontrolle tatsächlich als Minimum angenommen wird

Satz 4.3

Sei $V(t, x)$ die Wertefunktion des optimalen Steuerungsproblems $\mathcal{I}_{t,x}$ aus Definition 3.2. Es gelte

- (i) V ist stetig differenzierbar
- (ii) und für alle $(t, x) \in [t_0, T] \times \mathbb{R}^n$ besitzt das Optimierungsproblem $(\mathcal{I}_{t,x})$ einen Minimierer $\bar{y}(\cdot)$ mit stetiger Kontrollfunktion $\bar{u}(\cdot)$.

Dann ist $V(t, x)$ Lösung der HJ-PDE (4.5).

Beweis:

Wähle $(t, x) \in [t_0, T] \times \mathbb{R}^n$, $\tau \in [t, T]$, y stetig differenzierbar mit $y: [t, T] \rightarrow \mathbb{R}^n$ und ein Paar (\bar{y}, \bar{u}) als Minimierer für $\mathcal{I}_{t,x}$, wobei \bar{u} eine stetige Kontrollfunktion ist. Aufgrund von Satz 3.6 gilt

$$V(t, x) \leq \int_t^\tau L(s, y(s), u(s)) ds + V(\tau, y(\tau))$$

und

$$V(t, x) = \int_t^\tau L(s, \bar{y}(s), \bar{u}(s)) ds + V(\tau, \bar{y}(\tau)).$$

Umstellen der beiden Ungleichungen ergibt

$$V(\tau, y(\tau)) - V(t, x) + \int_t^\tau L(s, y(s), u(s)) ds \geq 0$$

und

$$V(\tau, \bar{y}(\tau)) - V(t, x) + \int_t^\tau L(s, \bar{y}(s), \bar{u}(s)) ds = 0.$$

Sei $y(s) = x + (s - t) \cdot f(t, x, u)$ und $\tau = t + \varepsilon$ mit $\varepsilon \in (0, T - t)$, dann gilt

$$V(t + \varepsilon, x + \varepsilon \cdot f(t, x, u)) - V(t, x) + \int_t^{t+\varepsilon} L(s, x + (s - t) \cdot f(t, x, u), u(s)) ds \geq 0$$

und

$$V(t + \varepsilon, x + \int_t^{t+\varepsilon} \dot{\bar{y}}(s) ds) - V(t, x) + \int_t^{t+\varepsilon} L(s, \bar{y}(s), \bar{u}(s)) ds = 0.$$

Mit $\varepsilon > 0$ erhält man

$$\begin{aligned} & \frac{1}{\varepsilon} [V(t + \varepsilon, x + \varepsilon \cdot f(t, x, u)) - V(t, x)] \\ & + \frac{1}{\varepsilon} \int_t^{t+\varepsilon} L(s, x + (s-t) \cdot f(t, x, u), u(s)) ds \geq 0 \end{aligned} \quad (4.6)$$

und

$$\frac{1}{\varepsilon} \left[V(t + \varepsilon, x + \int_t^{t+\varepsilon} \dot{y}(s) ds) - V(t, x) \right] + \frac{\int_t^{t+\varepsilon} L(s, \bar{y}(s), \bar{u}(s)) ds}{\varepsilon} = 0. \quad (4.7)$$

Für $\varepsilon \searrow 0$ ergibt sich damit aus (4.6), da $V(t, x)$ als stetig differenzierbar vorausgesetzt ist,

$$V_t(t, x) + V_x(t, x) \cdot f(t, x, u) + L(t, x, u) \geq 0$$

und aus (4.7) ebenfalls durch die stetige Differenzierbarkeit von $V(t, x)$

$$V_t(t, x) + V_x(t, x) \cdot \dot{y}(t) + L(t, x, u) = 0.$$

Da u beliebig war, folgt

$$V_t(t, x) + \min_{u \in \mathcal{U}} \{V_x(t, x) \cdot f(t, x, u) + L(t, x, u)\} = 0.$$

Also löst $V(t, x)$ die HJ-PDE (4.5). Die Randbedingung $V(T, x) = g(x)$ folgt direkt aus der Definition der Wertefunktion 3.2. □

4.1.2 Eindeutigkeit der Wertefunktion als klassische Lösung der Hamilton-Jacobi-Gleichung

Die Eindeutigkeit der Wertefunktion als klassische Lösung der Hamilton-Jacobi-Gleichung lässt sich als Spezialfall des Eindeutigkeitsbeweises der Viskositätslösungen (siehe Abschnitt 4.2.2 auf Seite 36) behandeln. Dazu muss nur die zusätzliche Annahme der stetigen Differenzierbarkeit der Wertefunktion V (diese wird im glatten Fall vorausgesetzt) getroffen werden. Die Aussage folgt dann analog.

4.2 Hamilton-Jacobi-Ansatz mit nichtglatter Wertefunktion

Im Laufe der Entwicklung des Hamilton-Jacobi-Ansatzes und der in diesem Zusammenhang wichtigen dynamischen Programmierung hat sich ein schwerwiegendes Problem offenbart, als man versuchte, die Anwendbarkeit der erschlossenen Bedingungen zu erweitern: Das Fehlen geeigneter Verfahren zur Untersuchung lokaler Eigenschaften von nicht differenzierbaren Funktionen und von Mengen mit nicht differenzierbaren Beschränkungen.

Diese Problematik lässt sich bei Betrachtung der dynamischen Programmierung leicht verifizieren. Die dynamische Programmierung legt das Hauptaugenmerk auf die Verbindung von optimalen Steuerungsproblemen wie

$$\mathcal{I} \left\{ \begin{array}{l} \text{Minimiere } \int_{t_0}^T L(t, x(t; t_0, x_0, u), u(t)) dt + g(x(T)) \\ \text{über den messbaren Funktionen } u : [t_0, T] \rightarrow \mathbb{R}^m \\ \text{und } x \in \mathcal{W}^{1,1}([t_0, T]; \mathbb{R}^n) \\ \text{s.d. } \dot{x}(t) = f(t, x(t), u(t)) \quad \text{f.ü.} \\ u(t) \in U \quad \text{f.ü.} \\ x(t_0) = x_0 \end{array} \right.$$

und der Hamilton-Jacobi-Gleichung

$$\text{(HJ-PDE)} \left\{ \begin{array}{l} \Phi_t(t, x) + \min_{u \in U} \{ \Phi_x(t, x) f(t, x, u) + L(t, x, u) \} = 0 \\ \forall (t, x) \in (t_0, T) \times \mathbb{R}^n \\ \Phi(T, x) = g(x) \quad \forall x \in \mathbb{R}^n. \end{array} \right.$$

Die klassische Theorie unter Voraussetzung einer stetig differenzierbaren (glatten) Wertefunktion $V(t, x)$ liefert unter Berücksichtigung einiger anderer Annahmen, dass $V(t, x)$ eine Lösung der (HJ-PDE) darstellt.

Allerdings schließt die klassische Theorie *nicht* die Existenz weiterer Lösungen für die (HJ-PDE) aus und stellt somit keine vollständige Charakterisierung der Wertefunktion als Lösung der (HJ-PDE) dar. Zudem schränkt die Differenzierbarkeit von $V(t, x)$ als notwendige Voraussetzung die Anwendbarkeit der klassischen Theorie sehr stark ein. Dies kann mit Hilfe eines einfachen Beispiels³ eines Mayer-Problems, stellvertretend für einen Problemtypus, welcher häufig in der Konstruktionsplanung auftritt, belegt werden.

³[Vin00, S. 36]

Beispiel 4.4

$$\left\{ \begin{array}{l} \text{Minimiere } x(1) \\ \text{über den messbaren Funktionen } u : [0, 1] \rightarrow \mathbb{R} \\ \text{und } x \in \mathcal{W}^{1,1}([0, 1]; \mathbb{R}) \\ \text{s.d. } \dot{x}(t) = xu \quad \text{f.ü.} \\ u(t) \in [-1, +1] \quad \text{f.ü.} \\ x(0) = 0 \end{array} \right.$$

Aus der allgemeinen Hamilton-Jacobi-Gleichung (4.5)

$$\text{(HJ-PDE)} \quad \left\{ \begin{array}{l} \Phi_t(t, x) + \min_{u \in U} \{ \Phi_x(t, x) f(t, x, u) + L(t, x, u) \} = 0 \\ \forall (t, x) \in (t_0, T) \times \mathbb{R} \\ \Phi(T, x) = g(x) \quad \forall x \in \mathbb{R} \end{array} \right.$$

ergibt sich mit $t_0 = 0$, $T = 1$, $L(t, x, u) \equiv 0$ und $g(x) = x$ zunächst

$$\left\{ \begin{array}{l} \Phi_t(t, x) + \min_{u \in U} \{ \Phi_x(t, x) f(t, x, u) \} = 0 \\ \forall (t, x) \in (0, 1) \times \mathbb{R} \\ \Phi(1, x) = x \quad \forall x \in \mathbb{R}. \end{array} \right.$$

Durch Verwendung von $U = [-1, +1]$ und $f(t, x, u) = xu$ lässt sich nun das Minimum bilden

$$\min_{u \in [-1, +1]} \{ \Phi_x(t, x) xu \} = - |\Phi_x(t, x)x|.$$

Somit erhält man für die (HJ-PDE)

$$\left\{ \begin{array}{l} \Phi_t(t, x) - |\Phi_x(t, x)x| = 0 \quad \forall (t, x) \in (0, 1) \times \mathbb{R} \\ \Phi(1, x) = x \quad \forall x \in \mathbb{R}. \end{array} \right.$$

Die Wertefunktion ergibt sich zu

$$V(t, x) = \begin{cases} x \exp^{+(1-t)} & \text{falls } x \geq 0 \\ x \exp^{-(1-t)} & \text{falls } x < 0. \end{cases}$$

$V(t, x)$ erfüllt die Hamilton-Jacobi-Gleichung auf dem Gebiet $\{(t, x) \in (0, 1) \times \mathbb{R} : x \neq 0\}$ mit

$$V_t(t, x) = \begin{cases} -x \exp^{+(1-t)} & \text{falls } x \geq 0 \\ x \exp^{-(1-t)} & \text{falls } x < 0 \end{cases}$$

und

$$V_x(t, x) = \begin{cases} \exp^{+(1-t)} & \text{falls } x \geq 0 \\ \exp^{-(1-t)} & \text{falls } x < 0. \end{cases}$$

Da $V(t, x)$ auf der Teilmenge $\{(t, x) \in (0, 1) \times \mathbb{R} : x = 0\}$ nicht differenzierbar ist, stellt $V(t, x)$ jedoch keine klassische Lösung dar.

4.2.1 Viskositätslösungen

Da die Bedingung der Existenz einer stetig differenzierbaren Wertefunktion $V(t, x)$ die Anwendung der im bisherigen Verlauf der Arbeit entwickelten klassischen Theorie auf viele praktische Probleme unmöglich macht, hat man nach einer Möglichkeit gesucht, den Lösungsbegriff von partiellen nichtlinearen Differentialgleichungen zu erweitern, um auch für nicht glatte Wertefunktionen eine Existenz- und Eindeutigkeitstheorie entwickeln zu können.

Dies führt zum Begriff der Viskositätslösungen für nichtlineare partielle Differentialgleichungen der Form

$$F(x, w(x), Dw(x)) = 0 \text{ für } x \in \Omega \stackrel{\text{offen}}{\subset} \mathbb{R}^n \quad (4.8)$$

mit $F: \Omega \times \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}$ stetig und Dw dem Gradienten von w . Viskositätslösungen sind verallgemeinerte Lösungen, welche nur die Stetigkeit als Voraussetzung haben und nicht unbedingt differenzierbar sein müssen. Für die Definition der Viskositätslösungen gibt es in der Literatur zwei mögliche Ansätze, welche jedoch nachweislich äquivalent sind. Diese Arbeit stützt sich hierbei auf [BCD08] und [Grü04].

Die erste Definition von Viskositätslösungen erfolgt nun mit Hilfe des *Fréchet-Superdifferentials* und des *Fréchet-Subdifferentials* (siehe Kapitel 2.2). Wie auch schon in diesem Kapitel wird auch hier anfänglich zur Formulierung der Aussagen auf eine explizite Zeitabhängigkeit verzichtet, die aber über eine einfache Koordinatentransformation wieder erhalten werden kann.

Definition 4.5

Eine stetige Funktion $w: \Omega \rightarrow \mathbb{R}$ heißt *Viskositätslösung* von (4.8), falls gilt:

$$F(x, w(x), p) \leq 0 \quad \forall x \in \Omega, \forall p \in D^+w(x) \quad (4.9)$$

$$\text{und } F(x, w(x), p) \geq 0 \quad \forall x \in \Omega, \forall p \in D^-w(x). \quad (4.10)$$

Bemerkung 4.6

Falls w differenzierbar ist, gilt mit (2.22) auf Seite 15

$$F(x, w(x), p) = 0 \quad \forall x \in \Omega, \forall p \in Dw(x). \quad (4.11)$$

Diese Definition der Viskositätslösung 4.5 orientiert sich stark an der Formulierung der klassischen Lösung. In Beweisen empfiehlt sich jedoch eine technischere Formulierung der Definition von Viskositätslösungen.

Definition 4.7

Eine stetige Funktion $w \in C(\Omega; \mathbb{R})$ heißt *Viskositätslösung* von (4.8), falls für alle Testfunktionen $\phi \in C^1(\Omega; \mathbb{R})$ gilt:

$$\begin{aligned} w - \phi \text{ besitzt in } x_0 \text{ ein lokales Maximum} &\Rightarrow F(x_0, w(x_0), D\phi(x_0)) \leq 0, \\ w - \phi \text{ besitzt in } x_0 \text{ ein lokales Minimum} &\Rightarrow F(x_0, w(x_0), D\phi(x_0)) \geq 0. \end{aligned}$$

Die Äquivalenz der beiden Definition 4.5 und Definition 4.7 ergibt sich mit Hilfe der Eigenschaften der Subdifferentialiale als direkte Folgerung von Lemma 2.16.

Überträgt man diese Formulierungen nun auf den evolutionären Fall

$$u_t + H(t, x, Du(x)) = 0 \text{ für } (t, x) \in]0, T[\times \Omega \quad (4.12)$$

ergibt sich folgende Definition.

Definition 4.8

Eine stetige Funktion $u \in C(]0, T[\times \Omega; \mathbb{R})$ heißt *Viskositätslösung* von (4.12), falls für alle Testfunktionen $\phi = \phi(t, x) \in C^1(]0, T[\times \Omega; \mathbb{R})$ gilt:

$$\begin{aligned} u - \phi \text{ besitzt in } (t_0, x_0) \text{ ein lokales Maximum} &\Rightarrow \phi_t(t_0, x_0) + H(t_0, x_0, D\phi) \leq 0, \\ u - \phi \text{ besitzt in } (t_0, x_0) \text{ ein lokales Minimum} &\Rightarrow \phi_t(t_0, x_0) + H(t_0, x_0, D\phi) \geq 0. \end{aligned}$$

Satz 4.9

Sei nun Annahme 3.3 von Seite 19 erfüllt. Dann ist die Wertefunktion $V(t, x)$ Viskositätslösung der Hamilton-Jacobi-Gleichung

$$- [V_t + H(t, x, DV)] = 0 \quad (t, x) \in]0, T[\times \mathbb{R}^n \quad (4.13)$$

mit der Endbedingung

$$V(T, x) = g(x) \quad x \in \mathbb{R}^n \quad (4.14)$$

und dem Hamiltonfunktional

$$H(t, x, p) := \inf_{u \in U} \{p \cdot f(t, x, u) + L(t, x, u)\}. \quad (4.15)$$

Beweis:

Durch Satz 3.4 ist die Wertefunktion $V(t, x)$ stetig auf $[0, T] \times \mathbb{R}^n$. Die Endbedingung (4.14) ist trivialerweise erfüllt. Um zu zeigen, dass $V(t, x)$ eine Viskositätslösung ist, sei $\phi \in C^1(]0, T[\times \mathbb{R}^n)$. Durch die Struktur des Problems ergeben sich zwei separate Eigenschaften, welche bewiesen werden müssen:

(P1) $V(t, x)$ ist Viskositätsunterlösung, d.h.

falls $V - \phi$ ein lokales Maximum bei $(\bar{t}, \bar{x}) \in]0, T[\times \mathbb{R}^n$ besitzt, dann gilt

$$\phi(\bar{t}, \bar{x}) + \inf_{u \in U} \{D\phi(\bar{t}, \bar{x})f(\bar{t}, \bar{x}, u) + L(\bar{t}, \bar{x}, u)\} \geq 0. \quad (4.16)$$

(P2) $V(t, x)$ ist Viskositätsüberlösung, d.h.
falls $V - \phi$ ein lokales Minimum bei $(\bar{t}, \bar{x}) \in]0, T[\times \mathbb{R}^n$ besitzt, dann gilt

$$\phi(\bar{t}, \bar{x}) + \inf_{u \in U} \{D\phi(\bar{t}, \bar{x})f(\bar{t}, \bar{x}, u) + L(\bar{t}, \bar{x}, u)\} \leq 0. \quad (4.17)$$

(P1) Sei $\phi \in C^1(]0, T[\times \mathbb{R}^n)$ und (\bar{t}, \bar{x}) ein lokales Maximum von $V - \phi$, d.h. es existiert eine lokale Umgebung $N = N(\bar{t}, \bar{x})$ von (\bar{t}, \bar{x}) , so dass

$$\begin{aligned} (V - \phi)(\bar{t}, \bar{x}) &\geq (V - \phi)(s, z), && \text{für alle } (s, z) \in N \\ \Leftrightarrow \exists r > 0 : V(\bar{t}, \bar{x}) - \phi(\bar{t}, \bar{x}) &\geq V(s, z) - \phi(s, z), && \text{falls } |\bar{x} - z| < r, |\bar{t} - s| < r \\ \Leftrightarrow \phi(\bar{t}, \bar{x}) - \phi(s, z) &\leq V(\bar{t}, \bar{x}) - V(s, z), && \text{falls } |\bar{x} - z| < r, |\bar{t} - s| < r. \end{aligned}$$

Sei $u \in U$ konstant, aber beliebig und sei $y_{\bar{t}, \bar{x}}(\cdot) := y(\cdot; \bar{t}, \bar{x}, u)$ die zugehörige Trajektorie zur konstanten Kontrolle $u(t) = u$ für alle t . Wenn τ klein genug ist, existiert nach (2.7) ein $r > 0$, so dass $|y_{\bar{t}, \bar{x}}(\bar{t} + \tau) - \bar{x}| < r$ und auch $|\bar{t} - (\bar{t} + \tau)| < r$. Somit kann man $s = \bar{t} + \tau$ und $z = y_{\bar{t}, \bar{x}}(\bar{t} + \tau)$ setzen und es folgt

$$\phi(\bar{t}, \bar{x}) - \phi(\bar{t} + \tau, y_{\bar{t}, \bar{x}}(\bar{t} + \tau)) \leq V(\bar{t}, \bar{x}) - V(\bar{t} + \tau, y_{\bar{t}, \bar{x}}(\bar{t} + \tau)). \quad (4.18)$$

Durch das Infimum aus dem Satz „Bellman’sches Optimalitätsprinzip“ (Satz 3.6, S. 23) erhält man die Ungleichung

$$V(\bar{t}, \bar{x}) \leq \int_{\bar{t}}^{\bar{t} + \tau} L(s, y_{\bar{t}, \bar{x}}(s), u) ds + V(\bar{t} + \tau, y_{\bar{t}, \bar{x}}(\bar{t} + \tau)). \quad (4.19)$$

Somit folgt aus (4.18) mit (4.19) :

$$\begin{aligned} \phi(\bar{t}, \bar{x}) - \phi(\bar{t} + \tau, y_{\bar{t}, \bar{x}}(\bar{t} + \tau)) &\leq \int_{\bar{t}}^{\bar{t} + \tau} L(s, y_{\bar{t}, \bar{x}}(s), u) ds \\ &\quad + V(\bar{t} + \tau, y_{\bar{t}, \bar{x}}(\bar{t} + \tau)) \\ &\quad - V(\bar{t} + \tau, y_{\bar{t}, \bar{x}}(\bar{t} + \tau)) \\ &= \int_{\bar{t}}^{\bar{t} + \tau} L(s, y_{\bar{t}, \bar{x}}(s), u) ds. \end{aligned}$$

Als nächstes dividiert man durch τ und betrachtet den Grenzwert für $\tau \rightarrow 0$. Damit erhält man

$$-\phi(\bar{t}, \bar{x}) - D\phi(\bar{t}, \bar{x}) \cdot \underbrace{\dot{y}(\bar{t}; \bar{t}, \bar{x}, u)}_{f(\bar{t}, \bar{x}, u)} \leq L(\bar{t}, \bar{x}, u)$$

$$\Leftrightarrow -\phi(\bar{t}, \bar{x}) - D\phi(\bar{t}, \bar{x}) \cdot f(\bar{t}, \bar{x}, u) - L(\bar{t}, \bar{x}, u) \leq 0.$$

Da $u \in U$ beliebig gewählt ist, gilt diese Aussage auch für das Supremum

$$\begin{aligned} & -\phi(\bar{t}, \bar{x}) + \sup_{u \in U} \{-D\phi(\bar{t}, \bar{x}) \cdot f(\bar{t}, \bar{x}, u) - L(\bar{t}, \bar{x}, u)\} \leq 0 \\ \Leftrightarrow & -\left[\phi(\bar{t}, \bar{x}) + \inf_{u \in U} \{D\phi(\bar{t}, \bar{x}) \cdot f(\bar{t}, \bar{x}, u) + L(\bar{t}, \bar{x}, u)\} \right] \leq 0 \\ \Leftrightarrow & \phi(\bar{t}, \bar{x}) + \inf_{u \in U} \{D\phi(\bar{t}, \bar{x}) \cdot f(\bar{t}, \bar{x}, u) + L(\bar{t}, \bar{x}, u)\} \geq 0. \end{aligned}$$

Somit ist (4.16) erfüllt und $V(t, x)$ ist Viskositätssublösung.

(P2) Sei $\phi \in C^1(]0, T[\times \mathbb{R}^n)$ und (\bar{t}, \bar{x}) ein lokales Minimum von $V - \phi$, d.h. es existiert analog zu (P1) eine lokale Umgebung $N = N(\bar{t}, \bar{x})$ von (\bar{t}, \bar{x}) , so dass

$$\begin{aligned} & (V - \phi)(\bar{t}, \bar{x}) \leq (V - \phi)(s, z), & \text{für alle } (s, z) \in N \\ \Leftrightarrow \exists r > 0 : & V(\bar{t}, \bar{x}) - \phi(\bar{t}, \bar{x}) \leq V(s, z) - \phi(s, z), & \text{falls } |\bar{x} - z| < r, |\bar{t} - s| < r \\ \Leftrightarrow \phi(\bar{t}, \bar{x}) - \phi(s, z) & \geq V(\bar{t}, \bar{x}) - V(s, z), & \text{falls } |\bar{x} - z| < r, |\bar{t} - s| < r. \end{aligned}$$

Aufgrund von Satz 3.6 existiert für alle $\varepsilon > 0$ und für alle $\tau > 0$ ein $\bar{u} \in \mathcal{U}$ (abhängig von ε und τ), so dass gilt

$$V(\bar{t}, \bar{x}) \geq \int_{\bar{t}}^{\bar{t}+\tau} L(s, y_{\bar{t}, \bar{x}}(s), \bar{u}(s)) ds + V(\bar{t} + \tau, y_{\bar{t}, \bar{x}}(\bar{t} + \tau)) - \tau\varepsilon, \quad (4.20)$$

wobei $y_{\bar{t}, \bar{x}}(\cdot) := y(\cdot; \bar{t}, \bar{x}, \bar{u}(\cdot))$ die Lösungstrajektorie zum Anfangswert (\bar{t}, \bar{x}) mit Kontrolle \bar{u} ist.

□

4.2.2 Eindeutigkeit der Wertefunktion als Viskositätslösung der Hamilton-Jacobi-Gleichung

Zusätzlich zur Existenz der Lösung ist natürlich die Eindeutigkeit sehr wichtig. Um diese zu erhalten, benötigen wir das sogenannte Vergleichsprinzip, welches eine Art Monotonie in der Betrachtung zweier Lösungen der Hamilton-Jacobi-Bellman-Gleichung bezeichnet.

Satz 4.10 Vergleichsprinzip

Sei $H: [0, T] \times \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ stetig und erfülle

$$|H(t, x, p) - H(t, x, q)| \leq C |p - q| \quad (4.21a)$$

$$|H(t, x, p) - H(s, y, p)| \leq \omega_H(|x - y| + |t - s|)(1 + |p|) \quad (4.21b)$$

mit ω_H Stetigkeitsmodul von H .

Seien u, v beschränkte, gleichmäßig stetige Viskositätsunter- und Viskositätsüberlösungen von

$$u_t + H(t, x, Du) = 0 \quad \text{auf } (0, T) \times \mathbb{R}^n \quad (4.22)$$

$$u(0, x) = g(x) \quad \text{auf } \mathbb{R}^n. \quad (4.23)$$

Falls

$$u(0, x) \leq v(0, x)$$

für alle $x \in \mathbb{R}^n$ gilt, dann folgt

$$u(t, x) \leq v(t, x) \quad \forall (t, x) \in [0, T] \times \mathbb{R}^n. \quad (4.24)$$

Zunächst benötigen wir als vorbereitende Maßnahme eine Aussage, welche die Definition der Viskositätsunterlösung auf die obere Grenze des Zeitintervalls erweitert.

Lemma 4.11

Sei u eine stetige Funktion auf $[0, T] \times \mathbb{R}^n$, welche eine Viskositätsunterlösung für (4.22) auf $]0, T[$ darstellt. Falls $\phi \in C^1$ existiert, so dass $u - \phi$ ein lokales Maximum bei (T, x_0) besitzt, dann gilt die Ungleichung

$$\phi_t(T, x_0) + H(T, x_0, D\phi(T, x_0)) \leq 0. \quad (4.25)$$

Beweis:

$u - \phi$ nehme bei (T, x_0) ein streng lokales Maximum an. Betrachte für alle $\varepsilon > 0$ die Funktion

$$\phi_\varepsilon(t, x) = \phi(t, x) + \frac{\varepsilon}{T - t}.$$

Jede Funktion $u - \phi_\varepsilon$ besitzt somit ein lokales Maximum bei $(t_\varepsilon, x_\varepsilon)$ mit

$$t_\varepsilon < T, \quad (t_\varepsilon, x_\varepsilon) \rightarrow (T, x_0) \text{ für } \varepsilon \rightarrow 0^+.$$

Da u eine Viskositätsunterlösung ist und $(t_\varepsilon, x_\varepsilon) \in]0, T[\times \mathbb{R}^n$ gilt

$$\begin{aligned} & \underbrace{\phi_{\varepsilon, t}(t_\varepsilon, x_\varepsilon)}_{\frac{\partial}{\partial t}(\phi(t, x) + \frac{\varepsilon}{T-t})} + H(t_\varepsilon, x_\varepsilon, D\phi_\varepsilon(t_\varepsilon, x_\varepsilon)) \leq 0 \\ \Leftrightarrow & \quad \phi_t(t_\varepsilon, x_\varepsilon) + H(t_\varepsilon, x_\varepsilon, D\phi(t_\varepsilon, x_\varepsilon)) \leq -\frac{\varepsilon}{(T - t_\varepsilon)^2}. \end{aligned}$$

Mit dem Grenzübergang $\varepsilon \rightarrow 0$ folgt somit (4.25). □

Mit diesem Lemma können wir nun das Vergleichsprinzip beweisen. Die Beweisidee folgt dabei den Ausführungen von [Bre01, S. 19ff.] und [CL13, S. 21].

Dieser Beweis wird aufgrund seines Umfangs in mehrere Schritte gegliedert:

Beweis (Satz 4.10, Vergleichsprinzip):

- (i) Betrachten wir zunächst den Fall, dass u, v klassische Lösungen sind.
Falls die Aussage des Satzes nicht erfüllt wäre, müsste

$$\sup_{t,x} \{u(t, x) - v(t, x)\} > 0$$

gelten.

Somit kann man $\sigma, \lambda > 0$ finden, so dass

$$\sup_{t,x} \{u(t, x) - v(t, x) - 2\lambda t\} = \sigma > 0 \quad (4.26)$$

ist. Angenommen das Supremum wird tatsächlich in einem Punkt⁴ $(t_\varepsilon, x_\varepsilon)$ angenommen, möglicherweise sogar mit $t_\varepsilon = T$. Da u, v klassische Lösungen sind, also im Besonderen differenzierbar sind, gilt nach der notwendigen Extremalbedingung $\nabla(u(t, x) - v(t, x) - 2\lambda t) = 0$. Somit folgt für $(t_\varepsilon, x_\varepsilon)$

$$\begin{aligned} u_t(t_\varepsilon, x_\varepsilon) - v_t(t_\varepsilon, x_\varepsilon) - 2\lambda &= 0 \\ \Leftrightarrow u_t(t_\varepsilon, x_\varepsilon) - v_t(t_\varepsilon, x_\varepsilon) &> 0 \\ \Leftrightarrow u_t(t_\varepsilon, x_\varepsilon) &> v_t(t_\varepsilon, x_\varepsilon) \end{aligned}$$

und

$$Du(t_\varepsilon, x_\varepsilon) - Dv(t_\varepsilon, x_\varepsilon) \Leftrightarrow Du(t_\varepsilon, x_\varepsilon) = Dv(t_\varepsilon, x_\varepsilon).$$

Allerdings gilt nach der Definition 4.8 der Viskositätslösung (siehe S. 34) auch für klassische Lösungen:

$$\begin{aligned} u_t(t_\varepsilon, x_\varepsilon) + H(t_\varepsilon, x_\varepsilon, Du(t_\varepsilon, x_\varepsilon)) &\leq 0 \\ v_t(t_\varepsilon, x_\varepsilon) + H(t_\varepsilon, x_\varepsilon, Dv(t_\varepsilon, x_\varepsilon)) &\geq 0. \end{aligned}$$

Durch Subtrahieren der beiden Ungleichungen erhält man

$$u_t(t_\varepsilon, x_\varepsilon) - v_t(t_\varepsilon, x_\varepsilon) + \underbrace{H(t_\varepsilon, x_\varepsilon, Du(t_\varepsilon, x_\varepsilon)) - H(t_\varepsilon, x_\varepsilon, Dv(t_\varepsilon, x_\varepsilon))}_{=0, \text{ da } Du(t_\varepsilon, x_\varepsilon) = Dv(t_\varepsilon, x_\varepsilon)} \leq 0$$

$$\Leftrightarrow u_t(t_\varepsilon, x_\varepsilon) - v_t(t_\varepsilon, x_\varepsilon) \leq 0$$

$$\Leftrightarrow u_t(t_\varepsilon, x_\varepsilon) \leq v_t(t_\varepsilon, x_\varepsilon).$$

Dies stellt einen Widerspruch zu $u_t(t_\varepsilon, x_\varepsilon) > v_t(t_\varepsilon, x_\varepsilon)$ dar, was die Aussage des Satzes für diesen Spezialfall beweist.

⁴ $(t_\varepsilon, x_\varepsilon)$ hängt in diesem Fall nicht explizit von einem $\varepsilon > 0$ ab. Um die Parallelen mit der Ausweitung des Beweises in (ii) aufzeigen zu können, wurden die Variablen hier genauso bezeichnet.

- (ii) Die Idee des Widerspruchsbeweises aus (i) muss nun auf den Fall der allgemeinen Viskositätslösungen ohne Differenzierbarkeitsannahme ausgeweitet werden. Statt an dieser Stelle die Ableitungen von u und v direkt zu betrachten, führt man die glatten Testfunktionen ϕ und ψ ein, um das Verhalten der Ursprungsfunktionen zu untersuchen.

Dazu verwendet man eine Kombination zweier Beweistechniken: Verdopplung der Variablen und Einführung von Straftermen. Dieses Verfahren orientiert sich an mehreren Beweisen in [BCD08, S. 151ff.] und [CL13, S. 21].

Man betrachtet daher die Funktion:

$$\Phi(t, x, s, y) := u(t, x) - v(s, y) - \frac{|t - s|^2 + |x - y|^2}{2\varepsilon} - \lambda(t + s) - \frac{\varepsilon}{2}(|x|^2 + |y|^2) \quad (4.27)$$

mit $\lambda, \varepsilon > 0$. Die einzelnen Strafterme kommen folgendermaßen zustande:

$\frac{ t-s ^2+ x-y ^2}{2\varepsilon}$	Bestraft die Entfernung der künstlichen Variablen zueinander und stellt sicher, dass diese eng beieinander liegen, so dass sie annähernd wieder als „eine“ Variable gesehen werden können. Wir werden dazu im Folgenden beweisen, dass dieser Quotient für $\varepsilon \rightarrow 0$ gegen 0 konvergiert.
$\lambda(t + s)$	Bewirkt, dass die Bedingung der Viskositätsunterlösung strikt erfüllt wird.
$\frac{\varepsilon}{2}(x ^2 + y ^2)$	Bestraft die „Unendlichkeit“. Dadurch wird sichergestellt, dass das Supremum auch wirklich angenommen wird, da Φ an sich auf einem unbeschränkten Gebiet definiert ist.

Wähle $\sigma, \lambda, \varepsilon > 0$ so, dass (4.26) aus (i) erfüllt ist.

Da Φ als Summe von stetigen Funktionen auf dem gesamten Gebiet stetig ist und $\lim_{|x|+|y|\rightarrow\infty} \Phi = -\infty$ gilt, existiert ein Quadrupel $(t_\varepsilon, x_\varepsilon, s_\varepsilon, y_\varepsilon) \in ([0, T] \times \mathbb{R}^n)^2$, so dass

$$\sup_{([0, T] \times \mathbb{R}^n)^2} \Phi = \Phi(t_\varepsilon, x_\varepsilon, s_\varepsilon, y_\varepsilon). \quad (4.28)$$

(Beweisskizze für die Existenz dieses Quadrupels:

Es existiert ein Kompaktum, außerhalb dessen die Funktion auf jeden Fall kleiner ist als eine vorgegebene Schranke. Aufgrund der Stetigkeit der Funktion wird das Supremum innerhalb dieses Kompaktums angenommen.)

Zudem gilt aus Kombination von (4.26) und (4.27) für ε hinreichend klein:

$$\Phi(t_\varepsilon, x_\varepsilon, s_\varepsilon, y_\varepsilon) \geq \max_{t,x} \Phi(t, x, t, x) = \max_{t,x} \left\{ u(t, x) - v(t, x) - 2\lambda t - 2\frac{\varepsilon}{2}|x|^2 \right\} \geq \frac{\sigma}{2}. \quad (4.29)$$

(a) Definiere die Testfunktion

$$\phi(t, x) := \left[v(s_\varepsilon, y_\varepsilon) + \frac{|t - s_\varepsilon|^2 + |x - y_\varepsilon|^2}{2\varepsilon} + \lambda(t + s_\varepsilon) + \frac{\varepsilon}{2}(|x|^2 + |y_\varepsilon|^2) \right].$$

Die Differenz $u(t, x) - \phi(t, x)$ besitzt wegen (4.28) bei $(t, x) = (t_\varepsilon, x_\varepsilon)$ ein Maximum. Daher gilt, weil $u(t, x)$ Viskositätsunterlösung ist:

$$\begin{aligned} \phi_t(t_\varepsilon, x_\varepsilon) + H(t_\varepsilon, x_\varepsilon, D\phi(t_\varepsilon, x_\varepsilon)) &\leq 0 \\ \Leftrightarrow \lambda + \frac{t_\varepsilon - s_\varepsilon}{\varepsilon} + H(t_\varepsilon, x_\varepsilon, \frac{x_\varepsilon - y_\varepsilon}{\varepsilon} + x_\varepsilon) &\leq 0. \end{aligned} \quad (4.30)$$

Als zweite Testfunktion setzt man

$$\psi(s, y) := \left[u(t_\varepsilon, x_\varepsilon) - \frac{|t_\varepsilon - s|^2 + |x_\varepsilon - y|^2}{2\varepsilon} - \lambda(t_\varepsilon + s) - \frac{\varepsilon}{2}(|x_\varepsilon|^2 + |y|^2) \right].$$

Analog besitzt $v(s, y) - \psi(s, y)$ bei $(s, y) = (s_\varepsilon, y_\varepsilon)$ ein Minimum. Daher gilt, weil $v(s, y)$ Viskositätsüberlösung ist:

$$\begin{aligned} \psi_t(s_\varepsilon, y_\varepsilon) + H(s_\varepsilon, y_\varepsilon, D\psi(s_\varepsilon, y_\varepsilon)) &\geq 0 \\ \Leftrightarrow -\lambda - \frac{t_\varepsilon - s_\varepsilon}{\varepsilon} + H(s_\varepsilon, y_\varepsilon, \frac{x_\varepsilon - y_\varepsilon}{\varepsilon} - y_\varepsilon) &\geq 0. \end{aligned} \quad (4.31)$$

(b) Wir subtrahieren (4.31) von (4.30) und setzen die Dreiecksungleichung ein:

$$\begin{aligned} 2\lambda &\leq H(s_\varepsilon, y_\varepsilon, \frac{x_\varepsilon - y_\varepsilon}{\varepsilon} - y_\varepsilon) - H(t_\varepsilon, x_\varepsilon, \frac{x_\varepsilon - y_\varepsilon}{\varepsilon} + x_\varepsilon) \\ &\leq \left| H(s_\varepsilon, y_\varepsilon, \frac{x_\varepsilon - y_\varepsilon}{\varepsilon} - y_\varepsilon) - H(s_\varepsilon, y_\varepsilon, \frac{x_\varepsilon - y_\varepsilon}{\varepsilon} + x_\varepsilon) \right| \\ &\quad + \left| H(s_\varepsilon, y_\varepsilon, \frac{x_\varepsilon - y_\varepsilon}{\varepsilon} + x_\varepsilon) - H(t_\varepsilon, x_\varepsilon, \frac{x_\varepsilon - y_\varepsilon}{\varepsilon} + x_\varepsilon) \right|. \end{aligned}$$

Anschließend wenden wir die Voraussetzungen (4.21a) und (4.21b) an und erhalten

$$\begin{aligned} 2\lambda &\leq C|-\varepsilon y_\varepsilon - \varepsilon x_\varepsilon| + \omega(|t_\varepsilon - s_\varepsilon| + |x_\varepsilon - y_\varepsilon|) \left(1 + \left| \frac{x_\varepsilon - y_\varepsilon}{\varepsilon} + \varepsilon x_\varepsilon \right| \right) \\ &\leq C\varepsilon(|y_\varepsilon| + |x_\varepsilon|) + \omega(|t_\varepsilon - s_\varepsilon| + |x_\varepsilon - y_\varepsilon|) \left(1 + \frac{|x_\varepsilon - y_\varepsilon|}{\varepsilon} + \varepsilon|x_\varepsilon| \right). \end{aligned}$$

Mit $|x_\varepsilon| \leq |x_\varepsilon| + |y_\varepsilon|$ gilt

$$2\lambda \leq C\varepsilon(|y_\varepsilon| + |x_\varepsilon|) + \omega(|t_\varepsilon - s_\varepsilon| + |x_\varepsilon - y_\varepsilon|) \left(1 + \frac{|x_\varepsilon - y_\varepsilon|}{\varepsilon} + \varepsilon(|x_\varepsilon| + |y_\varepsilon|) \right). \quad (4.32)$$

Um damit einen Widerspruch zu erhalten, beweisen wir im Folgenden, dass die rechte Seite von (4.32) für $\varepsilon \rightarrow 0$ gegen 0 konvergiert.

- (c) Aus der Ungleichung $\Phi(t_\varepsilon, x_\varepsilon, t_\varepsilon, x_\varepsilon) + \Phi(s_\varepsilon, y_\varepsilon, s_\varepsilon, y_\varepsilon) \leq 2\Phi(t_\varepsilon, x_\varepsilon, s_\varepsilon, y_\varepsilon)$ erhält man direkt durch das Einsetzen in die Definition (4.27)

$$\frac{|t_\varepsilon - s_\varepsilon|^2 + |x_\varepsilon - y_\varepsilon|^2}{\varepsilon} \leq u(t_\varepsilon, x_\varepsilon) - u(s_\varepsilon, y_\varepsilon) + v(t_\varepsilon, x_\varepsilon) - v(s_\varepsilon, y_\varepsilon). \quad (4.33)$$

Da u und v global beschränkt sind, gilt

$$\frac{|t_\varepsilon - s_\varepsilon|^2 + |x_\varepsilon - y_\varepsilon|^2}{\varepsilon} \leq C' \quad \text{mit } C' = 2 \max |u| + 2 \max |v|$$

$$\Rightarrow |t_\varepsilon - s_\varepsilon| + |x_\varepsilon - y_\varepsilon| \leq C'' \sqrt{\varepsilon} \quad \text{mit } C'' = \sqrt{2C'} \quad (4.34)$$

$$\Rightarrow \max\{|t_\varepsilon - s_\varepsilon|, |x_\varepsilon - y_\varepsilon|\} \leq C'' \sqrt{\varepsilon}. \quad (4.35)$$

Aufgrund der gleichmäßigen Stetigkeit von u und v existiert ein Stetigkeitsmodul ω , so dass

$$W(t, x) - W(s, y) = \omega(|t - s| + |x - y|) \quad \text{für } W(r) = \max\{\omega_u(r), \omega_v(r)\}. \quad (4.36)$$

Durch Kombination von (4.33), (4.34) und (4.36) erhalten wir

$$\frac{|t_\varepsilon - s_\varepsilon|^2 + |x_\varepsilon - y_\varepsilon|^2}{\varepsilon} \leq 2\omega(C'' \sqrt{\varepsilon}). \quad (4.37)$$

Da u und v global beschränkt sind, müssen die Strafterme einheitliche Beschränkungen, unabhängig von ε , besitzen. Dies folgt aus (4.29):

$$\begin{aligned} & \Phi(t_\varepsilon, x_\varepsilon, s_\varepsilon, y_\varepsilon) \geq \frac{\sigma}{2} \\ \Rightarrow & u(t_\varepsilon, x_\varepsilon) - v(s_\varepsilon, y_\varepsilon) - \lambda(t_\varepsilon + s_\varepsilon) - \frac{\varepsilon}{2}(|x_\varepsilon|^2 + |y_\varepsilon|^2) \geq \frac{\sigma}{2} \\ \Rightarrow & \frac{\varepsilon}{2}(|x_\varepsilon|^2 + |y_\varepsilon|^2) \leq \underbrace{\sup u(t_\varepsilon, x_\varepsilon) - \inf v(s_\varepsilon, y_\varepsilon) - \frac{\sigma}{2}}_{=:\bar{C}>0} \\ \Rightarrow & (|x_\varepsilon| + |y_\varepsilon|) \leq \frac{\bar{C}}{\sqrt{\varepsilon}} \quad \text{mit } \bar{C} = 2\sqrt{\bar{C}} \\ \Rightarrow & \max\{|x_\varepsilon|, |y_\varepsilon|\} \leq \frac{\bar{C}}{\sqrt{\varepsilon}}. \end{aligned}$$

Somit gilt:

$$\frac{\varepsilon}{2}(|x_\varepsilon| + |y_\varepsilon|) \leq \bar{C} \sqrt{\varepsilon}. \quad (4.38)$$

Durch (4.37) und (4.38) erhalten wir, dass die rechte Seite von (4.32) für $\varepsilon \rightarrow 0$ gegen 0 konvergiert. Dies führt zum erwünschten Widerspruch. □

Bemerkung 4.12

Die Eindeutigkeit folgt direkt aus dem Vergleichsprinzip (Satz 4.10) mit $u(0, x) = v(0, x)$.

4.3 Zusammenhang mit Erreichbarkeitsmengen

Um die Erreichbarkeitsmenge als Niveaumenge der Lösung der HJ-PDE bestimmen zu können, verwendet man folgende Korrelation zwischen der Erreichbarkeitsmenge und der Wertefunktion, welche die Viskositätslösung der HJ-PDE darstellt.

Behauptung 4.13

Die Erreichbarkeitsmenge $\mathcal{R}(\tau)$ zum Zeitpunkt τ entspricht der Niveaumenge der Lösung der HJ-PDE zum Zeitpunkt τ zum Niveau $\mathbf{0}$:

$$\mathcal{R}(\tau; t_0, x_0) = \left\{ x \in \mathbb{R}^n \mid \tilde{V}(\tau, x) \leq 0 \right\} \quad (4.39)$$

Beweis:

Betrachte $\tilde{V}(\tau, x) = \min_u \{ d^2(y(t_0), x_0) \mid y(\tau) = x \}$ mit zugehörigem Optimierungsproblem $\tilde{\mathcal{I}}_{\tau, x}$

$$\tilde{\mathcal{I}}_{\tau, x} \left\{ \begin{array}{l} \text{Minimiere } \|y(t_0) - x_0\|_2^2, \\ \text{s.d. } \dot{x}(t) = f(t, y(t), u(t)) \quad \text{f.ü.} \\ u(t) \in U \quad \text{f.ü.} \\ y(\tau) = x, \end{array} \right.$$

so dass $\tilde{V}(\tau, x) = \inf \tilde{\mathcal{I}}_{\tau, x}$.

Dieses Optimierungsproblem minimiert über den Abstand der Anfangswerte der Trajektorien mit Endwert x zum Zeitpunkt τ .

Zeigen wir nun die Gleichheit von Gleichung (4.39), indem wir beide Inklusionen zeigen: „f“, g

„ \supset “: Sei $x \in \mathbb{R}^n$ mit $\tilde{V}(\tau, x) \leq 0$ zum Zeitpunkt $\tau \in [t_0, T]$. Damit gilt nach der Definition von $\tilde{V}(t, x)$:

$$0 \geq \tilde{V}(\tau, x) = \min_{u \in U} \{ \|y(t_0) - x_0\|_2^2 \mid y(\tau) = x \}$$

\Rightarrow es existiert eine zulässige Lösung \tilde{y} und eine zulässige Kontrolle \tilde{u} mit

$$0 = \|\tilde{y}(t_0) - x_0\|_2^2 \text{ und } \tilde{y}(\tau) = x \text{ (da } \tilde{V}(\tau, x) \text{ nach Satz 3.4 stetig)}$$

\Rightarrow Für $\tilde{y}(\cdot)$ gilt: $\tilde{y}(t_0) = x_0$ und $\tilde{y}(\tau) = x$ mit \tilde{u} zulässig

$\Rightarrow \tilde{y}(\tau) = x \in \mathcal{R}(\tau; t_0, x_0)$.

„ \subset “: Sei $x \in \mathcal{R}(\tau; t_0, x_0)$.

$\Rightarrow \exists \tilde{y}(\cdot), \tilde{u}(\cdot)$ zulässig mit $\tilde{y}(t_0) = x_0$ und $\tilde{y}(\tau) = x$

$\Rightarrow \|\tilde{y}(t_0) - x_0\|_2^2 = 0$

$\Rightarrow \min_{u \in U} \|\tilde{y}(t_0) - x_0\|_2^2 \leq \|\tilde{y}(t_0) - x_0\|_2^2 = 0$

$\Rightarrow \tilde{V}(\tau, x) \leq 0$.

□

5 Numerische Berechnung von Erreichbarkeitsmengen

5.1 Verwendetes Verfahren

In diesem Kapitel soll auf das Verfahren eingegangen werden, welches in dieser Arbeit verwendet wurde, um mit der Software der ITN-SADCO-Gruppe die Beispiele zu analysieren und auszuwerten. Generell wird die Software nur als Blackbox verwendet, da der Quellcode der eigentlichen Hamilton-Jacobi-Bibliothek nicht vorliegt, sondern vorkompiliert gegeben ist. Auch wird auf numerische Analysen weitgehend verzichtet. Für genauere Untersuchungen sei daher auf [BFZ10] und [OF01] verwiesen, dennoch soll an dieser Stelle ein kurzer Einblick über das in dieser Arbeit verwendete Verfahren gegeben werden.

Ausgehend von [BFZ10, S. 4303] betrachten wir hier zur Vereinfachung das Hamiltonfunktional ohne explizite Zeitabhängigkeit. Auch orientiert sich der folgende Abschnitt sehr stark an der Betrachtung in [BFZ10, S. 4303]. Zur Approximation der Lösung der HJ-PDE wird ein Finite-Differenzen-Schema verwendet.

Das Hamilton-Funktional H wird als stetige Funktion angenommen, welche die an die Zeitunabhängigkeit angepassten und geringfügig erweiterten Voraussetzungen des Vergleichsprinzips (Satz 4.10 auf Seite 36) erfüllt. Explizit wird gefordert, dass das Hamilton-Funktional H die Bedingungen

$$|H(x, p) - H(x, q)| \leq C |p - q| \quad (5.1a)$$

$$|H(x, p) - H(y, p)| \leq C |x - y| (1 + |p|) \quad (5.1b)$$

für eine Konstante $C \geq 0$ und für alle x, y, p, q in \mathbb{R}^n erfüllt. Ausgehend von einem Gitter $\mathcal{G} := \{I\Delta x, I \in \mathbb{Z}^n\}$, welches mit Hilfe der Gittergrößen $\Delta x, \Delta t$ definiert ist, betrachtet man den diskreten Punkt (x_I, t_d) mit $x_I = I\Delta x$ und $t_d = d\Delta t$. Die Approximation der Wertefunktion V wird im Folgenden als ϑ bezeichnet, wobei ϑ entweder als Folge ϑ_I^d oder als Gitterpunkt $\vartheta(x_I, t_d)$ verwendet wird.

Betrachtet man das numerische Hamilton-Funktional $H_{num} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$, eine konsistente Approximation des analytischen Hamilton-Funktional, wird folgendes Schema angewandt:

$$\begin{cases} \min \left(\frac{\vartheta_I^{d+1} - \vartheta_I^d}{\Delta t} + H_{num}(x_I, D^+ \vartheta^d(x_I), D^- \vartheta^d(x_I)), \vartheta_I^{d+1} \right) = 0 \\ \vartheta_I^0 = \tilde{u}_0(x_I), \end{cases}$$

mit \tilde{u}_0 als Approximation von V_0 der Startmenge des Verfahrens und

$$\begin{aligned} D^+ \vartheta^d(x_I) &= (D_{x_1}^+ \vartheta^d(x_I), \dots, D_{x_n}^+ \vartheta^d(x_I)), \\ D^- \vartheta^d(x_I) &= (D_{x_1}^- \vartheta^d(x_I), \dots, D_{x_n}^- \vartheta^d(x_I)) \end{aligned}$$

als die Gradienten des diskreten Raumes der Funktion ϑ^d am Punkt x_I . Diese sind für die allgemeine Funktion ω folgenderweise definiert

$$D_{x_i}^\pm \omega(x_I) = \pm \frac{\omega(x_{I^{i,\pm}}) - \omega(x_I)}{\Delta x},$$

mit der Notation $I^{i,\pm} = (i_1, \dots, i_{k-1}, i_k \pm 1, i_{k+1}, \dots, i_n)$. Unter einigen Annahmen (auf diese wird an dieser Stelle nur hingewiesen, für eine explizite Betrachtung siehe (H4)-(H7) in [BFZ10, S. 4304]) an das numerische Hamilton-Funktional H_{num} , wie zum Beispiel Erfüllen einer Monotonie-Bedingung und einer Konsistenz-Bedingung etc., lässt sich eine Fehlerabschätzung für das Verfahren liefern. Dazu muss aber noch sichergestellt werden, dass das Verfahren

$$\begin{cases} \min \left(\frac{\vartheta_I^{d+1} - \vartheta_I^d}{\Delta t} + H_{num}(x_I, D^+ \vartheta^d(x_I), D^- \vartheta^d(x_I)), \vartheta_I^{d+1} \right) = 0 \\ \vartheta_I^0 = \tilde{u}_0(x_I), \end{cases} \quad (5.2)$$

auch wirklich monoton ist. Dies gelingt über die Hinzunahme der sogenannten CFL (Courant-Friedrichs-Lewy)-Bedingung

$$\frac{\Delta t}{\Delta x} \sum_{i=1}^n \left(\left| \frac{\partial H_{num}}{\partial P_i^-}(x, P^+, P^-) \right| + \left| \frac{\partial H_{num}}{\partial P_i^+}(x, P^+, P^-) \right| \right) \leq 1, \quad (5.3)$$

fast überall mit $(x, P^+, P^-) \in \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n$.

Diese stellt sicher, dass das numerische Verfahren nicht explodiert (siehe folgende Kapitel). Desweiteren kann diese Bedingung mittels (H5) ([BFZ10, S. 4304]) noch stark vereinfacht werden, sodass die CFL-Bedingung erfüllt ist, sobald $\frac{\Delta t}{\Delta x} \leq \frac{1}{2nC_2}$ mit dem C_2 aus (H5). (H5) entspricht dabei einer Übertragung von (5.1a) auf das numerische Hamilton-Funktional. Für die numerische Approximation existiert damit dann auch eine Fehlerabschätzung, auf die hier jedoch nicht genauer eingegangen wird. Für die exakte Aussage und ihren Beweis siehe Theorem 5 in [BFZ10, S. 4305].

Im Folgenden sollen nun einige Beispiele analysiert werden, welche mit Hilfe des Programmes der ITN-SADCO-Gruppe berechnet und visualisiert worden sind. Hierbei wird ausschließlich auf die graphischen Ergebnisse eingegangen und keine Fehlerbetrachtung vorgenommen. Bevor die Beispiele betrachtet werden, wird nun zuerst eine kleine Einführung in die Ausgabe des Programmes gegeben. Die Software an sich berechnet die Wertefunktion der Hamilton-Jacobi-Bellman-Gleichung auf einem Gitter. Diese lässt sich mit der Plot-Funktion `output_view.m` in einer 3-dimensionalen Interpolation der Wertefunktion anzeigen. Zusätzlich zu diesem 3-dimensionalen Abbild bildet `output_view.m` noch ein Level-Set der Wertefunktion zum Level 0 ab (numerisch wird hier 0.0001 betrachtet). Als Ausgabe erhält man hier, wie beispielsweise in Abb. 5.5 zu sehen, eine Menge blau gefärbter Punkte, welche die Menge der Gitterpunkte innerhalb des Level-Sets darstellt, sowie - optional - eine Kontur-Interpolation der Grenze dieser Menge (rot).

5.2 Beispiel 1

Betrachte das zeitinvariante 2-dimensionale affine Kontrollsystem⁵:

$$\begin{aligned} \dot{y}(t) &= \underbrace{Ay(t) + Bu(t)}_{f(y(t),u)}, \text{ f.ü.} \\ u(t) &\in [-1, 1], \quad t \in [0, 2\pi] \\ y(0) &= \begin{pmatrix} 0 \\ 0 \end{pmatrix} \end{aligned}$$

mit $A = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ und $B = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$.

Dieses Beispiel beschreibt die Rotation und gleichzeitige Streckung einer Strecke. Dies lässt sich durch Erklärung der einzelnen Bestandteile der Differentialgleichung veranschaulichen. Ausgangspunkt bildet der Steuerbereich, welcher als Intervall gegeben ist. Durch die Multiplikation mit B wird dieses Intervall zu einer Strecke, welche durch den Additionsterm eine Rotation um den Ursprung erfährt. Nach Durchlauf des Zeitintervalls $[0, 2\pi]$ ergibt sich somit ein Kreis um den Ursprung. Diese Entwicklung ist in den Abbildungen 5.1 - 5.4 dargestellt.

Hinweis:

Generell gibt es für affine Kontrollsysteme effizientere Methoden zur Berechnung von Erreichbarkeitsmengen als den Hamilton-Jacobi-Ansatz. Dieses Beispiel wurde nur als erstes akademisches Testbeispiel gewählt. Für die Simulationen in diesem Beispiel wurden folgende Variablen-Werte verwendet (siehe Tabelle 5.1):

Variiert werden, falls keine anderen Parameter explizit genannt werden, nur die Anzahl der Gitterpunkte pro Koordinatenrichtung NN . Je nach Betrachtung werden zusätzlich ohne explizite Nennung auch Parameter zur Berechnung der exakten Lösung gesetzt, welche jedoch aufgrund ihres Charakters vornehmlich optische Auswirkungen haben.

Betrachten wir in Abbildung 5.5 zuerst eine numerische Approximation der Erreichbarkeitsmenge mit relativ hoher räumlicher Gitterfeinheit - die zeitliche Diskretisierung wird vom Programm in diesem Fall selbst gewählt, um die CFL-Bedingung (siehe Gleichung (5.3) auf Seite 44) zu erfüllen. Diese erreichbare Menge soll uns hier als Referenzmenge dienen, so dass wir diese mit Approximationen mit anderen Parametern vergleichen können. Zur Berechnung dieser Referenzmenge wurde die Gitterfeinheit von $NN = 100$ Gitterpunkten pro Koordinatenrichtung gewählt, die zeitliche Diskretisierung ist automatisch vom Algorithmus zum Einhalten der CFL-Bedingung (siehe Gleichung (5.3) auf Seite 44) auf $DT = 0.00753$ festgelegt worden. Setzt man die räumliche Diskretisierung auf weniger Gitterpunkte pro Koordinatenrichtung lässt sich nur eine deutlich gröbere Approximation der erreichbaren Menge

⁵siehe Beispiel 5.1.1 in [Cha03]

Tabelle 5.1: Feste Parameter für Beispiel 1

Variable	Wert
METHOD	MFD
TYPE_SCHEME	ENO2
TYPE_RK	RK1
T	$2*\pi$
DT	0.
CFL	0.8
XMIN [DIM]	{ -8. , -8. }
XMAX [DIM]	{ 8. , 8. }
MESH	1
cDIM	1
NCD [cDIM]	{ 2 }
UMIN [cDIM]	{ -1. }
UMAX [cDIM]	{ 1. }
BOUNDARY	1
VBORD	0.4

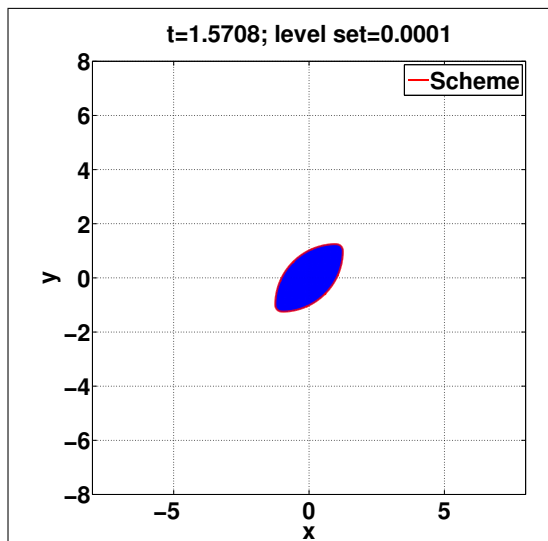


Abb. 5.1:
Erreichbare Menge mit $NN=800$ und adaptiver zeitlicher Diskretisierung zum Zeitpunkt $T=0.5\pi$.

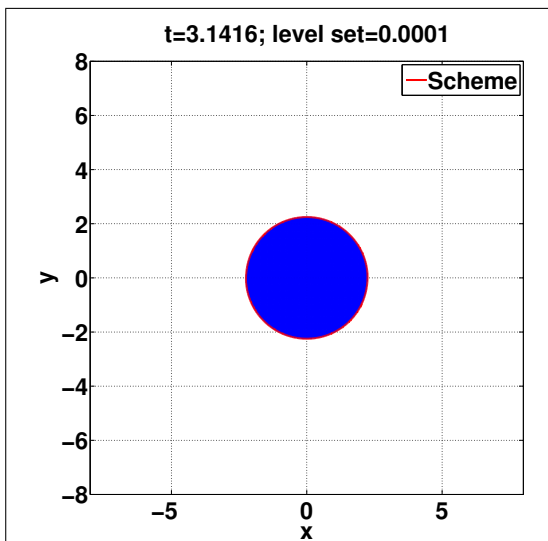


Abb. 5.2:
Erreichbare Menge mit $NN=800$ und adaptiver zeitlicher Diskretisierung zum Zeitpunkt $T=\pi$.

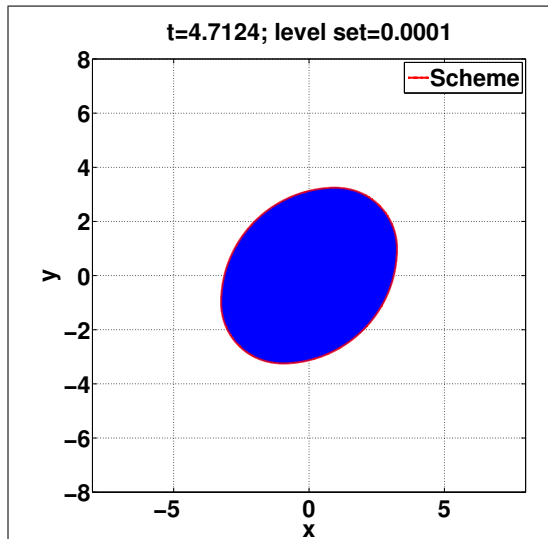


Abb. 5.3:
Erreichbare Menge mit $NN=800$ und adaptiver zeitlicher Diskretisierung zum Zeitpunkt $T=1.5\pi$.

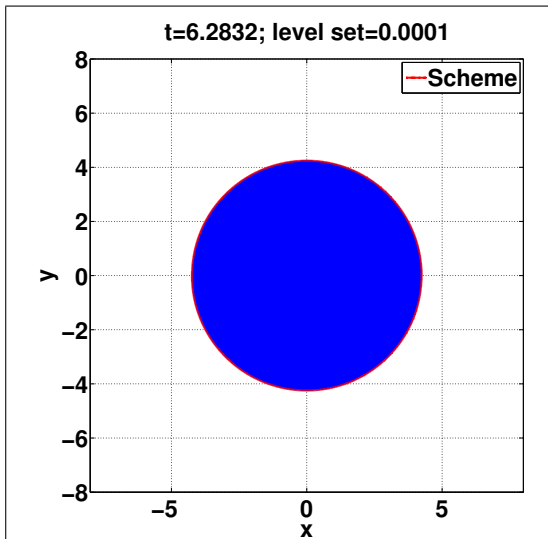
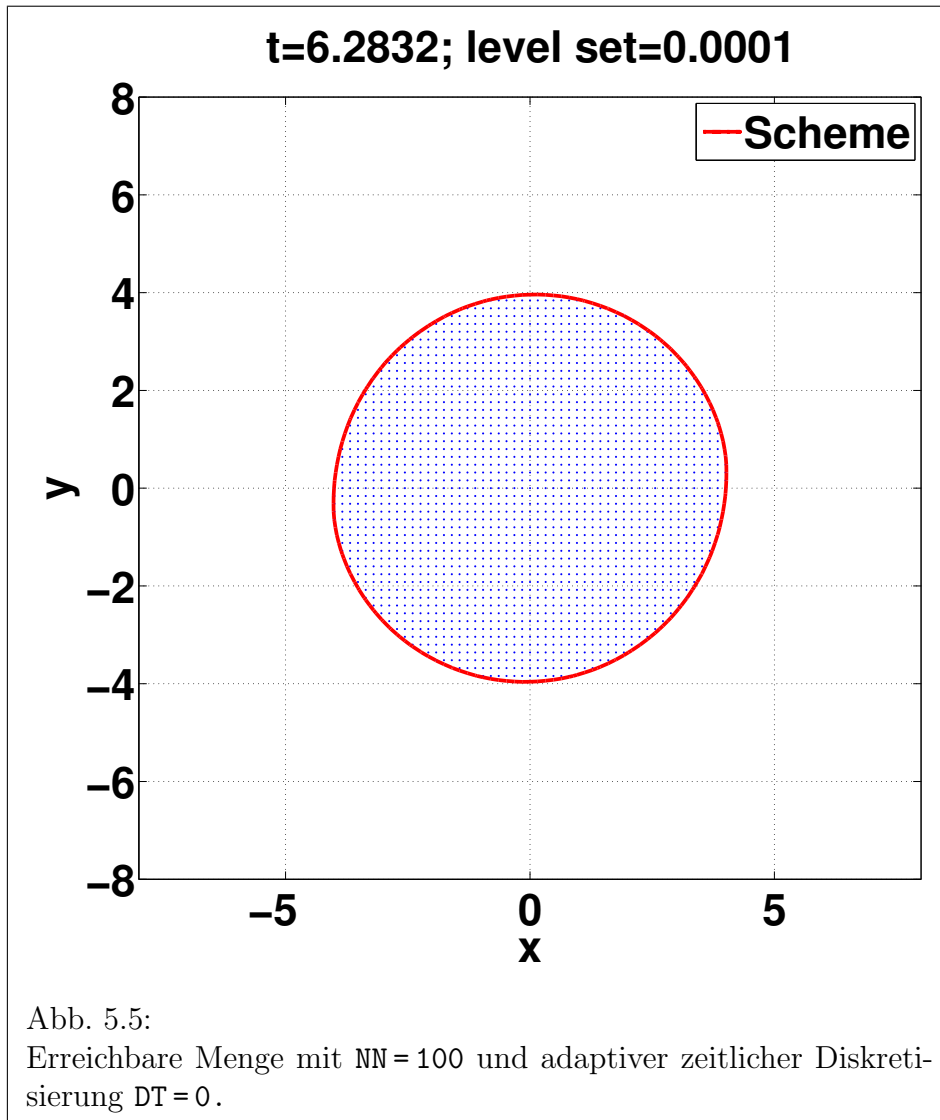
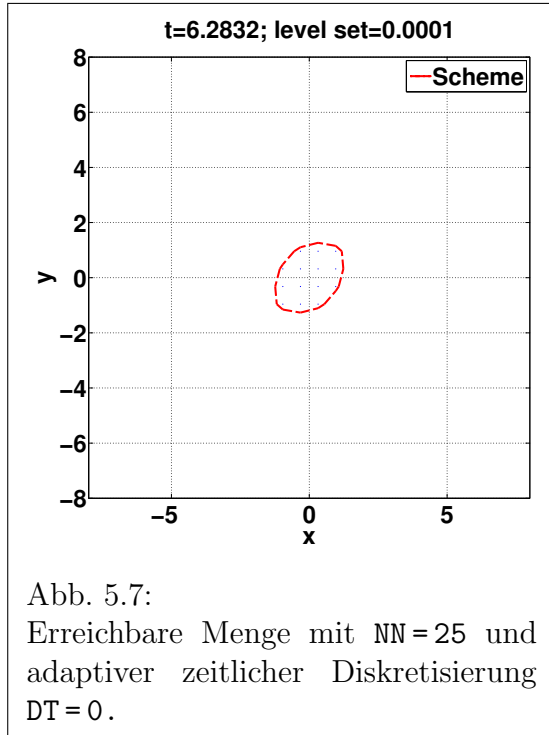
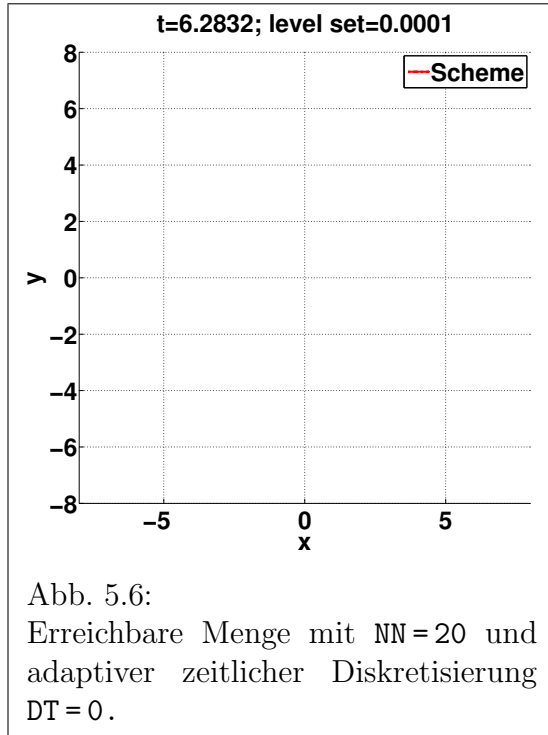


Abb. 5.4:
Erreichbare Menge mit $NN=800$ und adaptiver zeitlicher Diskretisierung zum Zeitpunkt $T=2\pi$.





erkennen (siehe Abb. 5.7 und Abb. 5.8). Die berechnete Approximation ist kleiner als die Referenzmenge, da durch das grobe Gitter der räumliche Fehler größer wird. Es werden nicht mehr alle Punkte der erreichbaren Menge durch den Algorithmus abgedeckt. Bei einer Anzahl von 20 Gitterpunkten pro Koordinatenrichtung und weniger verstärkt sich dieser Effekt, so dass keine Punkte der erreichbaren Menge abgedeckt werden. Somit bleibt die erreichbare Menge leer (siehe Abb. 5.6).

Bei einer Gitterfeinheit von $NN=50$ (siehe Abb. 5.8) lässt sich die Form der erreichbaren Menge tendenziell schon erkennen, jedoch sehen wir auch, dass die erreichbare Menge nicht vollständig erreicht wird. Die Gestalt nähert sich bereits der Kreisform der Referenzmenge an, zeigt jedoch noch immer ellipsoide Tendenzen. Stärker tritt dieser Effekt beispielsweise bei $NN=25$ auf.

Setzt man nun die Anzahl der Gitterpunkte auf $NN=800$, so wird die Referenzmenge vollständig überlappt (siehe Abb. 5.9).

In Tabelle 5.2 sieht man die Eingabeparameter NN , den automatisch bestimmten Parametern DT und NT gegenüber gestellt. Für die Ausführungszeit ergibt sich in Abhängigkeit des Eingabeparameters NN ein potentiell, annähernd kubisches Wachstum.

Im Folgenden wird nun ein bereits in der Software vorimplementiertes Beispiel `basicmodel.h` in mehreren Abschnitten unter verschiedenen Gesichtspunkten vorgestellt und analysiert. Auf diese Weise lassen sich verschiedene Phänomene demonstrieren, welche bei der Verwendung der Software auftreten können.

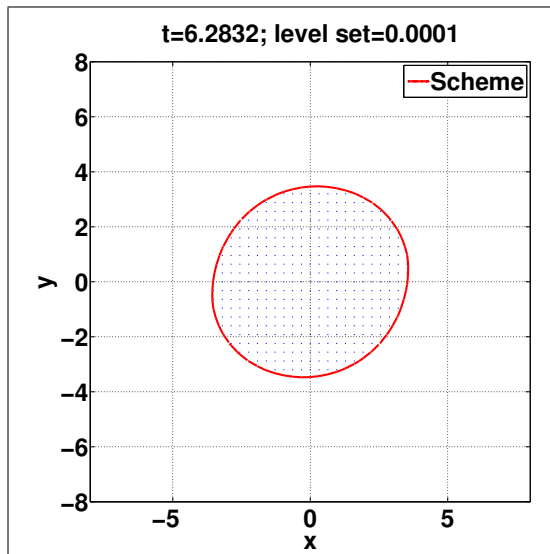


Abb. 5.8:
Erreichbare Menge mit $NN=50$ und adaptiver zeitlicher Diskretisierung $DT=0$.

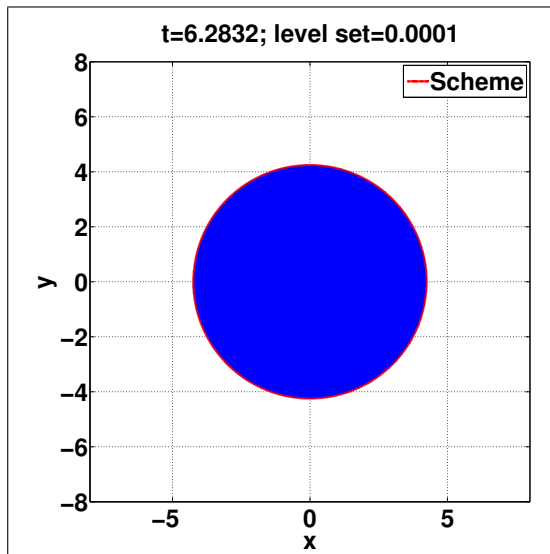


Abb. 5.9:
Erreichbare Menge mit $NN=800$ und adaptiver zeitlicher Diskretisierung $DT=0$.

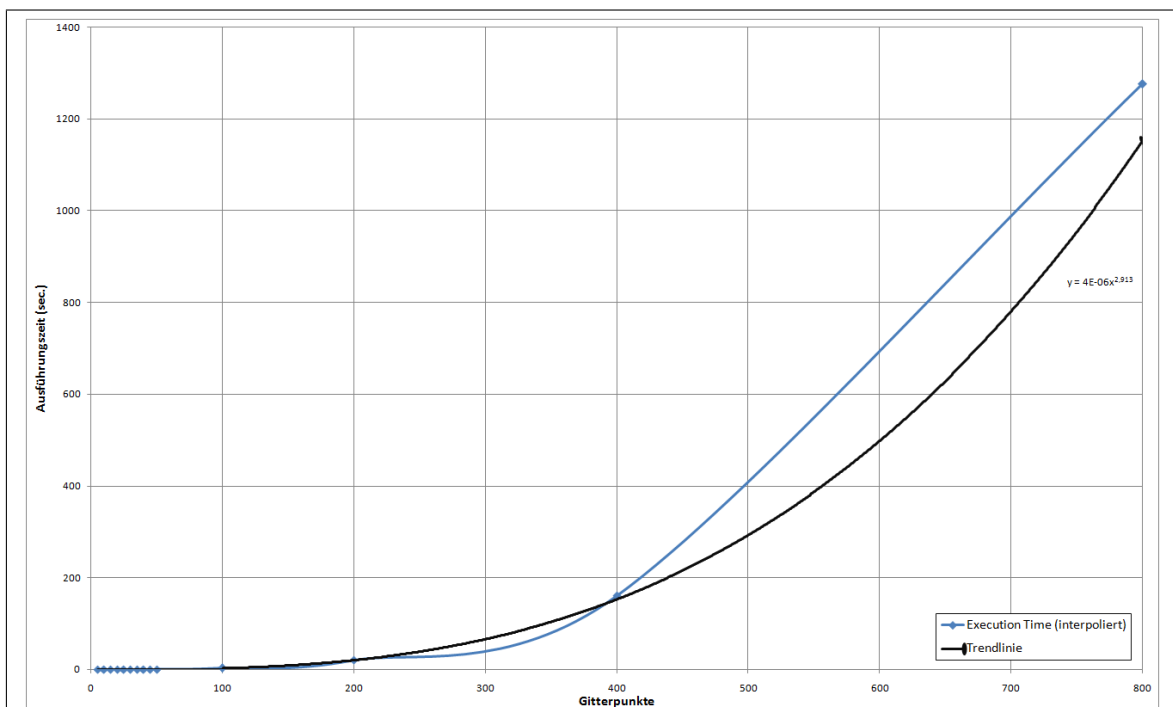


Abb. 5.10:
Ausführungszeit Beispiel 1 in Abhängigkeit der Anzahl der Gitterpunkte (Interpolation und exponentielle Trendlinie).

Tabelle 5.2: Simulationsauswertung für Beispiel 1

NN (Anzahl Gitterpunkte)	DT (Zeitschritt)	NT (Anzahl Zeitschritte)	ExecTime (Ausführungszeit)
5	0.15059	42	0.000569 sec.
10	0.07529	84	0.003945 sec.
15	0.05020	126	0.010345 sec.
20	0.03765	167	0.021395 sec.
25	0.03012	209	0.041118 sec.
30	0.02510	251	0.070261 sec.
35	0.02151	293	0.107423 sec.
40	0.01882	334	0.160875 sec.
45	0.01673	376	0.217347 sec.
50	0.01506	418	0.479873 sec.
100	0.00753	835	2.89794 sec.
200	0.00376	1669	20.3462 sec.
400	0.00188	3338	160.863 sec.
800	0.00094	6676	1275.62 sec.

5.3 Beispiel 2

Dieses Beispiel behandelt eine Gleichung, welche auf einfache Weise eine advektions-ähnliche Rotation modelliert. An diesem Beispiel lässt sich sehr gut die „Manipulation“ der Problemstellung erkennen, welche nötig ist, um die Vereinigung der erreichbaren Mengen

$$\mathcal{R}_{union}(T; t_0, x_0) := \bigcup_{t \in [t_0, T]} \mathcal{R}(t; t_0, x_0)$$

statt der erreichbaren Menge zum Endzeitpunkt T

$$\mathcal{R}(T; t_0, x_0)$$

zu erhalten.

Ausgehend von dem 2-dimensionalen ungesteuerten Differentialgleichungssystem

$$\begin{aligned} \dot{x} &= f(x) \\ &\text{mit } f(x_1, x_2) = 2\pi(-x_2, x_1) \text{ und} \\ x_0 &= \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \end{aligned} \tag{5.4}$$

welches folgende Hamilton-Jacobi-Gleichung [BDZ13b, S. 9] liefert:

$$\begin{cases} \frac{\partial u}{\partial t} + \max(0, -f(x) \cdot \nabla u) = 0 \\ u(t, x) = u_{bord}, \quad x \in \nabla\Omega, \quad t \in [0, T] \\ u(0, x) = u_0(x), \quad x \in \Omega, \end{cases} \tag{5.5}$$

erhält man durch Einführung einer virtuellen multiplikativen 1-dimensionalen Kontrolle mit den Kontrollwerten $a \in \{0, 1\}$, also der Manipulation der Dynamik zu

$$f(x, a) = a f(x), \tag{5.6}$$

ein neues gesteuertes Kontrollproblem:

$$\begin{aligned} \dot{x} &= a f(x) \\ &\text{mit } f(x_1, x_2) = 2\pi(-x_2, x_1), \\ &a \in \{0, 1\} \text{ und} \\ x_0 &= \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \end{aligned} \tag{5.7}$$

Die zugehörige Hamilton-Jacobi-Gleichung lautet dann

$$\begin{cases} \frac{\partial u}{\partial t} + \max_a(-f(x, a) \cdot \nabla u) = 0 \\ u(t, x) = u_{bord}, \quad x \in \nabla\Omega, \quad t \in [0, T] \\ u(0, x) = u_0(x), \quad x \in \Omega, \end{cases} \tag{5.8}$$

mit den Parametern:

- Gebiet $\Omega = [-2, 2]^2$,
- Endzeitpunkt $T = 0.5$ und
- betrachtetem Vektorfeld $f(x_1, x_2) = 2\pi(-x_2, x_1)$.
- Anfangsmenge u_0 entspricht $B_{0.25}(1, 0)$ (Ball mit Radius 0.25 um den Punkt (1,0)).

Somit lässt sich auf sehr einfache Weise wahlweise die Vereinigung der erreichbaren Mengen oder die erreichbare Menge an sich berechnen, indem man eine einfache Manipulation am Vektorfeld vornimmt.

Für die Erstellung der Graphiken wurden hier folgende Parameter verwendet (siehe Tabelle 5.3)

Tabelle 5.3: Feste Parameter für Beispiel 2

Variable	Wert
METHOD	MFD
TYPE_SCHEME	ENO2
TYPE_RK	RK1
T	0.75
DT	0.
CFL	0.8
XMIN [DIM]	{ -2. , -2. }
XMAX [DIM]	{ 2. , 2. }
MESH	0
cDIM	1
NCD [cDIM]	{ 2 }
UMIN [cDIM]	{ 0. }
UMAX [cDIM]	{ 1. }
BOUNDARY	1
VBORD	0.4

In den Abbildungen 5.11 und 5.12 lässt sich der Unterschied deutlich erkennen. Die Vereinigung der erreichbaren Menge (siehe Abb. 5.11), betrachtet über das gesamte

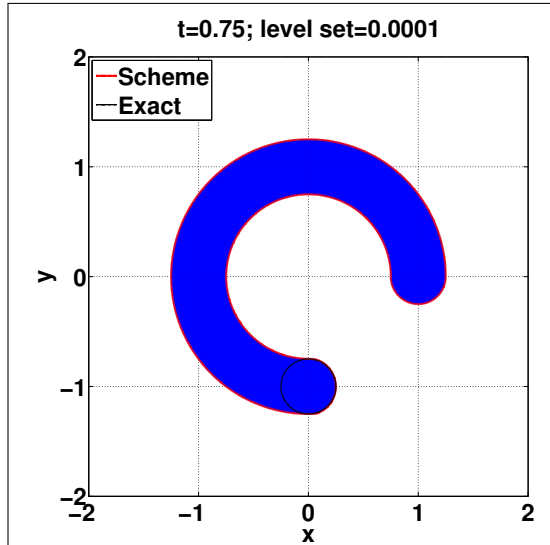


Abb. 5.11:
Vereinigung der erreichbaren Mengen bis zum Zeitpunkt T (inklusive Plot der exakten Lösung) mit $NN = 800$ und adaptiver zeitlicher Diskretisierung $DT = 0$.

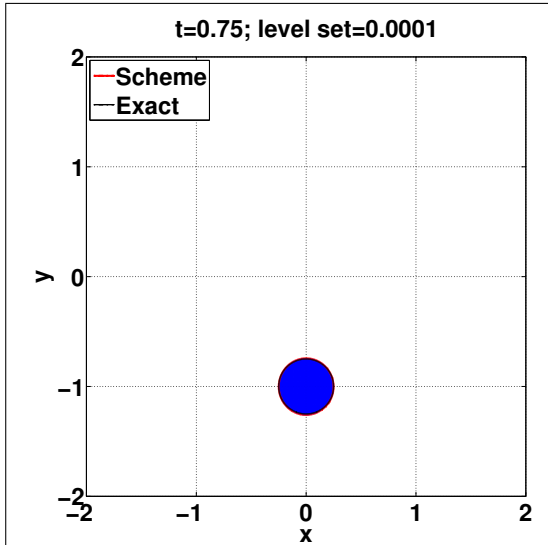
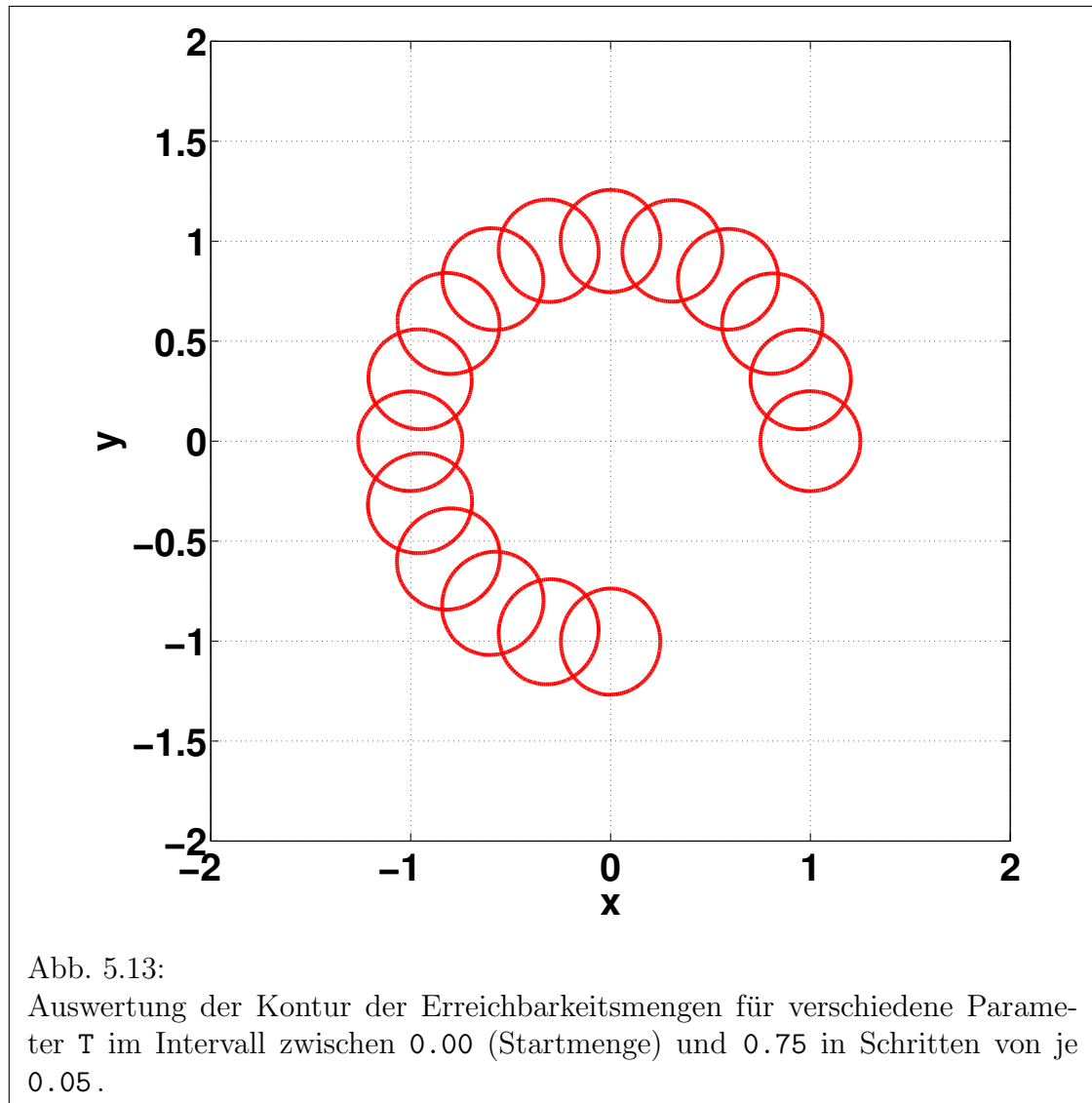


Abb. 5.12:
Erreichbare Menge zum Endzeitpunkt T (inklusive Plot der exakten Lösung) mit $NN = 800$ und adaptiver zeitlicher Diskretisierung $DT = 0$.

Zeitintervall $[0, T]$ mit $T = 0.75$, zeigt die vollständige Rotation der Erreichbarkeitsmenge um den Ursprung, während die zweite Grafik (siehe Abb. 5.12) lediglich die Erreichbarkeitsmenge zum Endzeitpunkt $T = 0.75$ abbildet. Durch die hohe räumliche Diskretisierung ($NN = 800$) und die automatisch gewählte zeitliche Diskretisierung, entsprechend der CFL-Bedingung (siehe Gleichung (5.3) auf Seite 44), lässt sich auch hier im Vergleich mit der exakten Lösung der Wertefunktion⁶ bereits eine sehr gute Approximation erkennen. Die numerisch berechnete Erreichbarkeitsmenge beinhaltet die exakte Erreichbarkeitsmenge vollständig, überlappt sie aber in vertikaler Ausrichtung (siehe Abb. 5.14). Somit ist hier ein numerischer Fehler graphisch erkennbar. Die Abbildung zeigt zur besseren Erkennbarkeit eine Einschränkung des Gebiets auf $[-0,5, 0,5] \times [-1,5, -0,5]$. Die verwendeten Parameter stimmen mit den aus Abbildung 5.12 überein. Betrachtet man die Auswertung der Erreichbarkeitsmenge für verschiedene Endzeitpunkte T , lässt sich anschaulich der Übergang der Erreichbarkeitsmenge zur Vereinigung der Erreichbarkeitsmengen (siehe Abb. 5.13) erkennen.

Da wir nun die Referenzmenge kennen und auch graphisch für dieses Beispiel belegt haben, dass der angewandte Algorithmus der ITN SADCO-Toolbox die Referenzmenge bei hoher räumlicher Diskretisierung erreicht und sogar überdeckt (numerischer Fehler), betrachten wir nun auch weniger dichte räumliche Diskretisierungen.

⁶Die exakte Wertefunktion ist hier als $B((0, -1), 0.25)$ gegeben.



Dabei behalten wir zunächst das angewandte Approximationsverfahren, welches bereits in den genannten Simulationen verwendet wurde (räumliche Diskretisierung nach ENO2-Verfahren und zeitliche Diskretisierung RK1) bei. Im Folgenden wird zur graphischen Untersuchung wieder die Vereinigung der erreichbaren Mengen betrachtet, da hier die Entwicklung der erreichbaren Menge in Abhängigkeit der räumlichen Diskretisierung besser beobachtet werden kann. Bei geringer räumlicher Diskretisierung $MN = 10$ (siehe Abb. 5.15) ist keine Ähnlichkeit der berechneten erreichbaren Menge mit der zu erwartenden Erreichbarkeitsmenge (siehe Abb. 5.11) zu erkennen. Die Simulation deckt im Prinzip nur die Anfangsmenge ab (diskretisiert). Schon mit einer geringfügig höheren Diskretisierung von $MN = 15$ Gitterpunkten pro Koordinatenrichtung (siehe Abb. 5.16) entwickelt sich die numerisch berechnete Vereinigung der Erreichbarkeitsmengen auf dem Intervall $[0, T]$ deutlich weiter und man erkennt

den ersten Anschein einer Rotation um den Ursprung. Allerdings zeigt auch diese Graphik nur eine relativ schlechte Approximation des Level-Sets, was auf die sehr grobe Diskretisierung zurückzuführen ist. Diese Rotationsbewegung wird mit zunehmender räumlicher Diskretisierung (siehe Abb. 5.17 - 5.21) immer besser erkennbar, der Plot nimmt immer mehr die Gestalt eines um den Ursprung rotierenden Balles an. Anfangs- und Endstücke werden zunehmend abgerundet, bis schließlich bei $NN = 100$ eine sehr große Ähnlichkeit der Approximation mit der Referenzmenge bei $NN = 800$ (siehe Abb. 5.11) vorliegt.

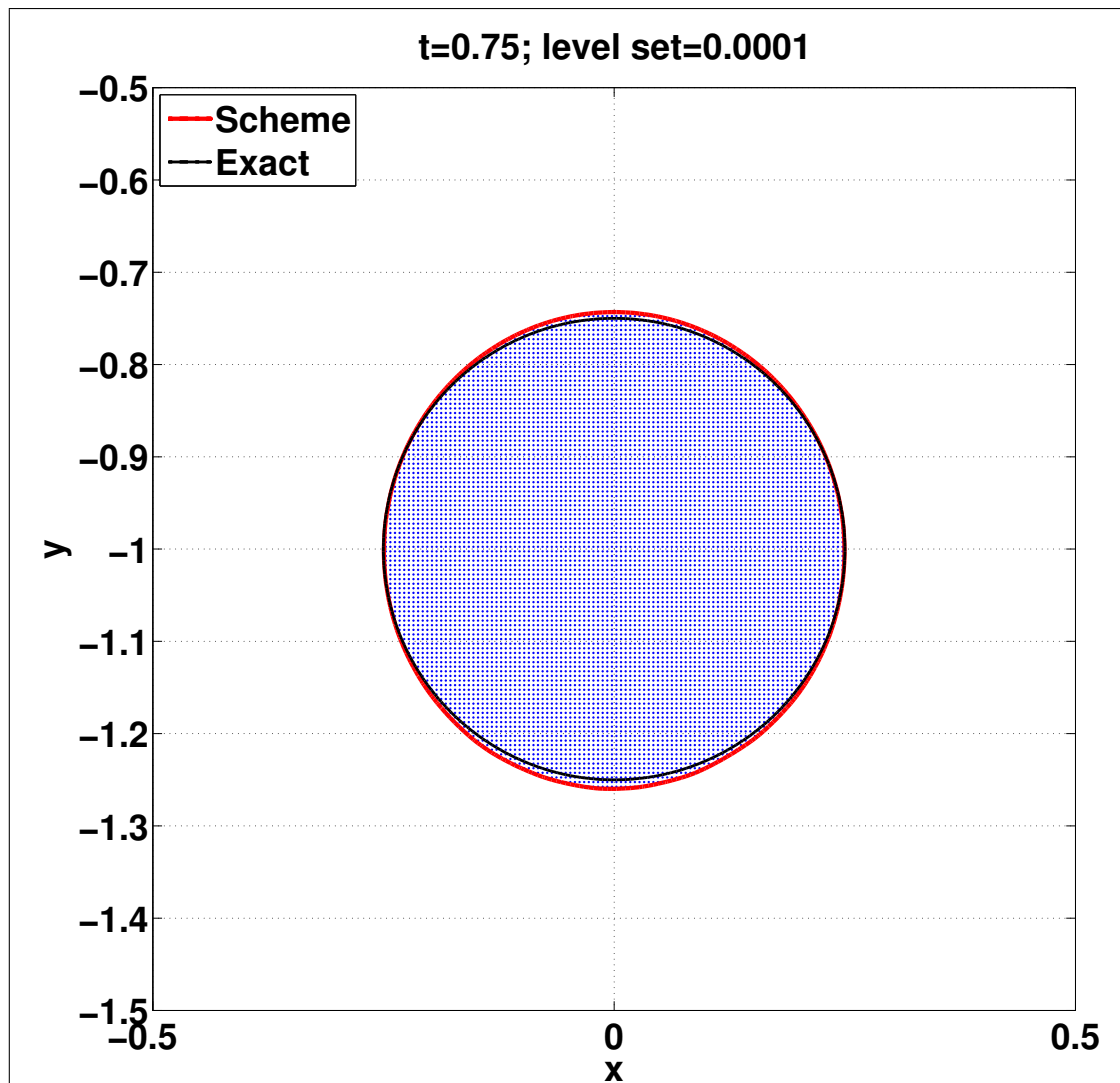


Abb. 5.14:
Vergleich des exakten Level-Sets der Wertefunktion mit der numerischen Approximation (NN = 800 und adaptive zeitliche Diskretisierung $DT = 0.$).

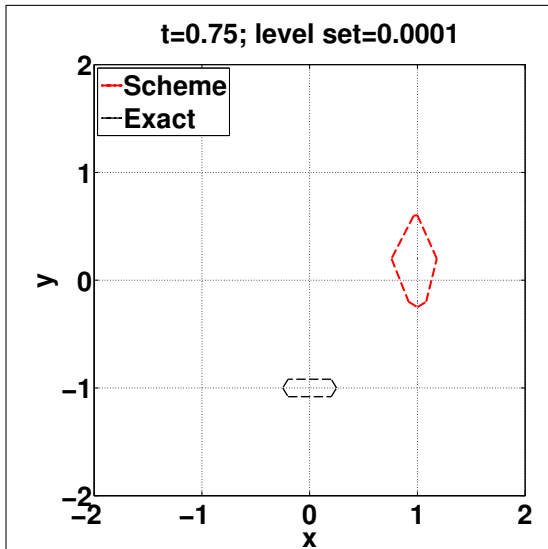


Abb. 5.15:
Vereinigung der erreichbaren Mengen bis zum Zeitpunkt T mit $NN=10$ und adaptiver zeitlicher Diskretisierung $DT=0$.

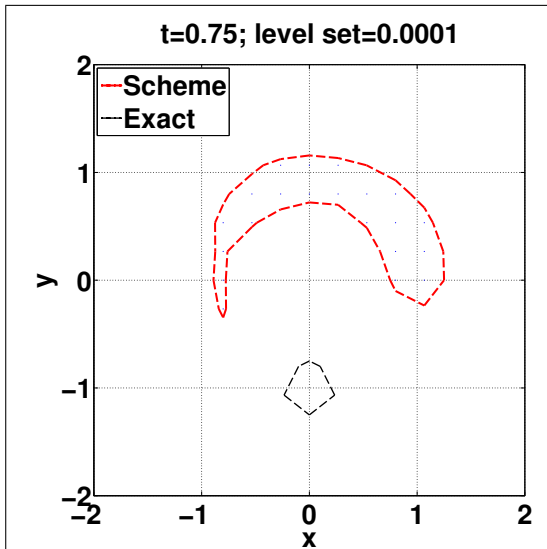


Abb. 5.16:
Vereinigung der erreichbaren Mengen bis zum Zeitpunkt T mit $NN=15$ und adaptiver zeitlicher Diskretisierung $DT=0$.

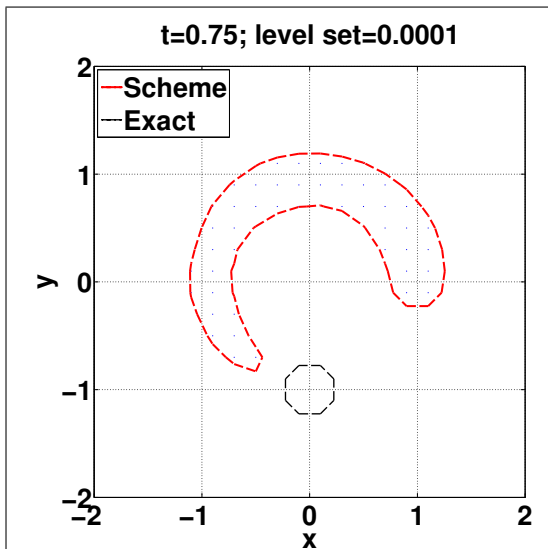


Abb. 5.17:
Vereinigung der erreichbaren Mengen bis zum Zeitpunkt T mit $NN=20$ und adaptiver zeitlicher Diskretisierung $DT=0$.

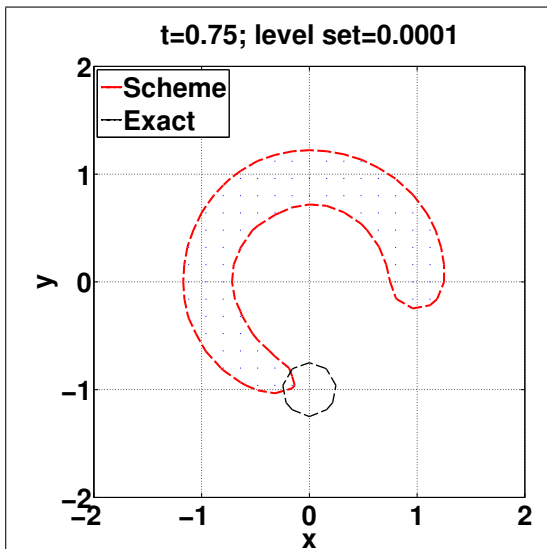


Abb. 5.18:
Vereinigung der erreichbaren Mengen bis zum Zeitpunkt T mit $NN=25$ und adaptiver zeitlicher Diskretisierung $DT=0$.

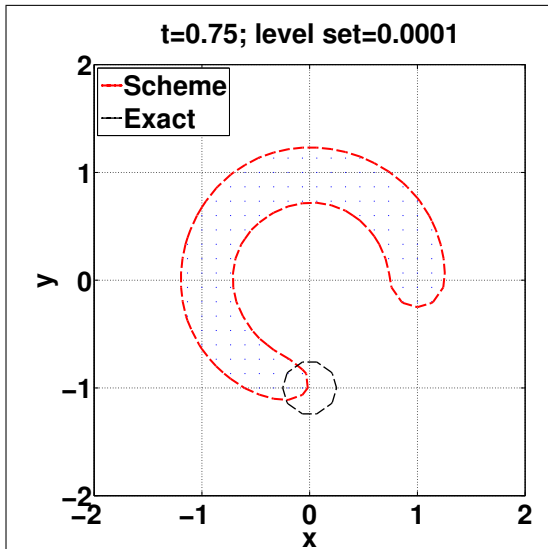


Abb. 5.19:
Vereinigung der erreichbaren Mengen bis zum Zeitpunkt T mit $NN=30$ und adaptiver zeitlicher Diskretisierung $DT=0$.

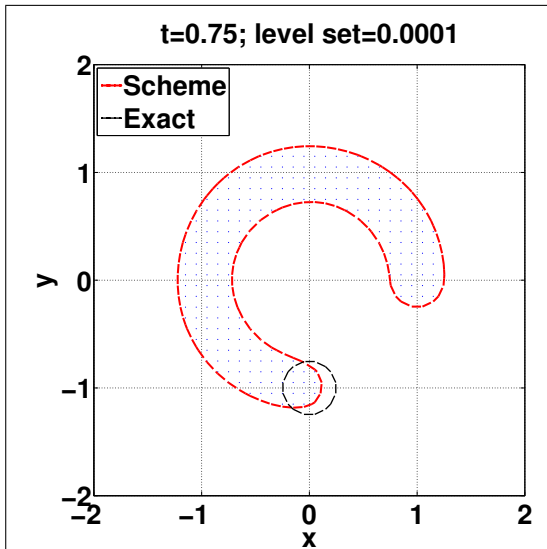


Abb. 5.20:
Vereinigung der erreichbaren Mengen bis zum Zeitpunkt T mit $NN=40$ und adaptiver zeitlicher Diskretisierung $DT=0$.

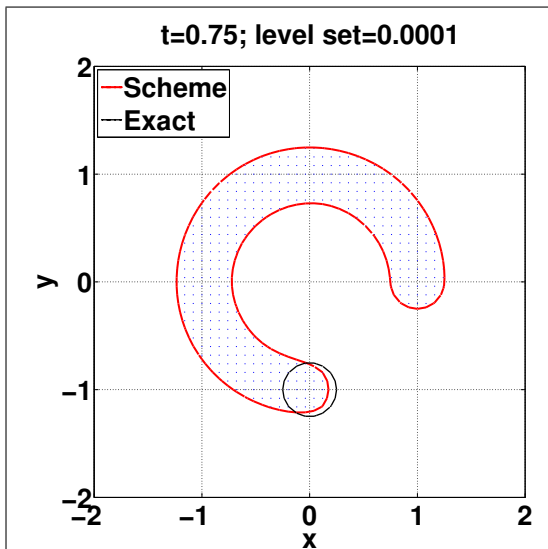


Abb. 5.21:
Vereinigung der erreichbaren Mengen bis zum Zeitpunkt T mit $NN=50$ und adaptiver zeitlicher Diskretisierung $DT=0$.

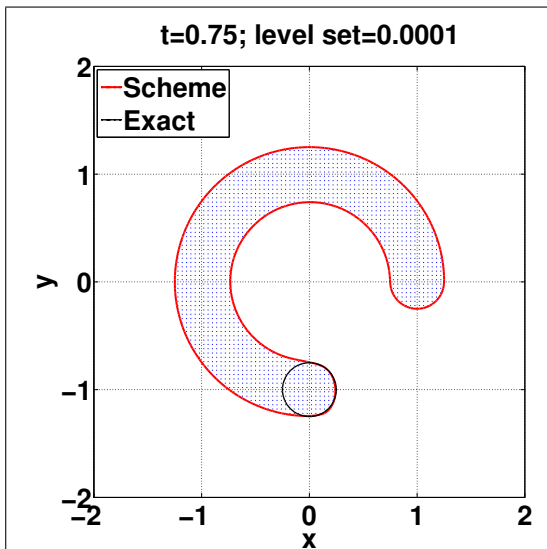


Abb. 5.22:
Vereinigung der erreichbaren Mengen bis zum Zeitpunkt T mit $NN=100$ und adaptiver zeitlicher Diskretisierung $DT=0$.

Betrachtet man hingegen nicht die Vereinigung der Erreichbarkeitsmengen (die virtuelle Kontrolle a wird weggelassen), fällt bei Betrachtung der 2-dimensionalen Erreichbarkeitsmengen auf, dass hier bei geringer Auflösung (grobe Diskretisierung) eine im Bezug zur Referenzmenge (siehe Abb. 5.23 - 5.26) bereits sehr ähnliche Menge berechnet wird.

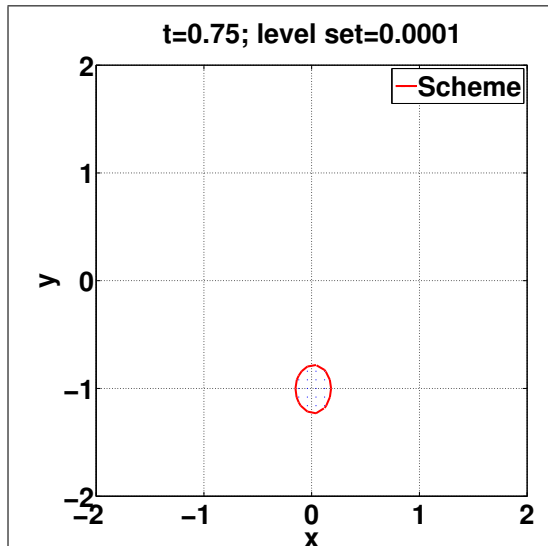


Abb. 5.23:
Erreichbare Menge zum Zeitpunkt T mit $NN=50$ und adaptiver zeitlicher Diskretisierung $DT=0$.

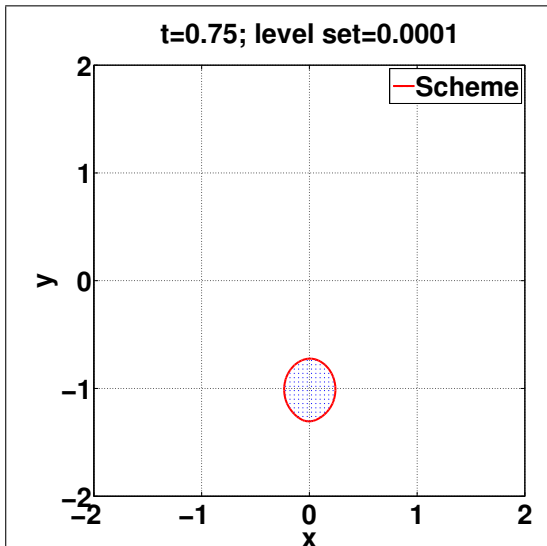


Abb. 5.24:
Erreichbare Menge zum Zeitpunkt T mit $NN=100$ und adaptiver zeitlicher Diskretisierung $DT=0$.

Eine Steigerung der Gitterpunktdichte bläht das Level-Set weiter auf (siehe Abb. 5.28), bis schließlich die Fläche der Referenzmenge erreicht wird (siehe Abb. 5.12 auf Seite 54).

Die Entwicklung der Erreichbarkeitsmenge lässt sich auch in den 3-dimensionalen Plots der Wertefunktion erkennen. In Abbildung 5.29 bis 5.32 wird zwar die Wertefunktion berechnet und ausgewertet, jedoch deckt das Level-Set zum Wert 0 noch nicht die komplette erreichbare Menge ab. Dies wird von den Grenzbedingungen, definiert durch den Parameter v_border , beeinflusst. In Abschnitt 6.4 wird erläutert, wie dieser und weitere Parameter in der numerischen Simulation gewählt werden. Mit zunehmender Dichte der Gitterpunkte (siehe Abb. 5.29 - 5.32) lässt sich eine verstärkte Ausprägung des Level-Sets erkennen. Damit verbunden ist eine Entwicklung in Richtung der xy -Ebene und damit in Richtung des Level-Sets der Referenzmenge (siehe Abb. 5.12 auf Seite 54).

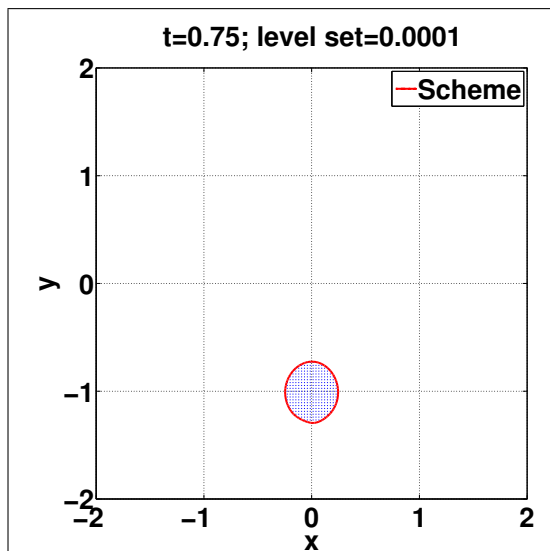


Abb. 5.25:
Erreichbare Menge zum Zeitpunkt T mit $NN=150$ und adaptiver zeitlicher Diskretisierung $DT=0$.

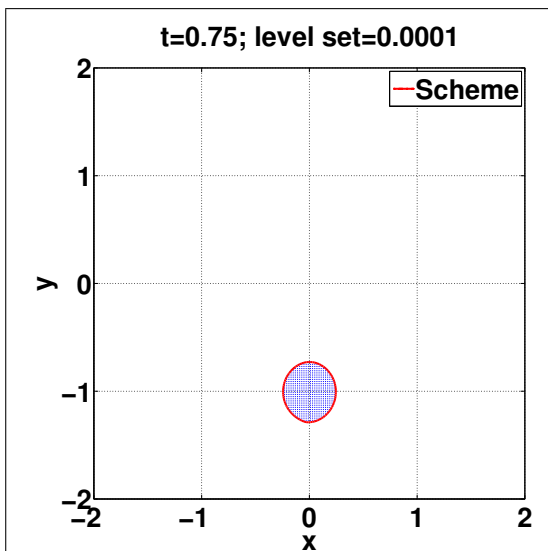


Abb. 5.26:
Erreichbare Menge zum Zeitpunkt T mit $NN=200$ und adaptiver zeitlicher Diskretisierung $DT=0$.

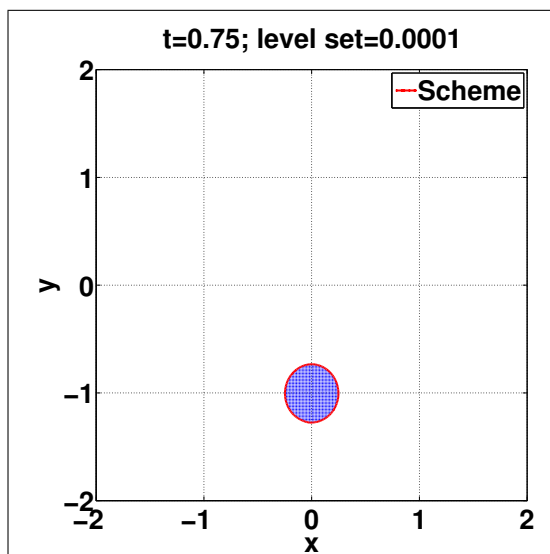


Abb. 5.27:
Erreichbare Menge zum Zeitpunkt T mit $NN=300$ und adaptiver zeitlicher Diskretisierung $DT=0$.

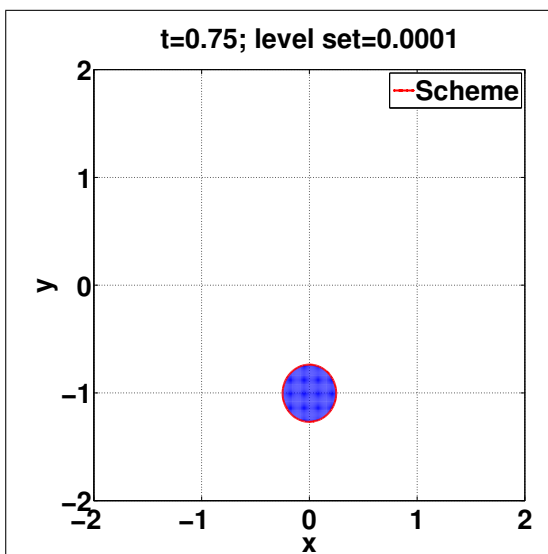


Abb. 5.28:
Erreichbare Menge zum Zeitpunkt T mit $NN=400$ und adaptiver zeitlicher Diskretisierung $DT=0$.

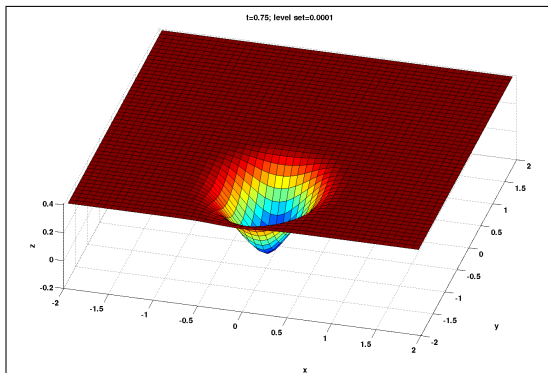


Abb. 5.29:
3-dimensionaler Plot der Wertefunktion zum Zeitpunkt T mit $NN=50$ und adaptiver zeitlicher Diskretisierung $DT=0$.

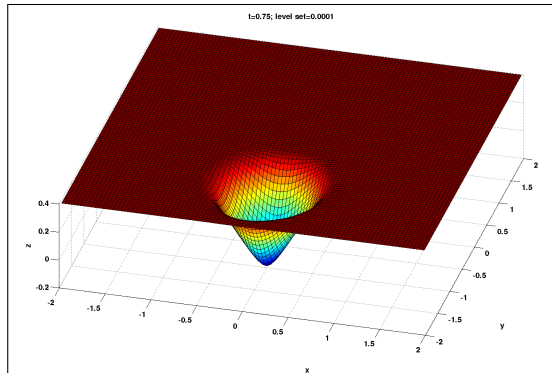


Abb. 5.30:
3-dimensionaler Plot der Wertefunktion zum Zeitpunkt T mit $NN=100$ und adaptiver zeitlicher Diskretisierung $DT=0$.

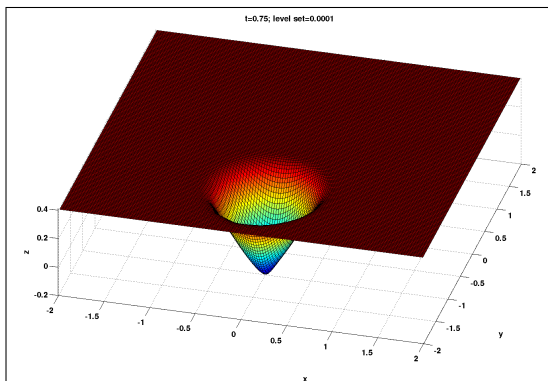


Abb. 5.31:
3-dimensionaler Plot der Wertefunktion zum Zeitpunkt T mit $NN=150$ und adaptiver zeitlicher Diskretisierung $DT=0$.

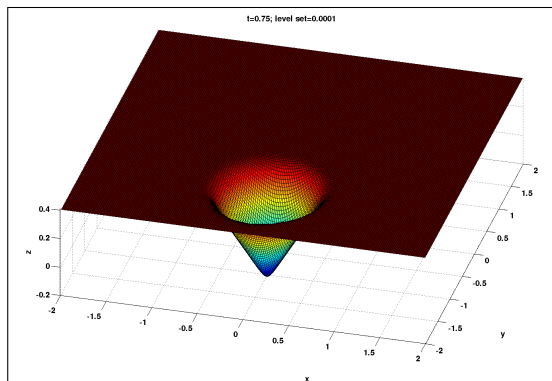
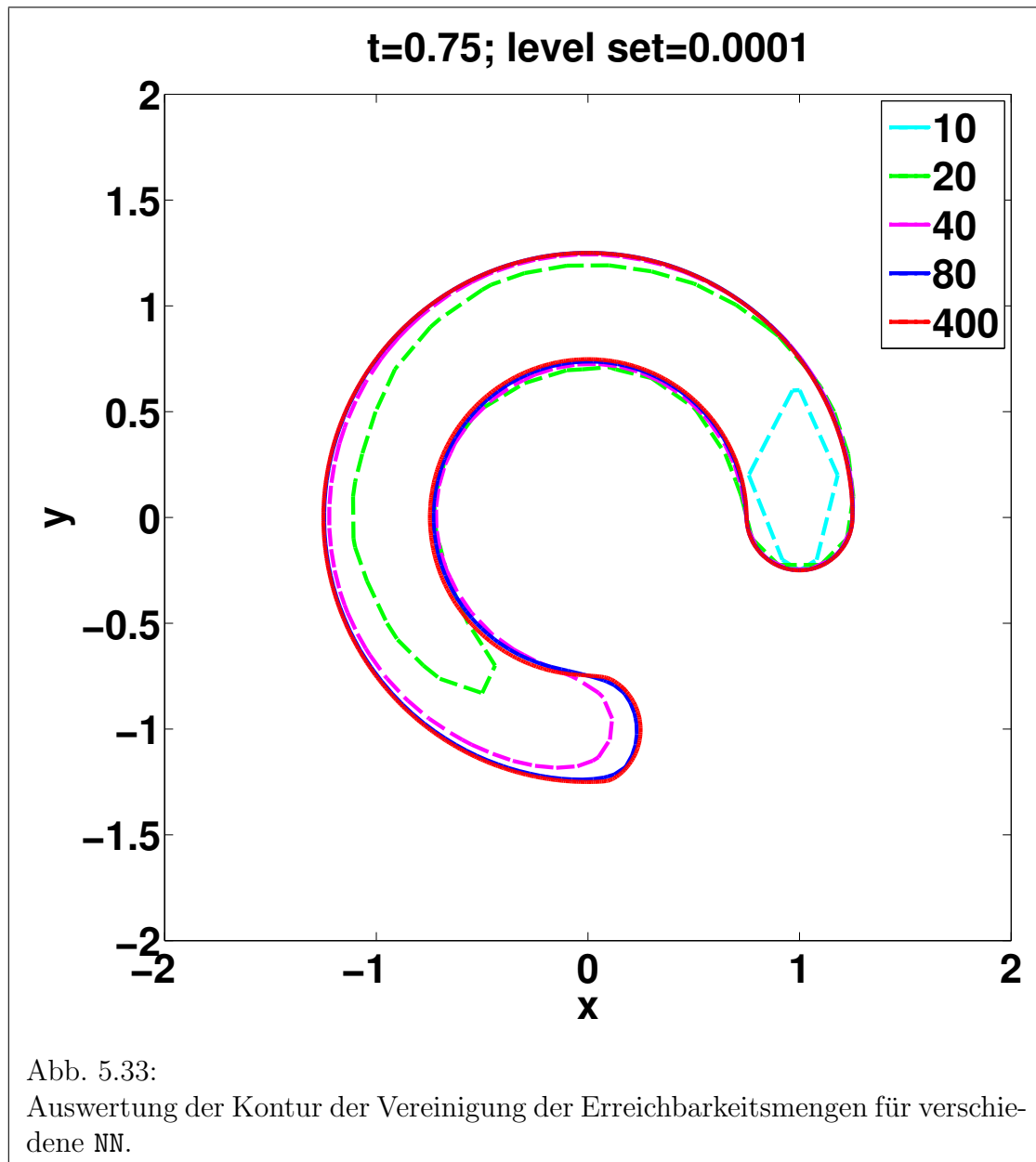


Abb. 5.32:
3-dimensionaler Plot der Wertefunktion zum Zeitpunkt T mit $NN=200$ und adaptiver zeitlicher Diskretisierung $DT=0$.

Betrachten wir nun zur optischen Abschätzung der Konvergenzordnung des verwendeten ENO2-Verfahrens Approximationen mit verschiedenen räumlichen Diskretisierungen (siehe Abb. 5.33). Nach [BFZ10, S. 4309] ist bekannt, dass für dieses häufig verwendete Verfahren kein Konvergenzbeweis existiert. Dennoch ist seine Anwendbarkeit für viele Problemstellungen empirisch nachgewiesen⁷.

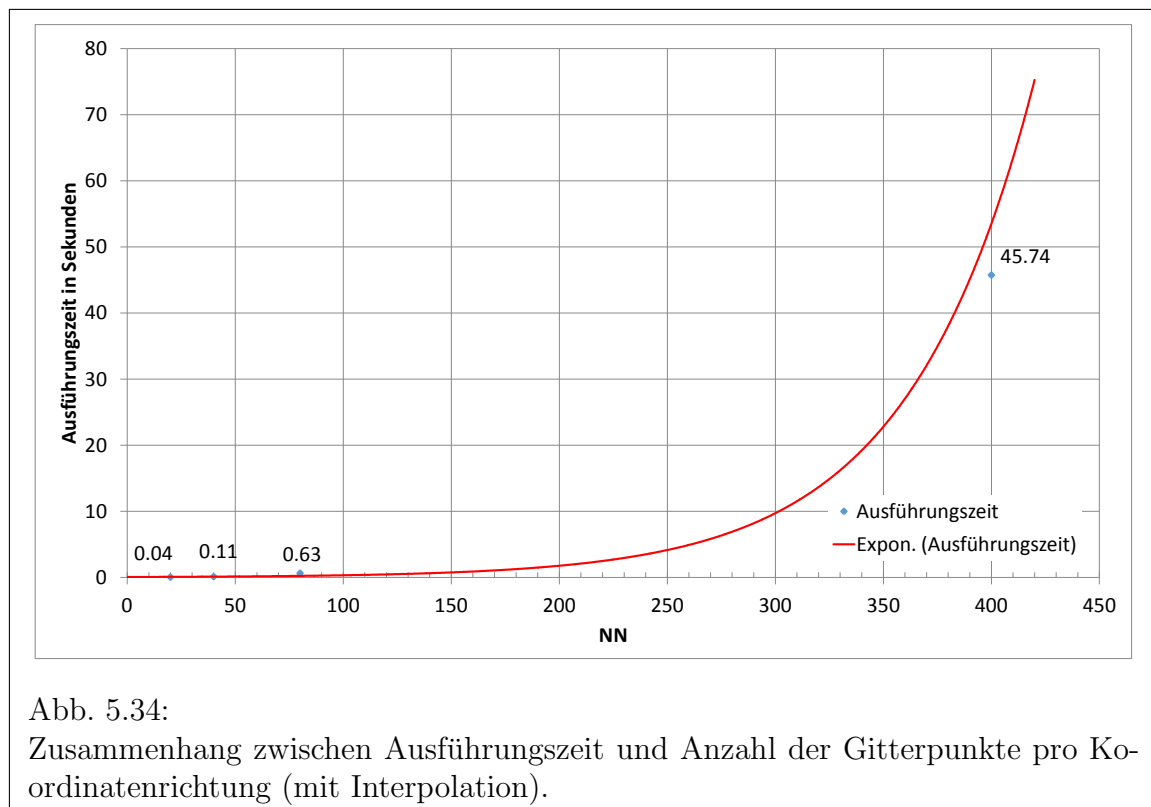


⁷siehe [BFZ10, S. 4309ff., Example 2]

Wertet man den Abstand zwischen der Approximation und der Referenzmenge ($NN = 400$) optisch aus, lässt sich in etwa folgende Beziehung darstellen:

NN	δ_{NN} (in cm)	Ausführungszeit (in s)
20	2,4	0,037066
40	0,5	0,107704
80	0,1	0,629072
400	-	45,7417

Eine Verdoppelung der Gitterpunkte pro Koordinatenrichtung bewirkt in etwa, dass sich der Fehler - gemessen am Abstand zur Referenzmenge mit $NN = 400$ - um 80% reduziert. Die Approximation mit Anzahl der Gitterpunkte $NN = 10$ ist bei dieser Betrachtung aufgrund fehlender Aussagekraft ignoriert worden. Zusätzlich zu dem Abstand zur Referenzmenge ist in der Tabelle auch noch die Ausführungszeit der einzelnen Approximationen gezeigt. Hierbei handelt es sich um die reine Ausführungszeit der Simulation ohne zusätzliche Funktionsaufrufe oder Initialisierungen. Auf diese Weise lässt sich ein annähernd exponentieller Zusammenhang erkennen (siehe Abb. 5.34).



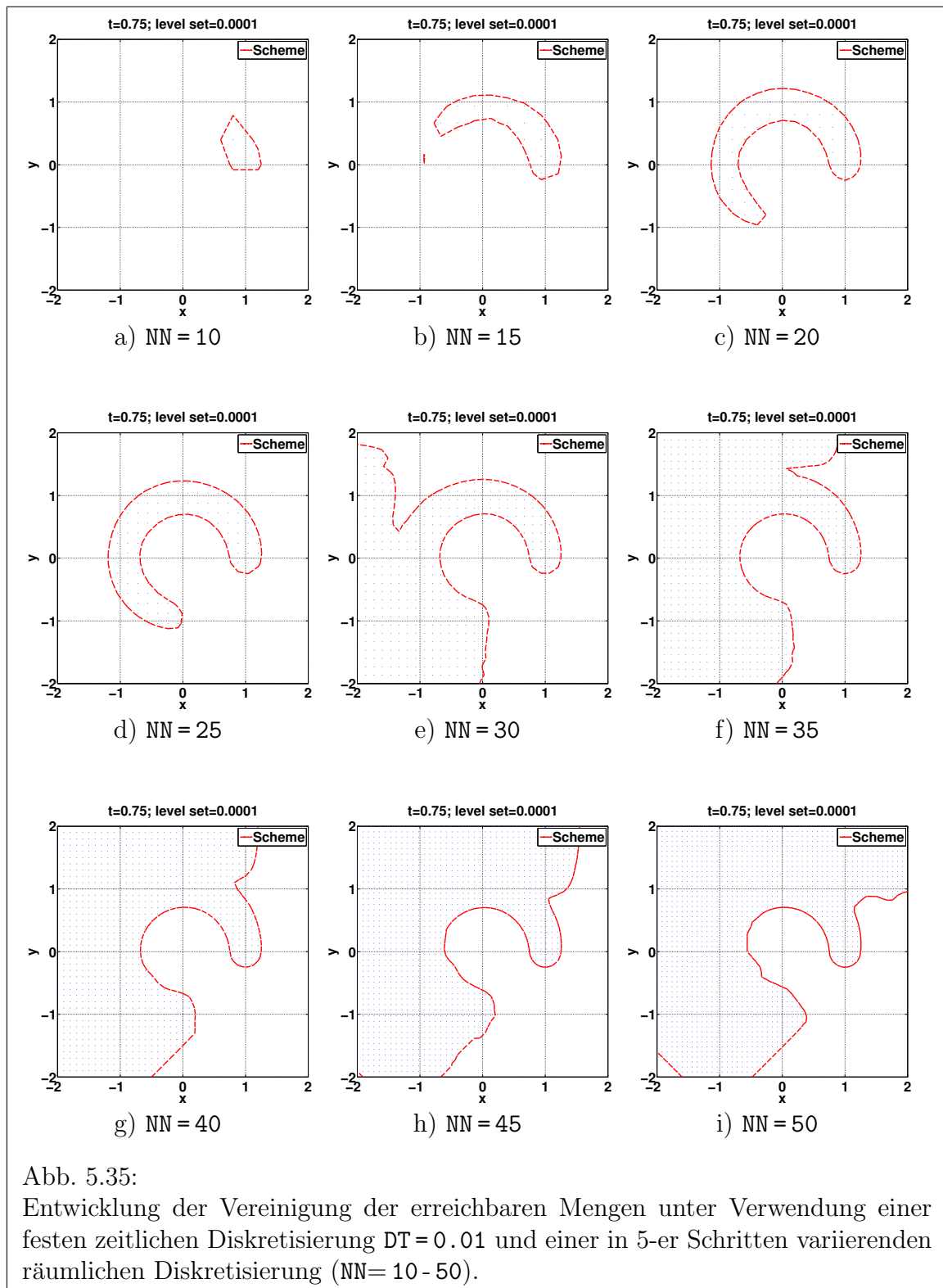
5.4 Beispiel 3

Im Folgenden werden wir nun dieses vorgestellte advektionsähnliche Beispiel dazu verwenden, ein besonderes numerisches Problem aufzuzeigen, welches beim Experimentieren mit verschiedenen Parametern auftreten kann. Wie schon zuvor angesprochen, haben wir in den ersten beiden Beispielen die zeitliche Diskretisierung dem Programm selbst überlassen. Der Algorithmus wählt hierzu die zeitliche Diskretisierung DT adaptiv in Abhängigkeit der räumlichen Diskretisierung, um die CFL-Bedingung zu erfüllen (siehe Gleichung (5.3) auf Seite 44). In den folgenden Graphen wurde bewusst auf die automatische Wahl verzichtet, um die numerischen Auswirkungen einer Verletzung der CFL-Bedingung zu illustrieren. Desweiteren wurden folgende Parameter gewählt (siehe Abb. 5.4)

Tabelle 5.4: Feste Parameter für Beispiel 3

Variable	Wert
METHOD	MFD
TYPE_SCHEME	ENO2
TYPE_RK	RK1
T	0.75
DT	0.01
CFL	0.8
XMIN [DIM]	{ -2. , -2. }
XMAX [DIM]	{ 2. , 2. }
MESH	1
cDIM	1
NCD [cDIM]	{ 2 }
UMIN [cDIM]	{ 0. }
UMAX [cDIM]	{ 1. }
BOUNDARY	1
VBORD	0.4

Anhand der Plotreihe in Abb. 5.35 lässt sich eine numerische Instabilität, optisch verifizierbar durch die „Explosion“ der Vereinigung der erreichbaren Mengen bei $NN = 30$, erkennen. Diese stammt aus der Verletzung der CFL-Bedingung. Die CFL-



Bedingung kann für dieses 2-dimensionale Beispiel folgendermaßen gewählt werden:

$$\underbrace{\frac{\Delta t}{\Delta x} \left(\max_x \max_a (|f_1(x, a)| + |f_2(x, a)|) \right)}_{:=LHS(CFL)} \leq C \lesssim 1, \quad (5.9)$$

wobei f_i der komponentenweise Zerlegung der Dynamik f entspricht und die CFL-Konstante C (siehe Tabelle 5.4) auf den Wert 0.8 gesetzt wurde.

Diese Bedingung wird für $NN=26$ erstmalig verletzt, wie Tabelle 5.5 zu entnehmen ist.

Tabelle 5.5: Verletzung der CFL-Bedingung

NN	Δx	Δt	$\frac{\Delta t}{\Delta x}$	LHS(CFL) aus (5.9)
5	0,80000	0,01	0,0125	0,157
10	0,40000	0,01	0,0250	0,314
15	0,26667	0,01	0,0375	0,471
20	0,20000	0,01	0,0500	0,628
25	0,16000	0,01	0,0625	0,785
26	0,15385	0,01	0,0650	0,817
27	0,14815	0,01	0,0675	0,848
28	0,14286	0,01	0,0700	0,880
29	0,13793	0,01	0,0725	0,911
30	0,13333	0,01	0,0750	0,942
35	0,11429	0,01	0,0875	1,100
40	0,10000	0,01	0,1000	1,257
45	0,08889	0,01	0,1125	1,414
50	0,08000	0,01	0,1250	1,571

Zur Verdeutlichung der Entwicklung der numerischen Explosion zeigt nachfolgende Plotreihe (siehe Abb. 5.36) die räumliche Diskretierungsabfolge in kleineren Schritten: Hier ist klar erkennbar, dass die numerische Explosion direkt bei der Wahl des Diskretisierungsparameters $NN=26$ zum ersten Mal auftritt, was die Berechnungen aus Tabelle 5.5 bestätigt.

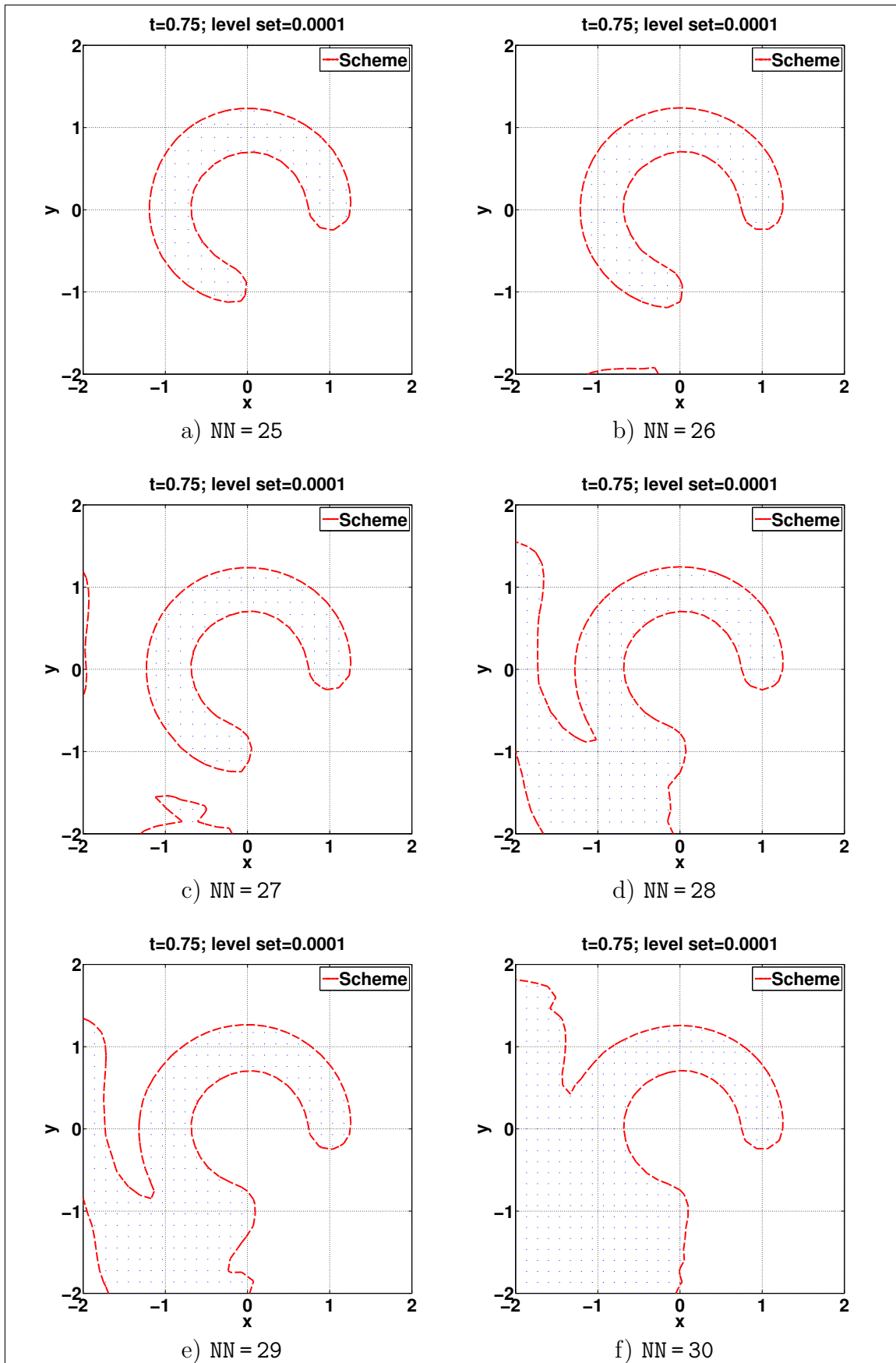


Abb. 5.36:
 Detailansicht der Entwicklung der numerischen Explosion in 1-er Schritten; Anzahl der Gitterpunkte pro Koordinatenrichtung: NN : 25 - 30.

6 Verwendete Software: ROC-HJ-Löser

6.1 Überblick

Im Anschluss soll nun die Software „ROC-HJ“ des ITN-SADCO-Arbeitskreises um Olivier Bokanowski, Anna Désilles und Hasnaa Zidani vorgestellt werden. Dabei wird sowohl auf die mitgelieferte Dokumentation [BDZ13a], das Handbuch [BDZ13b], als auch auf den Quellcode selbst zurückgegriffen. Laut Dokumentation beinhaltet die Software eine C++-Bibliothek, in welcher eine Vielzahl an numerischen Methoden implementiert sind, um Hamilton-Jacobi-Gleichungen zu lösen. Sie zeichnet sich dabei dadurch aus, dass sie auch für Anwender mit nur grundlegenden Programmierkenntnissen geeignet ist und dass sie die Bearbeitung einer weiten Klasse an deterministischen Kontrollproblemen ermöglicht. Darunter fallen unter anderem die in dieser Arbeit betrachteten Erreichbarkeitsprobleme, aber auch weiterführende Problemstellungen [BDZ13a, S. 1] wie Pfad-Planung, Kollisionsvermeidung, Kontrollprobleme auf unendlichem Zeithorizont, Minimal-Zeit-Probleme, Mayer- oder Bolza-Probleme (für eine sehr kurze Problembeschreibung siehe Seite 18). Zudem können auch noch zustandsbeschränkte Kontrollprobleme, Differentialspiele und auch Endzeitprobleme analysiert und gelöst werden. Auch bietet die Software direkt Möglichkeiten, um die Ergebnisse der numerischen Analysen mit Hilfe von Matlab zu visualisieren. Dazu ist eine Matlab-Funktionsdatei enthalten, welche die Ausgabe-Dateien in Graphiken umwandelt. Auch diese beinhaltet, ähnlich wie die Daten-Files, welche zur Problembeschreibung und Übergabe genutzt werden, einige Konfigurations- und Manipulationsmöglichkeiten (siehe [BDZ13b, S. 8]). Zuerst werden wir die Problemformulierung betrachten und anschließend die Implementierungsmöglichkeit eines neuen Problems betrachten. Für diese Betrachtung orientieren wir uns stark an der zur Verfügung stehenden Version des Handbuchs [BDZ13b].

6.2 Problemformulierung

Das Ziel der Software ist es, eine Approximation der Lösung der Hamilton-Jacobi-Gleichung auf endlichem Zeithorizont zu finden. Allgemein werden also HJ-PDEs folgender Gestalt betrachtet:

$$\begin{cases} \frac{\partial u}{\partial t} + H(t, x, \nabla u), & x \in \Omega, \quad t \in [0, T] \\ u(0, x) = u_0(x), & x \in \Omega \\ + \text{Grenzbedingungen auf } \partial\Omega \times (0, T), \end{cases}$$

wobei Ω ein endliches Gebiet $\prod_{i=1}^d [a_i, b_i]$ des \mathbb{R}^d und $u_0(x) : \mathbb{R}^d \rightarrow \mathbb{R}$ eine gegebene Funktion ist. Die bereits implementierten Grenzbedingungen beinhalten

$$u(t, x) = g_{border}(t, x) \quad (\text{Dirichlet Bedingung}),$$

$$\begin{aligned} u(t, x + \Pi) &= u(t, x) && \text{(periodische Bedingung) und} \\ \partial_{xx}u(t, x) &= 0 && \text{(lineare Extrapolation),} \end{aligned}$$

wobei in den analysierten Beispielen meistens eine konstante Randbedingung $u(t, x) = u_{border}$, $x \in \partial\Omega$, $t \in [0, T]$ Anwendung findet. Das hier betrachtete Hamiltonfunktional kann unterschiedlich definiert werden:

- als analytischer Ausdruck, falls ein solcher bekannt ist (z.B. $H(t, x, p) := c(t, x) |p| + f(t, x) \cdot p$), oder
- für unsere betrachtete Problemstellung zur Berechnung von Erreichbarkeitsmengen, abhängig von der Anzahl der möglichen Kontrollen, entweder als Ein-Spieler-Kontrollproblem

$$H(t, x, p) := \max_{a \in \mathcal{A}} \left(-f(t, x, a) \cdot p - l(t, x, a) \right),$$

oder

$$H(t, x, p) := \min_{a \in \mathcal{A}} \left(-f(t, x, a) \cdot p - l(t, x, a) \right),$$

mit einer Kontrollmenge $\mathcal{A} = \prod_{i=1}^{m_{\mathcal{A}}} [\alpha_i, \beta_i]$, ($m_{\mathcal{A}} \geq 1$), bzw. als Zwei-Spieler-Kontrollproblem

$$H(t, x, p) := \max_{a \in \mathcal{A}} \min_{b \in \mathcal{B}} \left(-f(t, x, a, b) \cdot p - l(t, x, a, b) \right),$$

oder

$$H(t, x, p) := \min_{a \in \mathcal{A}} \max_{b \in \mathcal{B}} \left(-f(t, x, a, b) \cdot p - l(t, x, a, b) \right),$$

mit den Kontrollmengen $\mathcal{A} = \prod_{i=1}^{n_{\mathcal{A}}} [\alpha_i^{\mathcal{A}}, \beta_i^{\mathcal{A}}]$ und $\mathcal{B} = \prod_{i=1}^{n_{\mathcal{B}}} [\alpha_i^{\mathcal{B}}, \beta_i^{\mathcal{B}}]$, f dem Vektorfeld des zugrundeliegenden optimalen Steuerungsproblems und der Kostenfunktion l . Da wir in dieser Arbeit nur Probleme der Form $\dot{y}(t) = f(t, y(t), a(t))$ betrachten und die Software zur Berechnung von Erreichbarkeitsmengen einsetzen, genügt die Betrachtung mit $l(t, x, a) = 0$ bzw. $l(t, x, a, b) = 0$, was im Ein-Spieler-Problem zu

$$H(t, x, p) := \max_{a \in \mathcal{A}} \left(-f(t, x, a) \cdot p \right),$$

bzw.

$$H(t, x, p) := \min_{a \in \mathcal{A}} \left(-f(t, x, a) \cdot p \right),$$

führt. Die Formulierung für Zwei-Spieler-Probleme ist analog. Da in der numerischen Analyse in dieser Arbeit aber nur Ein-Spieler-Probleme betrachtet werden, wird die exakte Formulierung an dieser Stelle vernachlässigt.

Gehen wir nun im nächsten Punkt direkt auf die Anwendung des ROC-HJ-Lösers ein.

6.3 Kompilation und Ausführung

In dieser Arbeit wird der ROC-HJ-Solver der ITN-SADCO-Arbeitsgruppe um Oliver Bokanowski verwendet. Die Analysen und Plots sind mit Version 1 (veröffentlicht im Frühjahr 2013) des ROCHJ-Solvers erstellt worden und die Anwendung selbst wurde auf einer virtuellen Maschine mit der Linux-Distribution Ubuntu 12.04 auf einem 32bit-System kompiliert und ausgeführt. Eine verbesserte Beta-Version 2.0 wird in naher Zukunft verfügbar sein.

Die Software selbst ist als gepacktes Archiv von <http://uma.ensta-paristech.fr/var/files/ROC-HJ/> heruntergeladen. Betrachten wir nun die ersten Schritte der Bereitstellung und der Ausführung des Programms. Um aus dem Inhalt des ZIP-Archivs „roc-hj-version1-linux32.zip“ des ROC-HJ-Solvers ein funktionierendes Programm zu erhalten, werden der C++-Compiler GCC⁸ und das Programmierwerkzeug CMake⁹ empfohlen, da mit Hilfe von CMake auf eine mitgelieferte Skript-Datei zurückgegriffen werden kann und direkt systemabhängige Make-Dateien erzeugt werden können.

Folgende Schritte sind nötig, um die Software zu kompilieren¹⁰ und zu starten:

1. Erstellung des Make-Files: `cmake .` (Bitte nicht den „Punkt“ vergessen!)
2. Kompilation: `make`

Desweiteren existieren noch zwei weitere Befehle zum Aufräumen und Löschen unbenötigter Dateien:

- `make clean`, bzw. `make cleanall`
- `./clean`

Die Ausführung des Programmes erfolgt direkt über das Terminal mit dem Befehl:

```
./exe
```

Dieser einfache Befehl verwendet die direkt im Programmcode festgelegten Parameter. Für die Ausführung mit verschiedenen Parametern existiert ein erweiterter Befehlscode:

```
./exe -nn NN -nc NC
```

Die beiden Einstellungen sind jeweils optional und beschreiben folgende Parameter:

- Option `-nn NN` : Anzahl der Gitterpunkte pro Koordinatenrichtung
- Option `-nc NC` : Anzahl der Kontrollen pro Koordinatenrichtung

⁸für weitere Informationen siehe <http://gcc.gnu.org/install/> .

⁹für weitere Informationen siehe <http://www.cmake.org/cmake/resources/software.html> .

¹⁰Die genannten Befehle sind in der Konsole im Hauptverzeichnis „roc-hj-version1-linux32“ auszuführen.

Zusätzlich zu den Parametern, welche das numerische Verfahren beeinflussen, ist noch ein Übergabeparameter zur Parallelisierung möglich. Mit

```
./exe -nt nbth
```

legt man die Anzahl der Prozesse fest. Zur Parallelisierung verwendet die Software OpenMP¹¹.

Beispielausführung

Um eines der auf der CD befindlichen Beispiele auszuführen, sind also folgende Schritte notwendig:

0. Erstellen des `make`-Files: `cmake .`
1. Anpassen der Programmparameter im Beispiel-Headerfile `data.my_model.h` (siehe Kapitel 6.4)
2. Auswahl der richtigen Beispiel-Header-Datei in der Datei `data.simulation.h`
3. Kompilation des Programmes: `make`
4. Ausführung des Programmes: `./exe -nn NN -nc NC -nt NT` (siehe oben).
Dieser Befehl kann beispielsweise noch um eine linux-spezifische Ausgabeumleitung erweitert werden, um die Ausgabe im Terminal direkt in einer vorher erstellten Datei zu speichern: `./exe -nn NN -nc NC -nt NT | tee display.txt`
Dies erleichtert die Auswertung von vielen Simulationen, ausgeführt mit verschiedensten Übergabeparametern.

Der Schritt 0. muss prinzipiell nur einmal ausgeführt werden, um die `make`-Datei zu erstellen, die Schritte 1. - 4. aber bei jeder Änderung im Quellcode. Man beachte, dass zur graphischen Visualisierung anschließend auch noch `output.view` ausgeführt werden muss oder alternativ eigene Programme zur Auswertung der ausgegebenen Daten angewandt werden können.

6.4 Vorstellung des Programms

Zusätzlich zu den einfachen Übergabeparametern zur Steuerung des numerischen Verfahrens bietet die Software auf Quellcode-Ebene noch viel mehr Variations- und Manipulationsmöglichkeiten. Im Folgenden werden diese in Bezug auf die in dieser Arbeit untersuchten Problemstellungen und auf bestimmte numerische Verfahren näher erklärt.

Zunächst beschäftigen wir uns mit der Programmstruktur und zeigen, auf welche Weise der Benutzer Probleme an das Programm übergeben und Einstellungen vornehmen kann. Die Programmstruktur ist in der schematischen Abbildung 6.1 gezeigt, welche aus der Dokumentation [BDZ13a, S. 4 Figure 1] entnommen ist.

¹¹siehe <http://openmp.org/>

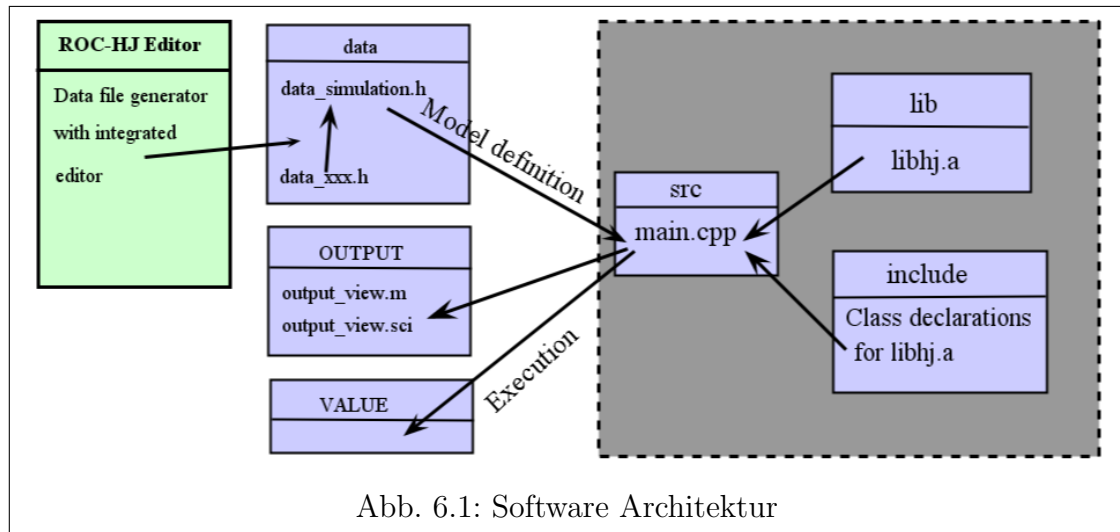


Abb. 6.1: Software Architektur

Die Hauptprogrammteile befinden sich in den Ordnern `src`, `lib` und `include`.

- Der Programmkern, die vorkompilierte Library `libhj.a` mit den implementierten numerischen Methoden (Klassen `HJB`, `HJB_FD` und `HJB_SL`) befindet sich im Ordner `lib`. In der `libhj.a` sind ebenfalls die Methoden zur Datenverwaltung (Gitter, Optionen, etc.) enthalten. Eine Analyse des Quellcodes und somit der Implementierung der Methoden ist nicht möglich, da die Bibliothek nicht als direktes Source-File vorliegt. Dies ist aber ohnehin nicht Ziel dieser Arbeit.
- Die benötigten Header-Dateien, welche die Klassen-Deklarationen beinhalten, liegen im Ordner `include`.
- Im Ordner `src` ist schließlich das Hauptprogramm `main.cpp` zu finden, welches Befehle zur Dateneinlesung, zum Anlegen der verwendeten Strukturen und Konstrukte und zum Aufruf der eigentlichen Berechnung beinhaltet. Betrachten wir den Quellcode und Aufbau des `main`-Programmes im Hinblick auf die Problemstellung (nichtrelevante Teile des Quellcodes wie Ausgaben und auch weitgehend enthaltene ausführliche Kommentare werden in der Betrachtung ignoriert):

main.cpp

```

1 #include "stdafx.h"
2 #include "HJB_FD.h"
3 #include "HJB_SL.h"
4
5 // - HERE PUT THE DESIRED "data_xx.h" FILE
6 // #include "../data/data_simulation_other.h"
7
8
9

```

```

10 #include "../data/data_simulation.h"
11
12 ///////////////////////////////////////////////////////////////////
13 ///////////////////////////////////////////////////////////////////  MAIN  ///////////////////////////////////////////////////////////////////
14 ///////////////////////////////////////////////////////////////////
15
16 int main(int argc, char *argv[], char *env[]){
17 ...

```

Zu Beginn der `main`-Datei werden in den Zeilen 2 und 3 die Header-Dateien für die implementierten numerischen Methoden `HJB_FD` (Finite Differenzen) und `HJB_SL` (Semi Lagrangian) eingebunden. In Zeile 10 wird die Datei `data_simulation.h` aufgenommen, welche die Daten für die Problemformulierung enthält (siehe dazu im Folgenden die Beschreibung zum Eingabeordner `data`). Anschließend ruft man in Zeile 10 die `main`-Funktion auf.

In Zeile 33 folgt die Initialisierung der Daten und das Auslesen der Eingabe-Dateien mit der Routine `init_data`.

main.cpp

```

32 ...
33     init_data();
34 ...

```

Nach einigen Kommentaren und Ausgaben wird der eigentliche Programmcode mit dem Setzen der Dimensionsparameter und dem Einlesen der Übergabeparameter fortgesetzt:

main.cpp

```

55 ...
56     int i, numthreads=1, griddim=2;
57     int taille_maillage[DIM];
58     for(i=0;i<DIM;i++)
59         taille_maillage[i]=ND[i];
60     int NCDuser[cDIM], NCuser=1;
61     for(i=0;i<cDIM;i++){
62         NCDuser[i]=NCD[i];
63         NCuser*=NCDuser[i];
64     }
65
66     if(argc>=3){
67         for(i=1 ; i<argc ; i+=2){
68             if(strcmp(argv[i], "-nn") == 0){
69                 int nn = atoi(argv[i+1]);
70                 for(int i=0;i<DIM;i++)
71                     taille_maillage[i]=nn;
72             }
73             else if(strcmp(argv[i], "-nc") == 0){

```

```

74         int nc = atoi(argv[i+1]);
75         NCuser = 1;
76         for(int i=0;i<cDIM;i++){
77             NCDuser[i]=nc;
78             NCuser*=nc;
79         }
80     }
81     else if(strcmp(argv[i], "-nt") == 0){
82         numthreads = atoi(argv[i+1]);
83     }
84     else if(strcmp(argv[i], "-nd") == 0){
85         griddim = atoi(argv[i+1]);
86     }
87 }
88 }
89 ...

```

In diesem Block (Zeile 56 - 88) werden die optionalen Übergabeparameter eingelesen und in Variablen geschrieben.

Zeile 56 - 64: Problemdimensionen und die Anzahl der Kontrolldimensionen,

Zeile 68 - 72: Anzahl der Gitterpunkte pro Koordinatenrichtung,

Zeile 73 - 80: Anzahl der diskretisierten Kontrollen pro Kontrolldimension,

Zeile 81 - 83: Anzahl der OpenMP-Prozesse zur Parallelisierung,

Zeile 84 - 86: ein zusätzliches Setzen der Gitterdimension per Übergabeparameter.

main.cpp

```

100 ...
101     //- SEQVERSION
102     unsigned mpidatas[griddim];
103
104     static struct mrp maillage_regulier_params =
105     { DIM, PRECOMPUTE_COORDS, MESH, ONEVAL, (unsigned*) ↵
106       taille_millage,
107       (double*) XMIN, (double*) XMAX, (int*) PERIODIC, ↵
108       BORDERSIZE, OBSTACLE, g_obstacle, OBSTACLE_TILDE, ↵
109       g_obstacle_tilde};
110
111     static struct cp commande_params =
112     { cDIM, (int*) NCDuser, NCuser, (double*) UMIN, (double*) ↵
113       UMAX };
114
115     int NC2=1;
116     for(i=0;i<cDIM2;i++)
117         NC2*=NCD2[i];
118     static struct cp commande_params2 =
119     { cDIM2, (int*) NCD2, NC2, (double*) UMIN2, (double*) ↵
120       UMAX2 };

```

```

117     static struct pp   parallel_params =
118     { griddim, mpidatas, my_id, numprocs, numthreads };
119
120     static struct gp   general_params =
121     { METHOD, NAME, EXTERNALV0,
122       COMPUTE_MAIN_LOOP, OPTIM, COMPUTE_TMIN, COMPUTE_VEX,
123       SAVE_VFALL, SAVE_VFALL_STEP,
124       VALUE_PB, SAVE_FOR_VALUE_PB, SAVE_FOR_VALUE_PB_FINAL,
125       CHECK_ERROR, CHECK_ERROR_STEP, C_THRESHOLD,
126       VBORD, T, DT, MAX_ITERATION, EPSILON,
127       BOUNDARY,
128       *dynamics, *dynamics2, *distributed_cost, ↵
129         *distributed_cost2, *v0, *Vex,
130       *g_border
131     };
132
133     static struct ppp  postprocess_params =
134     { TRAJPT, (double*) initialpoint, (int*) COUPE_DIMS, ↵
135       (double*) COUPE_VALS };
136 ...

```

In diesem Abschnitt von Zeile 102 bis Zeile 133 werden die Strukturen zur Ausführung der numerischen Simulation deklariert.

main.cpp

```

153 ...
154     if(general_params.METHOD == MFD){
155         static struct fdp  FD_params =
156         { TYPE_SCHEME, TYPE_RK, COMMANDS, CFL, *Hnum,
157           *compute_Hconst };
158         HJB_FD * pb = new HJB_FD(maillage_regulier_params, ↵
159           commande_params, commande_params2, parallel_params, ↵
160           general_params, postprocess_params, FD_params);
161         if(COMPUTE_MAIN_LOOP){
162             pb->initFiles();
163             pb->mainloop();
164         }
165         else {
166             if(COMPUTE_TMIN)
167                 pb->loadTmin(tminFile);
168             pb->loadV(VFile);
169         }
170         //printf("main.cpp: TRAJPT:=%i\n",TRAJPT);
171         //printf("computetraj=%i ...\n", computetraj);
172         pb->postprocess();
173         delete pb;
174     }
175 ...

```

Im Block von 154 bis 172 werden für das Finite-Differenzen-Verfahren die Konstruktoren für die oben deklarierten Objekte aufgerufen und schließlich

das Hauptprogramm mit dem Kommando `pb->mainloop()` gestartet. Wir gehen hier davon aus, dass der Parameter `COMPUTE_MAIN_LOOP` auf `TRUE` gesetzt ist, wie es bei den implementierten Beispielen der Fall ist. (Die anderen Parameter werden an dieser Stelle vernachlässigt.) Der weitere Quellcode des `main`-Programms ist nicht relevant, da dort der letzte gezeigte Block, lediglich an ein anderes Verfahren angepasst, wiederholt wird. Dieses andere Verfahren, MSL („Method Semi Lagrangian“), wird in dieser Arbeit nicht betrachtet.

Zur Ein- und Ausgabe sind die Ordner `data`, `OUTPUT` und `VALUE` erstellt.

- Innerhalb des Ordners `data` werden üblicherweise alle relevanten Eingaben des Benutzers vorgenommen. Dazu muss der Benutzer über grundlegendes programmiertechnisches Wissen verfügen. Für jedes Problem, welches berechnet und analysiert werden soll, wird hier vom Benutzer ein Daten-Header-File z.B. `data_yourModel.h` erstellt. In dieser Datei wird das zugrundeliegende Kontrollproblem spezifiziert, das Kostenfunktional und auch die Hamiltonfunktion (falls bekannt) implementiert. Auch finden sich hier die Anfangs- und Grenzbedingungen sowie alle anderen Parameter, welche das zu lösende Problem beschreiben. Zudem müssen hier ebenfalls die Verfahren konfiguriert werden und Einstellungen zur Berechnung bestimmter Ausgabewerte vorgenommen werden, welche bei der graphischen Auswertung genutzt werden können. Um dieses Header-File `data_yourModel.h` für das Programm zu aktivieren, muss es in der Datei `data_simulation.h` mit dem Befehl `#include data_yourModel.h` eingebunden werden. Indem man hier verschiedene Headerfiles verwendet, kann einfach zwischen verschiedenen Problemen gewechselt werden.
- In die Ordner `OUTPUT` und `VALUE` werden nach Ausführung des Programms die Ergebnisse in bestimmte Daten-Files geschrieben. Diese werden zur graphischen Auswertung benötigt, welche mit Hilfe der Matlab-Funktion `output_view.m` im Ordner `OUTPUT` aufgerufen wird.

Eingabeparameter

In diesem Abschnitt sollen nun die Eingabeparameter gezeigt und erläutert werden, welche der Nutzer der Software in den Headerfiles definieren muss, um eine numerische Simulation zu erhalten. Hierbei konzentrieren wir uns auf die Parameter, welche für unsere Problemstellung, also Kontrollprobleme auf endlichem Zeithorizont, relevant sind. Sie stellen also nur einen Ausschnitt der möglichen Parameter dar.¹² Diese Einführung in die Eingabeparameter soll nun anhand des kommentierten Beispiels `basicmodel.h` geschehen. Man beachte, dass die Überschriften dabei aus dem Original übernommen und nicht verändert sind, so dass eine leichtere Zuordnung zum Quellcode geschaffen werden kann. Desweiteren steht zu einigen Parametern ein Wert im Kommentar, welcher als Standard verwendet wird. Dies bedeutet, dass

¹²für eine komplette Übersicht siehe [BDZ13b, S. 3-7]

in den betrachteten Problemen meistens dieser Wert verwendet wurde. Außerdem werden auch Parameter, welche in den verwendeten Beispielen nicht gebraucht wurden, im Quellcode genannt, sind aber nicht in grün, sondern in grau (Originale Kommentare) gehalten. Diese Passagen werden jedoch im Benutzerhandbuch der Software [BDZ13b] genau beschrieben:

data_basicmodel.h

```

1  //-----
2  //-- Rotation example , dimension 2
3  //-----
4  //--   v_t + max_a < - f(x,a), nabla v > = 0;   f(x_1,x_2,a)= a )
      (-x_2,x_1); a in {0,1}
5  //--   v(0,x)=v0(x)
6  //-----
7
8
9  const char    NAME[150]          = "data_basicmodel.h, Feb. )
      2013\nAuthors : O. Bokanowski, A. Desilles, H. Zidani";
10 const int     DIM                = 2;                //-- Dimension 2 )
      [Konvention: 2]
11
12 const int     COMMANDS           = 1;                //-- 0: analytische )
      Hamiltonfunktion ist bekannt und wird weiter unten direkt )
      implementiert ; 1/2: Je nach Anzahl der Spieler wird hier )
      Ein-Spieler-Problem bzw. Zwei-Spieler-Problem festgelegt; )
      [Konvention: 1]
13 const int     OPTIM              = MAXIMUM;         //-- Art der )
      Hamiltonfunktion: Maximierung (MAXIMUM) oder Minimierung (MINIMUM) )
      [Konvention: MAXIMUM]
14
15
16 //-----
17 //-- DF method parameters:
18 //-----
19 const int     METHOD              = MFD;             //-- )
      Verfahrensauswahl: Finite Differenzen (MFD) oder Semi-Langresches )
      Verfahren [Konvention: MFD]
20 const int     TYPE_SCHEME        = ENO2;           //-- Auswahl des )
      räumlichen Diskretisierungsverfahrens: 1 oder LF für )
      Lax-Friedrichs-Verfahren; 2 or ENO2 für ENO2-Verfahren.
21 int TYPE_RK          = RK1;                //-- Auswahl des )
      zeitlichen Diskretisierungsverfahrens (für TYPE_SCHEME = ENO2): 1 )
      oder RK1 für Runge-Kutta-Verfahren 1. Ordnung; 2 oder RK2 für )
      Runge-Kutta-Verfahren 2. Ordnung; 3 oder RK3 für )
      Runge-Kutta-Verfahren 3. Ordnung
22
23 //-----
24 //-- stopping criteria parameters:

```



```

25 //-----
26 const double EPSILON          = 0.0;          //-- Mit diesem )
   epsilon kann der L1-Fehler zwischen zwei aufeinanderfolgenden )
   Iterationsschritten begrenzt werden und als Abbruchkriterium )
   genutzt werden (wird nicht verwendet, daher EPSILON = 0.0)
27 const int     MAX_ITERATION    = 100000;      //-- Anzahl der )
   Iterationen als Abbruchkriterium (da nicht verwendet, auf sehr )
   hohen Wert gesetzt)
28 double       T                 = 0.75;       //-- Endzeit als )
   Abbruchkriterium (wird für die betrachteten Endzeitprobleme )
   verwendet)
29
30
31 //-----
32 //-- discretization parameters:
33 //-----
34 const double  DT                = 0.;         //-- Setzen des )
   Zeitschrittes für die zeitliche Diskretisierung; bei DT = 0. wird )
   ein adaptiver Zeitschritt verwendet, sodass eine CFL-Bedingung )
   erfüllt ist
35 double       CFL                = 0.8;       //-- Schranke für die )
   CFL-Bedingung
36
37 const int     NN=200;           //-- Anzahl der )
   Gitterpunkte pro Koordinatenrichtung (wird vom Übergabeparameter )
   überschrieben)
38 int          ND[DIM]           = { NN , NN}; //-- Matrix mit Größe )
   des Gitters in jeder Koordinatenrichtung (Die Länge des Arrays )
   entspricht der Dimension DIM des Gitters)
39 double       XMIN[DIM]         = { -2. , -2.}; //-- Untere Grenze(n) )
   des betrachteten Gebiets
40 double       XMAX[DIM]         = { 2. , 2.}; //-- Obere Grenze(n) )
   des betrachteten Gebiets
41
42 const int     PERIODIC[DIM]     = { 0 , 0 }; //-- Periodizität des )
   Gitters für jede Koordinatenrichtung (1:periodisch; 0:nicht )
   periodisch) [Konvention: 0]
43 const int     MESH              = 0;         //-- Gitterdefinition: )
   0:xi ist Zentrum der Gitter-Zelle (ND[i] Gitterpunkte in Richtung )
   i) ; 1:xi beinhaltet dn Rand der Gitter-Zelle (ND[i]+1 )
   Gitterpunkte in Richtung i) [Konvention: 0]
44
45 const int     cDIM              = 1;         //-- Dimension der )
   Steuerung (Ein-Spieler-Problem)
46 const int     NCD[cDIM]         = { 2 };    //-- Matrix mit Anzahl )
   der (diskreten) Steuerungen pro Koordinatenrichtung (Die Länge des )
   Arrays entspricht der Dimension cDIM der Steuerung)
47 const double  UMIN[cDIM]        = { 0.};    //-- Untere Grenze(n) )
   der Steuerung in jede Richtung
48 const double  UMAX[cDIM]        = { 1.};    //-- Obere Grenze(n) )
   der Steuerung in jede Richtung
49

```



```

Fehler zwischen numerischer Simulation und exakter Lösung Vex alle )
CHECK_ERROR_STEP Schritte; 0:Keine Fehlerberechnung [Konvention: 0]
81 const int      CHECK_ERROR_STEP = 50;
82
83 //-----
84 //-- compute trajectory parameters
85 //-----
86 const int      TRAJPT          = 0;                //-- )
      1:Berechnet einen Vertreter der optimalen Trajektorien, welche in )
      der Vereinigung der erreichbaren Menge liegt; 0:Keine optimale )
      Trajektorie
87 const double   initialpoint[TRAJPT*DIM] = {0.0,1.0}; //-- )
      Anfangspunkt für den Vertreter der optimalen Trajektorien
88
89 //-----
90 //-- "coupe" : used only for exemples with d>=3 (cut into some plane )
      ==> results in files "coupe.dat"/"coupeex.dat")
91 //-----
92 const int      COUPE_DIMS[DIM] = {0 ,0 };
93 const double   COUPE_VALS[DIM] = {0.,0.};
94
95 //-----
96 //-- initial data
97 //-----
98
99 inline double   v0(const double * arg) {           )
      //-- Definition der Anfangswertemenge, hier kann )
      standardmäßig aus einem Ball oder ein Quadrat um den )
      theoretischen Anfangswert gewählt werden
100 double X0=1.0, Y0=0.0, RAYON=0.25;             )
      //-- Anfangswert und Radius bzw. Kantenlänge
101 double nx=sqrt( (arg[0]-X0)*(arg[0]-X0) )
      +(arg[1]-Y0)*(arg[1]-Y0))-RAYON; //-- Ball
102 // double nx=max( abs(arg[0]-X0) -RAYON ,abs(arg[1]-Y0) -RAYON); )
      //-- Quadrat
103 return MIN(VBORD,nx);
104 }
105
106 //-----
107 //-- dynamics and distributed cost functions used in the case of )
      COMMANDS=0 or 1 (assumes only 1 control)
108 //-----
109 inline void dynamics(const double* x, C u, double t, double* res){ )
      //-- Definition der Dynamik des zu betrachtenden )
      Kontrolproblems (Beachte die Zeitumkehr!)
110 double X=x[0], Y=x[1];
111 double f0,f1;
112 f0=-(-2*pi*Y)*u[0];
113 f1=-(+2*pi*X)*u[0];
114 res[0]=f0, res[1]=f1;
115 }
116

```

```

117 inline double distributed_cost(const double* arg, C u, double t){  }
      // - Definition des Kostenfunktional (Zur Berechnung )
      //   der erreichbaren Menge darf diese auf Null gesetzt werden.)
118   return 0.;
119 }
120
121
122 // -----
123 // - if COMMANDS=2 (2 player games): dynamics and distributed cost
124 // -----
125 inline void dynamics2(const double* x, C u, C u2, double t, double*  )
      res) { return; }
126 inline double distributed_cost2(const double* arg, C u, C u2, double  )
      t) { return 0.; }
127
128
129 // -----
130 // - Exact solution (if known)
131 // -----
132 inline double Vex(double t, const double* arg)  }
      // - Definition der exakten Lösung (falls bekannt)
133 {
134   double arg2[2];
135   double t1=-t;
136   arg2[0]=cos(2*pi*t1)*arg[0] - sin(2*pi*t1)*arg[1];
137   arg2[1]=sin(2*pi*t1)*arg[0] + cos(2*pi*t1)*arg[1];
138   return v0(arg2);
139 }
140
141 //! -----
142 //! - MORE ADVANCED PARAMETERS
143 //! -----
144
145 //! - Case COMMANDS=0: numerical Hamiltonian function Hnum.
146 inline void compute_Hconst(double* aMAX, double t){};
147 inline double Hnum(const double* x, const double* v, const double t){};
148
149 //! -----
150 //! - PARAMETERS FOR ADVANCED USERS
151 //! -----
152 const int    EXTERNALV0          = 0;          //! if 1 then starts )
      computation from data in VF.dat and not from the v0 function
153 const int    PRECOMPUTE.COORDS = 1;          //! to precompute mesh )
      coordinates (faster but needs more memory).
154
155 //! -----
156 //! - OBSTACLE g
157 //! -----
158 const int    OBSTACLE=0;
159 inline double g_obstacle(double t, const double* arg){ return 0.;}
160
161 //! -----

```

```

162  //!< - Special parameter for method SL
163  //!< -----
164  const int      P_INTERMEDIATE      = 1;    //!< number of discretisation ↵
        steps for approximating each trajectory in method MSL
165
166
167  //!< special parameters for SL METHOD
168  const int      TYPE_STA_LOOP       = NORMAL;
169  const int      INTERPOLATION       = BILINEAR;
170  const int      ORDER               = 1;
171  //!< special parameters for SL METHOD, SECOND ORDER (case ORDER=2)
172  const int      PARAMP              = 0;
173  inline double  funcR               (const double* x, C u, double t) { ↵
        return 0.; }
174  inline void    funcY               (const double* x, int k, double eps, ↵
        C u, double t, double h, double* res) { return; }
175  //!< -----
176  //!< - OTHER MAINLOOP PARAMETERS
177  //!< -----
178  const int      VALUE_PB            = 0;    //!< to compute value ↵
        v(t,x) s.t. 0 = w(t,x,z)=v(t,x)-z (assumes DIM=d+1, x in R^d, z in ↵
        R^1)
179  const int      SAVE_FOR_VALUE_PB   = 0;    //!< to save at each ↵
        iteration the value v(t,x) (assumes VALUE_PB=1)
180  const int      SAVE_FOR_VALUE_PB_FINAL = 0;    //!< to save final value ↵
        v(t,x) (assumes VALUE_PB=1)
181  //!< -----
182  //!< - OBSTACLE g tilde
183  //!< -----
184  const int      OBSTACLE_TILDE=0;
185  inline double  g_obstacle_tilde(double t, const double* arg){ return ↵
        0.;}
186  //!< -----
187  //!< - END TEMPORARY
188  //!< -----
189
190
191  //!< -----
192  //!< - For error computations
193  //!< -----
194  const double   C_THRESHOLD         = 0.2;    //!< Error threshold for ↵
        error computations
195
196  //!

```

7 Zusammenfassung

Im Rahmen dieser Arbeit wurde die Berechnung von Erreichbarkeitsmengen mit Hilfe des Hamilton-Jacobi-Bellman-Ansatzes untersucht. Dazu wurden stetige, jedoch nicht differenzierbare Wertefunktionen herangezogen, die zusammen mit dem Begriff der Viskositätslösungen eine deutlich verstärkte Anwendbarkeit des HJB-Ansatzes ermöglichen. Die in der klassischen Theorie sehr stark einschränkende Bedingung der stetigen Differenzierbarkeit der Wertefunktion verliert mit diesem Ansatz an Bedeutung. Trotz dieser abgeschwächten Bedingungen ist im Rahmen dieser Arbeit eine Existenz- und Eindeigkeitstheorie für die Wertefunktion als Viskositätslösung der HJB-Gleichung entwickelt worden. Zudem wurde die herkömmliche, meist zeitinvariante Theorie auf zeitvariante Kontrollsysteme erweitert.

Die Wertefunktion eines optimalen Steuerungsproblems erlaubt es, die Erreichbarkeitsmenge dieses Problems zu bestimmen. In der vorgestellten Software geschieht dies mit Hilfe des untersuchten Hamilton-Jacobi-Ansatzes. Allerdings wurde auf die numerische Belastbarkeit der in der Software verwendeten Algorithmen nicht eingegangen, da die Algorithmen zur Berechnung an sich in einer für mich vorkompilierten und geschützten Bibliothek vorlagen.

Auch bietet der Hamilton-Jacobi-Ansatz noch weiteres Forschungspotential: Innerhalb dieser Arbeit wurde dieser im Hinblick auf stetige, jedoch nicht differenzierbare Wertefunktionen untersucht. Darüber hinaus bietet es sich an, auch diese Voraussetzung abzuschwächen und die Stetigkeit als Voraussetzung komplett zu missachten. Dieser Ansatz wird auch in einigen Quellen, wie zum Beispiel in [BCD08] schon angesprochen und untersucht.

Generell ist die Untersuchung von erreichbaren Mengen im Hinblick auf die ständig zunehmende Automatisierung in unserem hochtechnisierten Zeitalter von großem praktischen Nutzen, da erreichbare Mengen mit vielen Sicherheitssystemen in unserem Alltag verknüpft sind. Hier spielt vor allem die schon genannte Kollisionsvermeidung einen offensichtlich wichtigen Anteil und fordert immer schnellere und effizientere Algorithmen zur Berechnung, um Ausfallsicherheit und Stabilität gewährleisten zu können.

A Anhang

Inhalt der DVD

Dieser Arbeit ist eine Double-Layer-DVD beigelegt, auf der sich eine PDF-Version der Diplomarbeit, ausgeführte Beispiele und weitere Unterlagen befinden.

Folgende Ordnerstruktur ist vorzufinden:

- **01_Diplomarbeit**
Hier befindet sich die finale Fassung dieser Diplomarbeit im PDF-Format.
- **02_Beispiele**
In diesem Ordner liegen alle während der Diplomarbeit ausgeführten Beispiele, geordnet und einsortiert in Unterordnern, nach den wichtigsten Parametern benannt. In diesen enthalten sind auch die verwendeten Header-Files, die jeweils protokollierte Bildschirmausgabe (Datei: `display.txt`) und alle Graphiken, konvertiert in verschiedenen Formaten. Ergänzend zu den im Text verwendeten Beispielen befinden sich dort auch noch weitere, welche nicht explizit in der Arbeit genannt worden sind.
- **03_Programm**
Dieser Ordner beinhaltet die zur Auswertung und Visualisierung verwendete Software „ROC-HJ-Solver“ der ITN-SADCO-Arbeitsgruppe um Prof. Bokanowski. Die Software liegt in Version 1.1 (veröffentlicht im Frühjahr 2013) vor. Zum Stand der Abgabe dieser Arbeit (Ende Juli 2014) war eine Version 2.0 bereits angekündigt. Je nach verfügbarer Prozessorarchitektur kann auf eine 32- bzw. 64-bit-Version zurückgegriffen werden. Die Archive enthalten das Originalprogramm ohne die Anpassungen, welche von mir für spezielle Ausgaben oder Graphiken vorgenommen wurden. Diese Änderungen können bei Bedarf selbst vorgenommen werden. Zur Anpassung und Verwendung des Programmcodes wird auf Kapitel 6 auf Seite 69ff., im Speziellen auf den Abschnitt zur Kompilation und Ausführung (S. 71ff.) sowie zu den Eingabeparametern (S. 77ff.) verwiesen.
- **04_Literatur**
In diesem Ordner sind zur besseren Übersicht alle Quellen und die gesamte Literatur enthalten, welche mir in digitaler Form zur Verfügung stand. Es wird kein Anspruch auf Vollständigkeit erhoben.

Abbildungsverzeichnis

2.1	Äquivalenzbeweis Viskositätslösung: Ringe R_n (nicht maßstabsgetreu)	14
2.2	Äquivalenzbeweis Viskositätslösung: Treppenfunktionen (Skizze)	14
5.1	Beispiel 1: Erreichbare Menge mit $NN=800$ und adaptiver zeitlicher Diskretisierung zum Zeitpunkt $T=0.5\pi$.	47
5.2	Beispiel 1: Erreichbare Menge mit $NN=800$ und adaptiver zeitlicher Diskretisierung zum Zeitpunkt $T=\pi$.	47
5.3	Beispiel 1: Erreichbare Menge mit $NN=800$ und adaptiver zeitlicher Diskretisierung zum Zeitpunkt $T=1.5\pi$.	47
5.4	Beispiel 1: Erreichbare Menge mit $NN=800$ und adaptiver zeitlicher Diskretisierung zum Zeitpunkt $T=2\pi$.	47
5.5	Beispiel 1: Erreichbare Menge mit $NN=100$ und adaptiver zeitlicher Diskretisierung $DT=0$.	48
5.6	Beispiel 1: Erreichbare Menge mit $NN=20$ und adaptiver zeitlicher Diskretisierung $DT=0$.	49
5.7	Beispiel 1: Erreichbare Menge mit $NN=25$ und adaptiver zeitlicher Diskretisierung $DT=0$.	49
5.8	Beispiel 1: Erreichbare Menge mit $NN=50$ und adaptiver zeitlicher Diskretisierung $DT=0$.	50
5.9	Beispiel 1: Erreichbare Menge mit $NN=800$ und adaptiver zeitlicher Diskretisierung $DT=0$.	50
5.10	Beispiel 1: Ausführungszeit in Abhängigkeit der Anzahl der Gitterpunkte (Interpolation und exponentielle Trendlinie).	50
5.11	Beispiel 2: Vereinigung der erreichbaren Mengen bis zum Zeitpunkt T (inklusive Plot der exakten Lösung) mit $NN=800$ und adaptiver zeitlicher Diskretisierung $DT=0$.	54
5.12	Beispiel 2: Erreichbare Menge zum Endzeitpunkt T (inklusive Plot der exakten Lösung) mit $NN=800$ und adaptiver zeitlicher Diskretisierung $DT=0$.	54
5.13	Beispiel 2: Auswertung der Kontur der Erreichbarkeitsmengen für verschiedene Parameter T im Intervall zwischen 0.00 (Startmenge) und 0.75 in Schritten von je 0.05 .	55
5.14	Beispiel 2: Vergleich des exakten Level-Sets der Wertefunktion mit der numerischen Approximation ($NN=800$ und adaptive zeitliche Diskretisierung $DT=0$).	57
5.15	Beispiel 2: Vereinigung der erreichbaren Mengen bis zum Zeitpunkt T mit $NN=10$ und adaptiver zeitlicher Diskretisierung $DT=0$.	58
5.16	Beispiel 2: Vereinigung der erreichbaren Mengen bis zum Zeitpunkt T mit $NN=15$ und adaptiver zeitlicher Diskretisierung $DT=0$.	58
5.17	Beispiel 2: Vereinigung der erreichbaren Mengen bis zum Zeitpunkt T mit $NN=20$ und adaptiver zeitlicher Diskretisierung $DT=0$.	58

5.18	Beispiel 2: Vereinigung der erreichbaren Mengen bis zum Zeitpunkt T mit $NN=25$ und adaptiver zeitlicher Diskretisierung $DT=0$	58
5.19	Beispiel 2: Vereinigung der erreichbaren Mengen bis zum Zeitpunkt T mit $NN=30$ und adaptiver zeitlicher Diskretisierung $DT=0$	59
5.20	Beispiel 2: Vereinigung der erreichbaren Mengen bis zum Zeitpunkt T mit $NN=40$ und adaptiver zeitlicher Diskretisierung $DT=0$	59
5.21	Beispiel 2: Vereinigung der erreichbaren Mengen bis zum Zeitpunkt T mit $NN=50$ und adaptiver zeitlicher Diskretisierung $DT=0$	59
5.22	Beispiel 2: Vereinigung der erreichbaren Mengen bis zum Zeitpunkt T mit $NN=100$ und adaptiver zeitlicher Diskretisierung $DT=0$	59
5.23	Beispiel 2: Erreichbare Menge zum Zeitpunkt T mit $NN=50$ und adaptiver zeitlicher Diskretisierung $DT=0$	60
5.24	Beispiel 2: Erreichbare Menge zum Zeitpunkt T mit $NN=100$ und adaptiver zeitlicher Diskretisierung $DT=0$	60
5.25	Beispiel 2: Erreichbare Menge zum Zeitpunkt T mit $NN=150$ und adaptiver zeitlicher Diskretisierung $DT=0$	61
5.26	Beispiel 2: Erreichbare Menge zum Zeitpunkt T mit $NN=200$ und adaptiver zeitlicher Diskretisierung $DT=0$	61
5.27	Beispiel 2: Erreichbare Menge zum Zeitpunkt T mit $NN=300$ und adaptiver zeitlicher Diskretisierung $DT=0$	61
5.28	Beispiel 2: Erreichbare Menge zum Zeitpunkt T mit $NN=400$ und adaptiver zeitlicher Diskretisierung $DT=0$	61
5.29	Beispiel 2: 3-dimensionaler Plot der Wertefunktion zum Zeitpunkt T mit $NN=50$ und adaptiver zeitlicher Diskretisierung $DT=0$	62
5.30	Beispiel 2: 3-dimensionaler Plot der Wertefunktion zum Zeitpunkt T mit $NN=100$ und adaptiver zeitlicher Diskretisierung $DT=0$	62
5.31	Beispiel 2: 3-dimensionaler Plot der Wertefunktion zum Zeitpunkt T mit $NN=150$ und adaptiver zeitlicher Diskretisierung $DT=0$	62
5.32	Beispiel 2: 3-dimensionaler Plot der Wertefunktion zum Zeitpunkt T mit $NN=200$ und adaptiver zeitlicher Diskretisierung $DT=0$	62
5.33	Beispiel 2: Auswertung der Kontur der Vereinigung der Erreichbarkeitsmengen für verschiedene NN	63
5.34	Beispiel 2: Zusammenhang zwischen Ausführungszeit und Anzahl der Gitterpunkte pro Koordinatenrichtung (mit Interpolation).	64
5.35	Beispiel 3: Entwicklung der Vereinigung der erreichbaren Mengen unter Verwendung einer festen zeitlichen Diskretisierung $DT=0.01$ und einer in 5-er Schritten variierenden räumlichen Diskretisierung ($NN=10-50$).	66
5.36	Beispiel 3: Detailansicht der Entwicklung der numerischen Explosion in 1-er Schritten; Anzahl der Gitterpunkte pro Koordinatenrichtung: $NN: 25-30$	68
6.1	Software Architektur	73

Tabellenverzeichnis

5.1	Beispiel 1: Parameter	46
5.2	Beispiel 1: Simulationsauswertung	51
5.3	Beispiel 2: Parameter	53
5.4	Beispiel 3: Parameter	65
5.5	Beispiel 3: Verletzung der CFL-Bedingung	67

Literatur

- [Bau00] J. Baumeister. *Steuerungstheorie*. Vorlesungsskript. Frankfurt am Main, 2000.
- [BBZ09] O. Bokanowski, A. Briani und H. Zidani. »Minimum time control problems for non-autonomous differential equations«. In: *Systems & Control Letters* 58.10–11 (2009), S. 742–746. ISSN: 0167-6911. URL: <http://www.sciencedirect.com/science/article/pii/S0167691109001066>.
- [BCD08] Martino Bardi und Italo Capuzzo Dolcetta. *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*. Modern Birkhäuser classics. Reprint of the 1997 edition. Boston [u.a.]: Birkhäuser, 2008, xvii + 570 S. : graph. Darst. ISBN: 0-8176-4754-6 - 978-0-8176-4754-4 - 978-0-8176-4755-1.
- [BDZ13a] Olivier Bokanowski, Anna Désillesy und Hasnaa Zidani. *ROC-HJ: Reachability analysis and Optimal Control problems - Hamilton-Jacobi equations*. 2013.
- [BDZ13b] Olivier Bokanowski, Anna Désillesy und Hasnaa Zidani. *User's guide for the ROC-HJ solver: Finite Differences and Semi-Lagrangian methods*. 2013.
- [BFZ10] O. Bokanowski, N. Forcadel und H. Zidani. »Reachability and Minimal Times for State Constrained Nonlinear Problems without Any Controllability Assumption«. In: *SIAM Journal on Control and Optimization* 48.7 (2010), S. 4292–4316. DOI: 10.1137/090762075. eprint: <http://dx.doi.org/10.1137/090762075>. URL: <http://dx.doi.org/10.1137/090762075>.
- [BFZ11] Olivier Bokanowski, Nicolas Forcadel und Hasnaa Zidani. »Deterministic state-constrained optimal control problems without controllability assumptions«. In: *ESAIM: Control, Optimisation and Calculus of Variations* 17 (04 Okt. 2011), S. 995–1015. ISSN: 1262-3377. DOI: 10.1051/cocv/2010030. URL: http://www.esaim-cocv.org/article_S1292811910000308.
- [Bre01] Alberto Bressan. »Viscosity solutions of Hamilton-Jacobi equations and optimal control problems«. In: *An illustrated tutorial. Original version: Trondheim, NTNU* (2001).
- [Cha03] Ilyes Aïssa Chahma. *Set-valued discrete approximation of state-constrained differential inclusions*. 67. Dissertation, Universität Bayreuth, Bayreuth, 2003. Mathematisches Institut der Universität Bayreuth, 2003, S. 3–161.
- [CL13] Adina Ciomaga und Jessica Lin. *Viscosity Solutions for Hamilton-Jacobi Equations. An Introduction*. Lecture notes. Chicago, 2013.

- [CL84] M. G. Crandall und P.-L. Lions. »Two approximations of solutions of Hamilton-Jacobi equations«. In: *Math. Comp.* 43.167 (1984), S. 1–19. ISSN: 0025-5718. DOI: 10.2307/2007396. URL: <http://dx.doi.org/10.2307/2007396>.
- [Cla+98] F. H. Clarke u. a. *Nonsmooth analysis and control theory*. Bd. 178. Graduate Texts in Mathematics. New York: Springer-Verlag, 1998, S. xiv+276. ISBN: 0-387-98336-8.
- [Cla89] F. H. Clarke. *Optimization and nonsmooth analysis*. Reprint of the 1983 original. Montreal, QC: Université de Montréal Centre de Recherches Mathématiques, 1989, S. xiv+312. ISBN: 2-921120-01-1.
- [Ger03] Matthias Gerds. *Nichtdifferenzierbare Optimierung*. Vorlesungsskript. Bayreuth, 2003.
- [Ger10] Matthias Gerds. *Optimale Steuerung*. Vorlesungsskript. Würzburg, 2010.
- [Gra12] Knut Graichen. *Methoden der Optimierung und optimalen Steuerung*. Vorlesungsskript. Ulm, 2012.
- [Grü04] Lars Grüne. *Viskositätslösungen von Hamilton–Jacobi–Bellman Gleichungen — eine Einführung*. Vortrag im Seminar “Numerische Dynamik von Kontrollsystemen”. Seminararbeit. Bayreuth, 2004/2005.
- [Grü06a] Lars Grüne. *Mathematische Kontrolltheorie I: Lineare Systeme*. Vorlesungsskript. Bayreuth, 2006.
- [Grü06b] Lars Grüne. *Mathematische Kontrolltheorie II: Nichtlineare Systeme*. Vorlesungsskript. Bayreuth, 2006.
- [Grü09] Lars Grüne. *Numerik Dynamischer Systeme*. Vorlesungsskript. Bayreuth, 2009.
- [Kru03] A.Ya. Kruger. »On Fréchet Subdifferentials«. English. In: *Journal of Mathematical Sciences* 116.3 (2003), S. 3325–3358. ISSN: 1072-3374. DOI: 10.1023/A:1023673105317. URL: <http://dx.doi.org/10.1023/A%3A1023673105317>.
- [Nat77] Isidor P. Natanson. *Theorie der Funktionen einer reellen Veränderlichen*. Unveränd. Nachdr. der 4. Aufl. ”Teorija funkcij ve s cestvennoj peremenoj Aus dem Russ. übers. Zürich [u.a.]: Deutsch, 1977, xii + 590 S. : graph. Darst. ISBN: 3-871-44217-8.
- [OF01] Stanley Osher und Ronald P. Fedkiw. »Level Set Methods: An Overview and Some Recent Results«. In: *Journal of Computational Physics* 169.2 (2001), S. 463–502. ISSN: 0021-9991. DOI: <http://dx.doi.org/10.1006/jcph.2000.6636>. URL: <http://www.sciencedirect.com/science/article/pii/S0021999100966361>.

- [Rod10a] Matthias Rodler. *Einführung in die Hamilton-Jacobi-Toolbox in MATLAB*. Seminararbeit. 2010.
- [Rod10b] Matthias Rodler. *Mengenwertige Runge-Kutta-Verfahren und Distanzfunktionen*. Seminararbeit. 2010.
- [Son98] Eduardo D. Sontag. *Mathematical control theory: deterministic finite dimensional systems*. 2. ed. Texts in applied mathematics ; 6. New York, NY [u.a.]: Springer, 1998, xvi + 531 S. : graph. Darst. ISBN: 0-387-98489-5.
- [Unr06] Carolin Unrath. *Nichtglatte Optimierungsstrategien in der dynamischen Programmierung*. Diplomarbeit. 2006.
- [Vin00] Richard B. Vinter. *Optimal control*. Systems & control: foundations & applications. Literaturverz. S. 493 - 504. Boston; Basel; Berlin: Birkhäuser, 2000, xv + 507 S. : graph. Darst. ISBN: 3-7643-4075-4 - 0-8176-4075-4.

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen.

Bayreuth, den 31. Juli 2014

.....
Matthias Rodler