

Fakultät für Mathematik und Physik

UNIVERSITÄT BAYREUTH

Professur für Angewandte Mathematik

Prof. Dr. Lars Grüne



UNIVERSITÄT
BAYREUTH

Diplomarbeit

Lokale Optimierungsstrategien in der dynamischen Programmierung

vorgelegt von
Michael Jarczyk

Betreuer : Prof. Dr. Lars Grüne
Professur für Angewandte Mathematik
Eingereicht am : 6. Juli 2005

Inhaltsverzeichnis

Kurzfassung	IX
1 Einleitung	1
2 Grundlagen der Dynamischen Systeme	3
2.1 Kontrollsysteme	3
2.2 Diskontierte Optimale Steuerung	8
2.3 Optimale Wertefunktion	11
3 Diskretisierung optimaler Steuerungsprobleme	17
3.1 Diskretisierung in der Zeit	17
3.1.1 Ein Iterationsverfahren	19
3.1.2 Zustandsraumbeschränkung	21
3.2 Diskretisierung im Raum	22
3.2.1 Funktionen auf Gittern	22
3.2.2 Die vollständige Diskretisierung	24
4 Berechnung approximativ optimaler Kontrollen	27
5 Optimierungsstrategien	29
5.1 Rekursive Suche	30
5.1.1 Bedingungen	30
5.1.2 Vorgehensweise	30
5.1.3 Aufwandsabschätzung	31

5.2	Brent-Verfahren	32
5.2.1	Bedingungen	33
5.2.2	Vorgehensweise	34
5.2.3	Aufwandsabschätzung	36
5.3	Erweitertes Brent-Verfahren	37
5.3.1	Bedingungen	38
5.3.2	Vorgehensweise	38
5.3.3	Abbruchkriterien	40
6	Numerische Tests	43
6.1	Bezeichnungen	43
6.2	Einfaches Modell	45
6.3	Investitionsmodell	47
6.4	Makroökonomisches Modell	51
6.5	Ökonomisches Wachstumsmodell	57
6.6	Räuber-Beute-Modell	67
7	Schluss	71
A	Beispiele	73
B	Programmbedienung	77
C	Verwendete Funktionen	81
D	Auswertungstabellen	85
E	CD-ROM Inhalt	111
	Literaturverzeichnis	113
	Erklärung	115

Abbildungsverzeichnis

3.1	Beispielgitter	23
5.1	Mögliche Konfiguration zu Beginn von <i>brent()</i>	34
5.2	Typische Konfiguration zum Schluß von <i>brent()</i>	36
5.3	Mögliche Konfiguration zu Beginn von <i>erw-b()</i>	39
6.1	Einfaches Modell - Wertefunktion	46
6.2	Einfaches Modell - Optimale Steuerung	46
6.3	Einfaches Modell - Differenz <i>diskr</i> [10] – <i>brent</i> [3]	47
6.4	Investitionsmodell - Wertefunktion	49
6.5	Investitionsmodell - Optimale Steuerung	49
6.6	Investitionsmodell - Differenz <i>diskr</i> [10] – <i>brent</i> [6]	49
6.7	Investitionsmodell - Differenz <i>diskr</i> [100] – <i>brent</i> [6]	50
6.8	Investitionsmodell - Differenz <i>erw-b</i> [3] – <i>brent</i> [3]	51
6.9	Investitionsmodell - Differenz <i>erw-b</i> [6] – <i>brent</i> [6]	51
6.10	Makroökonomisches Modell - 1d - Wertefunktion	53
6.11	Makroökonomisches Modell - 1d - Optimale Steuerung	53
6.12	Makroökonomisches Modell - 1d - Differenz <i>diskr</i> [100] – <i>brent</i> [25]	53
6.13	Makroökonomisches Modell - 1d - Differenz <i>erw-b</i> [6] – <i>brent</i> [6]	54
6.14	Makroökonomisches Modell - 2d - Wertefunktion	55
6.15	Makroökonomisches Modell - 2d - Optimale Steuerung	55
6.16	Makroökonomisches Modell - 2d - Differenz <i>diskr</i> [100] – <i>brent</i> [25]	56
6.17	Makroökonomisches Modell - 2d - Differenz <i>erw-b</i> [6] – <i>brent</i> [6]	56
6.18	Makroökonomisches Modell - 2d - Differenz <i>erw-b</i> [25] – <i>brent</i> [25]	56

6.19	Ökonomisches Wachstumsmodell - Wertefunktion	59
6.20	Ökonomisches Wachstumsmodell - Optimale Steuerung	59
6.21	Ökonomisches Wachstumsmodell - Differenz <i>exakt</i> – <i>diskr</i> [10]	59
6.22	Ökonomisches Wachstumsmodell - Differenz $u_{exakt} - u_{diskr[10]}$	59
6.23	Ökonomisches Wachstumsmodell - Differenz <i>exakt</i> – <i>diskr</i> [100]	60
6.24	Ökonomisches Wachstumsmodell - Differenz $u_{exakt} - u_{diskr[100]}$	60
6.25	Ökonomisches Wachstumsmodell - Differenz <i>exakt</i> – <i>rekur</i> [6]	60
6.26	Ökonomisches Wachstumsmodell - Differenz $u_{exakt} - u_{rekur[6]}$	60
6.27	Ökonomisches Wachstumsmodell - Differenz <i>exakt</i> – <i>erw-b</i> [6]	60
6.28	Ökonomisches Wachstumsmodell - Differenz $u_{exakt} - u_{erw-b[6]}$	60
6.29	Ökonomisches Wachstumsmodell - Differenz <i>exakt</i> – <i>brent</i> [6]	61
6.30	Ökonomisches Wachstumsmodell - Differenz $u_{exakt} - u_{brent[6]}$	61
6.31	Ökonomisches Wachstumsmodell - Differenz <i>rekur</i> [6] – <i>brent</i> [6]	61
6.32	Ökonomisches Wachstumsmodell - Differenz <i>erw-b</i> [6] – <i>brent</i> [6]	61
6.33	Räuber-Beute-Modell - Wertefunktion	69
6.34	Räuber-Beute-Modell - Optimale Steuerung	69
6.35	Räuber-Beute-Modell - Differenz <i>diskr</i> [10] – <i>brent</i> [6]	69
6.36	Räuber-Beute-Modell - Differenz <i>diskr</i> [100] – <i>brent</i> [6]	70
6.37	Räuber-Beute-Modell - Differenz <i>erw-b</i> [10] – <i>brent</i> [10]	70

Tabellenverzeichnis

5.1	Rekursive Suche - optimales n für $\epsilon = 0.01$	32
5.2	Rekursive Suche - optimales n für $\epsilon = 0.00001$	32
6.1	Einfaches Modell - Parameter	45
6.2	Einfaches Modell - Ergebnisse	45
6.3	Investitionsmodell - Parameter	48
6.4	Investitionsmodell - Ergebnisse	48
6.5	Makroökonomisches Modell - Parameter	52
6.6	Makroökonomisches Modell - 1d - Ergebnisse	52
6.7	Makroökonomisches Modell - 2d - Ergebnisse	54
6.8	Ökonomisches Wachstumsmodell - Parameter	57
6.9	Ökonomisches Wachstumsmodell - Ergebnisse	58
6.10	Brent-Maximierung am Beispiel von x_{368}	63
6.11	Brent-Maximierung am Beispiel von x_{369}	63
6.12	$erw-b()$ -Maximierung am Beispiel von x_{368}	65
6.13	$erw-b()$ -Maximierung am Beispiel von x_{369}	66
6.14	Räuber-Beute-Modell - Parameter	68
6.15	Räuber-Beute-Modell - Ergebnisse	68

Kurzfassung

Diese Arbeit baut auf einem in [13] vorgestellten Iterationsverfahren auf, welches die Lösung optimaler Steuerungsprobleme numerisch approximiert. Innerhalb dieses Verfahrens wird auf Optimierungsstrategien zur Maximumermittlung eingegangen, welche sich der parabolischen Interpolation bedienen. Ein zentrales Thema soll dabei das Verfahren von Brent spielen.

Wir erarbeiten zunächst die theoretischen Grundlagen der dynamischen Systeme, auf denen das Iterationsverfahren beruht. Es folgt eine genaue Betrachtung der Optimierungsstrategien, die anschließend ausführlich an ausgewählten Beispielen getestet und mit der bisherigen Vorgehensweise der äquidistanten Diskretisierung verglichen werden. Dabei wird deutlich, dass das Brent-Verfahren die effektivste Methode darstellt.

Kapitel 1

Einleitung

Es gibt in der heutigen Zeit kaum einen Menschen, dem der Begriff Optimierung fremd ist. Angesichts sinkender Zinssätze, Lohndumpings und schlechter Konjunkturlage sind es nicht nur Wissenschaftler und Unternehmen, die nach immer effizienteren Optimierungstrategien suchen. Die hohe Arbeitslosenquote zwingt immer mehr Familien ihre vorhandenen Mittel so einzusetzen, um den für sie daraus resultierenden Nutzen zu maximieren. Die Zeiten, in denen eine suboptimale Lösung den Ansprüchen der Meisten genügte, ist vorbei.

Es gibt also genügend Gründe sich mit der Effizienzuntersuchung von Optimierungsstrategien zu beschäftigen.

Gegenstand dieser Arbeit ist die Vorstellung und Untersuchung der Methode von Brent, sowie eines ähnlichen ebenfalls auf der parabolischen Interpolation beruhenden Verfahrens (Erweitertes Brent-Verfahren). Sie werden jedoch nicht isoliert betrachtet. Vielmehr soll ihre Effizienz als Baustein eines in [13] vorgestellten Iterationsverfahrens untersucht werden. Dieses Iterationsverfahren bediente sich bisher der äquidistanten Diskretisierung der Intervalle, um die maximale Wertefunktion eines optimalen Steuerungsproblems auf einem Gitter Γ approximativ zu berechnen. Die Diskretisierung soll nun durch die oben erwähnten Methoden ersetzt und die erhaltenen Ergebnisse hinsichtlich des benötigten Aufwands sowie der ermittelten Werte miteinander verglichen werden. Dies geschieht im Kapitel 6.

Doch zunächst wird die nötige Theorie vorgestellt (Kapitel 2-4). Nach einer Einführung der später benötigten Grundbegriffe, wie optimales Steuerungsproblem und optimale Wertefunktion, wird die für das Iterationsverfahren wichtige Diskretisierung des Steuerungsproblems sowohl im Raum als auch in der Zeit betrachtet.

Kapitel 5 widmet sich den Optimierungsstrategien zur Approximation des Maximums einer reellwertigen Funktion f auf einem Intervall $[a, b]$. Es wird auf das Verfahren von Brent, das erweiterte Brent-Verfahren sowie die rekursive Suche eingegangen. Die Vorstellung der Strategien und ihre numerische Auswertung (Kapitel 6) anhand von 5 Testbeispielen (siehe Anhang A) bilden den Schwerpunkt dieser Arbeit.

Eine Einschätzung der erhaltenen Ergebnisse bildet den Schluss der Arbeit.

Abschließend möchte ich mich bei Prof. Dr. Lars Grüne für die zahlreichen Anregungen und die ausgezeichnete Betreuung bedanken.

Kapitel 2

Grundlagen der Dynamischen Systeme

In diesem Kapitel werden die grundlegenden Definitionen und Sätze eingeführt, die zur Beschreibung des Hauptalgorithmus zur Lösung optimaler Steuerungsprobleme benötigt werden. Dabei wird auf Beweise weitgehend verzichtet, da diese bereits in vielen anderen Arbeiten geführt worden sind. Für den interessierten Leser wird jedoch ein Hinweis auf das Werk gegeben, in dem der Beweis nachzulesen ist. Insbesondere wird auf später benötigte Begriffe, wie Kontrollsysteme, das optimale Steuerungsproblem sowie die optimale Wertefunktion eingegangen.

2.1 Kontrollsysteme

Zunächst wird ein Kontrollsystem definiert.

2.1.1 Definition (Kontrollsystem)

(i) Ein Kontrollsystem in kontinuierlicher Zeit $T = \mathbb{R}$ im \mathbb{R}^d , $d \in \mathbb{N}$, ist gegeben durch die gewöhnliche Differentialgleichung

$$\frac{d}{dt}x(t) = f(x(t), u(t)), \quad (2.1)$$

wobei $f : \mathbb{R}^d \times U \rightarrow \mathbb{R}^d$ ein parameterunabhängiges stetiges Vektorfeld ist.

(ii) Ein Kontrollsystem in diskreter Zeit $T = h\mathbb{Z} = \{hk | k \in \mathbb{Z}\}$ für ein $h > 0$ im \mathbb{R}^d , $d \in \mathbb{N}$, ist gegeben durch die Differenzgleichung

$$x(t+h) = f_h(x(t), u(t)), \quad (2.2)$$

wobei $f_h : \mathbb{R}^d \times U \rightarrow \mathbb{R}^d$ eine stetige Abbildung ist.

(iii) Die Menge $U \subseteq \mathbb{R}^m$ heißt Kontrollwertebereich, und definiert die Werte, die $u(t)$ für $t \in \mathbb{R}$ annehmen darf.

(iv) Mit \mathcal{U} bzw. \mathcal{U}_h wird der Raum der zulässigen Kontrollfunktionen bezeichnet, also $\mathcal{U} := \{u : \mathbb{R} \rightarrow U | u \text{ zulässig}\}$ bzw. $\mathcal{U}_h := \{u_h : h\mathbb{Z} \rightarrow U | u \text{ zulässig}\}$

2.1.2 Bemerkung

Da in der Praxis keine unendlich kleinen oder großen Kontrollwerte realistisch sind, kann für den Rest der Arbeit die Menge U als kompakt vorausgesetzt werden.

Als nächstes wird die Frage der sinnvollen Bestimmung des Kontrollfunktionsraumes \mathcal{U} bzw. \mathcal{U}_h betrachtet. Dabei sollte sowohl eine möglichst große Anzahl von Funktionen zugelassen werden, als auch eine eindeutige Lösung existieren.

Der zeitdiskrete Fall ist relativ einfach zu lösen. Für \mathcal{U} werden alle möglichen Funktionen von $h\mathbb{Z}$ nach U zugelassen:

$$\mathcal{U}_h := \{u_h : h\mathbb{Z} \rightarrow U\}.$$

Die Induktion zeigt, dass dann für jeden Anfangswert $x_0 \in \mathbb{R}^d$ und jede Funktion $u_h \in \mathcal{U}_h$ eine eindeutige Lösung $\Phi(t, x_0, u_h)$ von (2.2) in positiver Zeitrichtung existiert.

D.h. für $\Phi : h\mathbb{N}_0 \times \mathbb{R}^d \times \mathcal{U}_h \rightarrow \mathbb{R}^d$ gilt:

$$\Phi_h(0, x_0, u_h) = x_0 \quad \text{und} \quad \Phi_h(t+h, x_0, u_h) = f_h(\Phi_h(t, x_0, u_h), u_h(t)). \quad (2.3)$$

Der kontinuierliche Fall gestaltet sich etwas schwieriger: Man könnte für \mathcal{U} die Menge aller stetigen Funktionen mit Werten in U nehmen. Zusammen mit der Bedingung, dass f Lipschitz-stetig in x ist, liesse sich dann für die Lösungen eine Existenz- und Eindeutigkeitsaussage machen. Dies ist aus der Theorie der gewöhnlichen Differentialgleichungen bekannt.

Jedoch ist dieses Kriterium für die Praxis zu streng. Bereits einfache Probleme besitzen optimale Steuerungsstrategien, die in t unstetig sind. Man muss Bedingungen also lockern und für \mathcal{U} den Raum der messbaren Funktionen, welche nachfolgend definiert werden, wählen.

2.1.3 Definition

Sei $I = [a, b] \subset \mathbb{R}$ ein abgeschlossenes Intervall.

- (i) Eine Funktion $g : I \rightarrow \mathbb{R}^m$ heisst *stückweise konstant*, falls eine Zerlegung von I in endlich viele Teilintervalle I_j , $j = 1, \dots, n$ existiert, so dass g auf I_j konstant ist $\forall j = 1, \dots, n$.
- (ii) Eine Funktion $g : I \rightarrow \mathbb{R}^m$ heisst (*Lebesque-*) *messbar*, falls eine Folge von stückweise konstanten Funktionen $g_i : I \rightarrow \mathbb{R}^m$, $i \in \mathbb{N}$, existiert mit $\lim_{i \rightarrow \infty} g_i(x) = g(x)$ für fast alle¹ $x \in I$.
- (iii) Eine Funktion $g : I \rightarrow \mathbb{R}^m$ heisst (*Lebesque-*) *messbar*, falls für jedes abgeschlossene Teilintervall $I = [a, b] \subset \mathbb{R}$ die Einschränkung $g|_I$ messbar im Sinne von (ii) ist.

Bleibe zu klären, ob die gelockerten Bedingungen immer noch eine sinnvolle Lösung für (2.1) liefern. Sie tun es, was der folgende Satz zeigt.

2.1.4 Satz (Satz von Carathéodory)

Betrachte ein Kontrollsystem mit folgenden Eigenschaften:

- (i) Der Raum der Kontrollfunktionen ist gegeben durch

$$\mathcal{U} = L_\infty(\mathbb{R}, U) := \{u : \mathbb{R} \rightarrow U \mid u \text{ ist messbar und essentiell beschränkt }^2\}.$$

¹d.h. für alle x aus einer Menge $J \subseteq I$ mit der Eigenschaft, dass $I \setminus J$ eine Lebesque-Nullmenge ist

²d.h. beschränkt außerhalb einer Lebesque-Nullmenge

(ii) Das Vektorfeld $f : \mathbb{R}^d \times U \rightarrow \mathbb{R}^d$ ist stetig.

(iii) Für jedes $R > 0$ existiert eine Konstante $L_R > 0$, so dass die Abschätzung

$$\| f(x_1, u) - f(x_2, u) \| \leq L_R \| x_1 - x_2 \|$$

$\forall x_1, x_2 \in \mathbb{R}^d$ und $\forall u \in U$ mit $\| x_1 \|, \| x_2 \|, \| u \| \leq R$ erfüllt ist.

Dann gibt es für jeden Punkt $x_0 \in \mathbb{R}^d$ und jede Kontrollfunktion $u \in U$ ein (maximales) offenes Intervall I mit $0 \in I$ und genau eine absolut stetige Funktion $x(t)$, die die Integralgleichung

$$x(t) = x_0 + \int_t^0 f(x(\tau), u(\tau)) d\tau$$

$\forall t \in I$ erfüllt.

Beweis: Ist z.B. in [19], Anhang C nachzulesen.

2.1.5 Definition

Wir nennen die eindeutige Funktion $x(t)$ aus Satz 2.1.4 die Lösung von (2.1) zum Anfangswert $x_0 \in \mathbb{R}^d$ und zur Kontrollfunktion $u \in \mathcal{U}$. Sie wird mit $\Phi(t, x_0, u)$ bezeichnet.

Im Folgenden wird ein Einschrittverfahren definiert, welches das kontinuierliche System (2.1) durch ein diskretes der Form (2.2) approximiert. Dabei wird die kontinuierliche Lösungsfunktion $\Phi(t, x_0, u)$ durch eine diskrete Gitterfunktion mit der Schrittweite $h > 0$ approximiert, deren Iterationsvorschrift f_h eine mit Hilfe des Computers auswertbare Abbildung darstellt. Solche numerischen Verfahren werden oft zur Lösung von gewöhnlichen Differentialgleichungen verwendet. In der Arbeit werden wir uns des einfachen Einschrittverfahren von Euler³ bedienen.

2.1.6 Definition (Euler-Verfahren für Kontrollsysteme)

Für einen Zeitschritt $h > 0$ und einen Kontrollwert $u \in U$ definieren wir das Euler Verfahren als das durch die Abbildung

$$f_h(x, u) := x + hf(x, u) \tag{2.4}$$

³Euler, Leonhard, schweiz. Mathematiker, *1707, †1783

definierte zeitdiskrete Kontrollsystem (2.2). Die Lösungen bezeichnen wir mit $\tilde{\Phi}_h(t, x_0, u_h)$.

Die Eigenschaften des Verfahrens faßt der folgende Satz zusammen.

2.1.7 Satz

Gegeben sei ein Kontrollsystem, für das die Voraussetzungen (i)-(iii) vom Satz 2.1.4 gelten, sowie $\bar{B}_R(0)$ ein abgeschlossener Ball mit Radius R um 0 im R^d . Dann gilt für das Verfahren aus Definition 2.1.6 und jede Konstante $R > 0$ die folgende Aussage:

- (i) Es existiert eine (von R unabhängige) Konstante $K > 0$, so dass für jede Kontrollfunktion $u \in \mathcal{U}$ mit $\|u\|_\infty \leq R$ und jeden Anfangswert $x_0 \in \bar{B}_R(0)$ eine diskrete Kontrollfunktion $u_h \in \mathcal{U}_h$ existiert, mit der Abschätzung

$$\|\tilde{\Phi}_h(t, x_0, u_h) - \Phi(t, x_0, u_h)\| \leq K\sqrt{he^{Lt}}$$

Sie gilt für alle $t \in h\mathbb{N}_0$ für die die Lösungen in $\bar{B}_R(0)$ liegen.

Ist das Kontrollsystem konvex, so gilt die schärfere Abschätzung

$$\|\tilde{\Phi}_h(t, x_0, u_h) - \Phi(t, x_0, u_h)\| \leq Kh(e^{Lt} - 1).$$

- (ii) Umgekehrt existiert eine von R abhängige Konstante $K > 0$, so dass für jedes $x_0 \in \bar{B}_R(0)$, jede diskrete Kontrollfunktion $u_h \in \mathcal{U}_h$ mit $\|u_h\|_\infty \leq R$ und die durch

$$u(\tau) := u_h(t), \quad \tau \in [t, t+h), \quad t \in h\mathbb{N}_0.$$

definierte stückweise konstante (also messbare) Kontrollfunktion die Abschätzung

$$\|\tilde{\Phi}_h(t, x_0, u_h) - \Phi(t, x_0, u_h)\| \leq Kh(e^{Lt} - 1).$$

gilt für alle $t \in h\mathbb{N}_0$ für die die Lösungen in $\bar{B}_R(0)$ liegen.

Beweis: Für den Fall eines konvexen Kontrollsystems nachzulesen in [13]. Für den langwierigen Beweis im allgemeinen Fall sei auf [10] verwiesen.

2.1.8 Bemerkung

Das Euler-Verfahren für Kontrollsysteme besitzt die Konvergenzordnung $O(\sqrt{h})$ bzw. $O(h)$.

2.2 Diskontierte Optimale Steuerung

Ziel der optimalen Steuerung ist es, die Kontrollfunktion $u \in \mathcal{U}$ in Abhängigkeit vom Anfangswert x_0 so zu wählen, dass der Wert, den man beim aufsummieren bzw. integrieren der Funktion entlang einer Trajektorie $\Phi(t, x_0, u)$ bzw. $\Phi_h(t, x_0, u)$ erhält, maximiert wird.

2.2.1 Definition (Optimales Steuerungsproblem)

Betrachte ein Kontrollsystem (2.1) bzw. (2.2). Für eine Funktion $g : \mathbb{R}^d \times U \rightarrow \mathbb{R}$ und einen Parameter $\delta > 0$ definieren wir das diskontierte Funktional auf unendlichem Zeithorizont in kontinuierlicher Zeit als

$$J(x, u) := \int_0^\infty e^{-\delta t} g(\Phi(t, x, u), u(t)) dt \quad (2.5)$$

und in diskreter Zeit als

$$J_h(x, u_h) := h \sum_{j=0}^{\infty} (1 - \delta h)^j g(\Phi_h(jh, x, u_h), u_h(jh)) \quad (2.6)$$

Das optimale Steuerungsproblem lautet nun: Bestimme die optimale Wertefunktion

$$v(x) := \sup_{u \in \mathcal{U}} J(x, u) \quad \text{bzw.} \quad v_h(x) := \sup_{u_h \in \mathcal{U}_h} J_h(x, u_h) \quad (2.7)$$

Hierbei werden folgende Annahmen gemacht:

- (i) Der Kontrollwertbereich U sei kompakt.
- (ii) In kontinuierlicher Zeit erfülle das Kontrollsystem (2.1) die Voraussetzungen (i)-(iii) von Satz 2.1.4, wobei die Lipschitz-Konstante $L_R = L$ unabhängig von R sei.

In diskreter Zeit existiere eine Konstante $L > 0$, so dass die Lipschitz-Abschätzung

$$\| f_h(x_1, u) - f_h(x_2, u) \| \leq (1 + Lh) \| x_1 - x_2 \|$$

$\forall x_1, x_2 \in \mathbb{R}^d, u \in U$ gilt.

(iii) Die Funktion g sei stetig und erfülle

$$|g(x, u)| \leq M_g \quad \text{und} \quad |g(x_1, u) - g(x_2, u)| \leq L_g \|x_1 - x_2\|$$

$\forall x, x_1, x_2 \in \mathbb{R}^d, u \in \mathcal{U}$ und geeignete Konstanten $M_g, L_g > 0$.

2.2.2 Bemerkung

(i) Bei der Minimierung lautet das optimale Steuerungsproblem analog

$$v(x) := \inf_{u \in \mathcal{U}} J(x, u) \quad \text{bzw.} \quad v_h(x) := \inf_{u_h \in \mathcal{U}_h} J_h(x, u_h)$$

(ii) Wir setzen hier nicht voraus, dass die optimalen Kontrollfunktionen $u \in U$ bzw. $u_h \in \mathcal{U}_h$ existieren, deshalb verwenden wir "sup" statt "max".

(iii) Man kann aus den Werten der optimalen Steuerung v bzw. v_h die optimale Kontrollfunktion u bzw. u_h berechnen. Es wird zum späteren Zeitpunkt nochmal darauf eingegangen.

(iv) Der sogenannte Diskontfaktor $e^{-\delta h}$ bzw. $1 - \delta h$ mit Diskontrate $\delta > 0$ stammt ursprünglich aus der Ökonomie. Er sorgt dafür, dass der Ertrag in naher Zukunft stärker gewichtet wird als derjenige in ferner Zukunft (liegt an der typischen Form einer Exponentialkurve).

Wichtig für den Algorithmus ist jedoch nicht die ökonomische Bedeutung, sondern die mathematische Auswirkung des Diskontfaktors auf die Lösung des optimalen Steuerungsproblems, welche im folgenden Lemma gezeigt wird.

2.2.3 Lemma

Das diskontierte Funktional ist endlich. Es gilt

$$|J(x, u)| \leq \frac{M_g}{\delta} \quad \text{und} \quad |J_h(x, u_h)| \leq \frac{M_g}{\delta}$$

insbesondere also auch $|v(x)|, |v_h(x)| \leq \frac{M_g}{\delta}$

Beweis: In kontinuierlicher Zeit gilt:

$$\begin{aligned}
 |J(x, u)| &= \left| \int_0^\infty e^{-\delta t} g(\Phi(t, x, u), u(t)) dt \right| \\
 &\leq \int_0^\infty |e^{-\delta t} g(\Phi(t, x, u), u(t))| dt \\
 &= \int_0^\infty e^{-\delta t} |g(\Phi(t, x, u), u(t))| dt \\
 &\leq \int_0^\infty e^{-\delta t} M_g dt \\
 &\leq M_g \int_0^\infty e^{-\delta t} dt \\
 &= M_g \left[-\frac{1}{\delta} e^{-\delta t} \right]_0^\infty = \frac{M_g}{\delta}
 \end{aligned}$$

In diskreter Zeit gilt:

$$\begin{aligned}
 |J_h(x, u_h)| &= \left| h \sum_{j=0}^{\infty} (1 - \delta h)^j g(\Phi_h(jh, x, u_h), u_h(jh)) \right| \\
 &\leq h \sum_{j=0}^{\infty} |(1 - \delta h)^j g(\Phi_h(jh, x, u_h), u_h(jh))| \\
 &= h \sum_{j=0}^{\infty} (1 - \delta h)^j |g(\Phi_h(jh, x, u_h), u_h(jh))| \\
 &\leq h \sum_{j=0}^{\infty} (1 - \delta h)^j M_g \\
 &= h M_g \sum_{j=0}^{\infty} (1 - \delta h)^j \\
 &= h M_g \frac{1}{\delta h} = \frac{M_g}{\delta}
 \end{aligned}$$

Nach Voraussetzung gilt $\delta h < 1$. Deswegen konvergiert die geometrische Reihe im vorletzten Schritt gegen $\frac{1}{\delta h}$.

□

2.3 Optimale Wertefunktion

Der nachfolgende Satz beinhaltet Aussagen zur Stetigkeit von optimalen Wertefunktionen.

2.3.1 Satz

Betrachte das optimale Steuerungsproblem aus der Definition 2.2.1. Ist $\delta > L$, so ist die optimale Wertefunktion Lipschitz stetig mit Konstante $\frac{Lg}{\delta-L}$. Ist $\delta \leq L$, so ist die optimale Wertefunktion Hölder stetig, d.h., es existieren Konstanten $K, \gamma > 0$ so dass $\forall x, y \in \mathbb{R}^d$ die Abschätzung

$$|v(x) - v(y)| \leq K \|x - y\|^\gamma$$

gilt. Hierbei ist $\gamma = \frac{\delta}{L}$, falls $\delta < L$ und $\gamma \in (0, 1)$ beliebig, falls $\delta = L$.

Beweis: Beruht auf der Abschätzung durch Konstanten, die von der Lipschitz-Konstante L abhängen. Ausführlich wird es in [13] gezeigt.

Die folgende Eigenschaft der optimalen Wertefunktion bildet die Basis für die numerische Approximation. Es handelt sich dabei um das Bellman'sche Optimalitätsprinzip⁴, welches auch das Prinzip der Dynamischen Programmierung genannt wird.

2.3.2 Satz (Bellmansches Optimalitätsprinzip)

Betrachte das optimale Steuerungsproblem aus Definition 2.2.1. dann erfüllt die optimale Wertefunktion $v(x)$ für jedes $x \in \mathbb{R}^d$ und jedes $T > 0$ die Gleichung

$$v(x) = \sup_{u \in \mathcal{U}} \left\{ \int_0^T e^{-\delta t} g(\Phi(t, x, u), u(t)) dt + e^{-\delta T} v(\Phi(T, x, u)) \right\} \quad (2.8)$$

bzw. für jedes $k \in \mathbb{N}$ die Gleichung

$$v_h(x) = \sup_{u_h \in \mathcal{U}_h} \left\{ \sum_{j=0}^k (1-\delta h)^j g(\Phi_h(jh, x, u_h), u_h(jh)) + (1-\delta h)^{k+1} v_h(\Phi_h((k+1)h, x, u_h)) \right\} \quad (2.9)$$

⁴Bellman, Richard E., amerik. Mathematiker, *1920, †1984

Beweis: Die Beweise verlaufen für beide Fälle analog. Deswegen wird hier nur der kontinuierliche Fall betrachtet. Im diskreten Fall betrachtet man statt der Integrale die entsprechenden Summen.

" \leq ": Seien $x \in \mathbb{R}^d, T > 0$ und $u \in \mathcal{U}$ beliebig. Dann gilt

$$\begin{aligned} J(x, u) &= \int_0^\infty e^{-\delta t} g(\Phi(t, x, u), u(t)) dt \\ &= \int_0^T e^{-\delta t} g(\Phi(t, x, u), u(t)) dt + \int_T^\infty e^{-\delta t} g(\Phi(t, x, u), u(t)) dt \\ &\leq \int_0^T e^{-\delta t} g(\Phi(t, x, u), u(t)) dt + e^{-\delta T} g(\Phi(T, x, u)) \end{aligned}$$

und da dies für jedes beliebige $u \in \mathcal{U}$ gilt, gilt die Ungleichung auch für das Supremum und damit für $v(x)$.

" \geq ": Seien $x \in \mathbb{R}^d, T > 0$ und $\epsilon > 0$ beliebig. Wähle $u_1 \in \mathcal{U}$ so, dass

$$\begin{aligned} &\sup_{u \in \mathcal{U}} \left\{ \int_0^T e^{-\delta t} g(\Phi(t, x, u), u(t)) dt + e^{-\delta T} v(\Phi(T, x, u)) \right\} \\ &\leq \int_0^T e^{-\delta t} g(\Phi(t, x, u_1), u_1(t)) dt + e^{-\delta T} v(\Phi(T, x, u_1)) + \epsilon \end{aligned}$$

Dadurch ist u_1 auf $[0, T]$ festgelegt. Wähle nun $u_1|_{(T, \infty)}$ so, dass

$$J(\Phi(T, x, u_1), u_1(T + \cdot)) \geq v(\Phi(T, x, u_1)) - \epsilon$$

Damit ergibt sich

$$\begin{aligned} &\sup_{u \in \mathcal{U}} \left\{ \int_0^T e^{-\delta t} g(\Phi(t, x, u), u(t)) dt + e^{-\delta T} v(\Phi(T, x, u)) \right\} \\ &\leq \int_0^T e^{-\delta t} g(\Phi(t, x, u_1), u_1(t)) dt + e^{-\delta T} J(\Phi(T, x, u_1), u_1(t)) dt + (1 + \epsilon^{-\delta T}) \epsilon \\ &\leq \int_0^T e^{-\delta t} g(\Phi(t, x, u_1), u_1(t)) dt + \int_T^\infty e^{-\delta t} g(\Phi(t, x, u_1), u_1(t)) dt + (1 + \epsilon^{-\delta T}) \epsilon \\ &= J(x, u_1) + (1 + \epsilon^{-\delta T}) \epsilon \leq v(x) + (1 + \epsilon^{-\delta T}) \epsilon \end{aligned}$$

und da $\epsilon > 0$ beliebig war, folgt somit die Behauptung. □

Der nächste Satz zeigt, dass $v(x)$ durch das Optimalitätsprinzip tatsächlich sogar eindeutig bestimmt ist. Dazu ist aber erst das folgende Lemma notwendig.

2.3.3 Lemma

Sei B eine beliebige Menge und betrachte zwei Abbildungen $a_1, a_2 : B \rightarrow \mathbb{R}$. Dann gelten die Abschätzungen

$$\left| \sup_{b_1 \in B} a_1(b_1) - \sup_{b_2 \in B} a_2(b_2) \right| \leq \sup_{b \in B} |a_1(b) - a_2(b)|.$$

Beweis: OBdA gelte:

$$\left| \sup_{b_1 \in B} a_1(b_1) - \sup_{b_2 \in B} a_2(b_2) \right| = \sup_{b_1 \in B} a_1(b_1) - \sup_{b_2 \in B} a_2(b_2)$$

Sei nun $\epsilon > 0$ beliebig. Wähle ein $b_\epsilon \in B$ so, dass gilt:

$$a_1(b_\epsilon) \geq \sup_{b_1 \in B} a_1(b_1) - \epsilon$$

Wegen $b_\epsilon \in B$ gilt offensichtlich

$$\sup_{b_2 \in B} a_2(b_2) \geq a_2(b_\epsilon)$$

und folglich

$$\begin{aligned} \sup_{b_1 \in B} a_1(b_1) - \sup_{b_2 \in B} a_2(b_2) &\leq a_1(b_\epsilon) - a_2(b_\epsilon) + \epsilon \\ &\leq |a_1(b_\epsilon) - a_2(b_\epsilon)| + \epsilon \\ &\leq \sup_{b \in B} |a_1(b) - a_2(b)| + \epsilon \end{aligned}$$

Da $\epsilon > 0$ beliebig war, folgt somit die Behauptung. □

Nun können wir folgenden Satz beweisen.

2.3.4 Satz

Betrachten wir das optimale Steuerungsproblem aus Definition 2.2.1 mit optimaler Wertefunktion v . Sei $T > 0$ gegeben und sei $w : \mathbb{R}^d \rightarrow \mathbb{R}$ eine beschränkte Funktion, die das Optimalitätsprinzip

$$w(x) = \sup_{u \in \mathcal{U}} \left\{ \int_0^T e^{-\delta t} g(\Phi(t, x, u), u(t)) dt + e^{-\delta T} w(\Phi(T, x, u)) \right\} \quad (2.10)$$

(bzw. das Gegenstück in diskreter Zeit) $\forall x \in \mathbb{R}^d$ erfüllt. Dann ist

$$w = v \quad \text{bzw.} \quad w = v_h$$

Beweis: Wir setzen $\forall x \in \mathbb{R}^d$

$$a_1(u) = \int_0^T e^{-\delta t} g(\Phi(t, x, u), u(t)) dt + e^{-\delta T} w(\Phi(T, x, u))$$

und

$$a_2(u) = \int_0^T e^{-\delta t} g(\Phi(t, x, u), u(t)) dt + e^{-\delta T} v(\Phi(T, x, u))$$

Mit Hilfe von Lemma 2.3.3 erhalten wir die Ungleichung

$$\begin{aligned} |w(x) - v(x)| &= \left| \sup_{u \in \mathcal{U}} a_1(u) - \sup_{u \in \mathcal{U}} a_2(u) \right| \leq \sup_{u \in \mathcal{U}} |a_1(u) - a_2(u)| \\ &= \sup_{u \in \mathcal{U}} \left| \int_0^T e^{-\delta t} g(\Phi(t, x, u), u(t)) dt + e^{-\delta T} w(\Phi(T, x, u)) \right. \\ &\quad \left. - \int_0^T e^{-\delta t} g(\Phi(t, x, u), u(t)) dt + e^{-\delta T} v(\Phi(T, x, u)) \right| \\ &= \sup_{u \in \mathcal{U}} \left\{ e^{-\delta T} |w(\Phi(T, x, u)) - v(\Phi(T, x, u))| \right\} \\ &\leq e^{-\delta T} \sup_{y \in \mathbb{R}^d} |w(y) - v(y)|. \end{aligned}$$

Da dies $\forall x \in \mathbb{R}^d$ gilt, folglich auch für

$$\sup_{y \in \mathbb{R}^d} |w(y) - v(y)| \leq e^{-\delta T} \sup_{y \in \mathbb{R}^d} |w(y) - v(y)|.$$

Es folgt

$$(1 - e^{-\delta t}) \sup_{y \in \mathbb{R}^d} |w(y) - v(y)| \leq 0.$$

Da $e^{-\delta t} < 1$ ist, folgt $\sup_{y \in \mathbb{R}^d} |w(y) - v(y)| = 0$. Also ist $w = v$.

□

Kapitel 3

Diskretisierung optimaler Steuerungsprobleme

Dieses Kapitel verfolgt das Ziel, das optimale Steuerungsproblem derart zu verändern, dass eine sowohl schnelle als auch sinnvolle Lösung erreicht werden kann. Das numerische Verfahren dazu wird ebenfalls vorgestellt. Im Wesentlichen besteht es aus zwei Schritten. Zuerst wird das Steuerungsproblem in der Zeit diskretisiert. Dabei wird ein zeitkontinuierliches Problem durch ein zeitdiskretes approximiert. Liegt das Problem in zeitdiskreter Form bereits vor, wird dieser Schritt ausgelassen. Als nächstes erfolgt eine Diskretisierung im Raum. Da jede Funktion aus unendlich vielen Punkten besteht, wird über den Definitionsbereich der Funktion ein Gitter Γ gelegt und die Funktion an den Gitterpunkten evaluiert. Dass diese Maßnahmen trotzdem eine vernünftige Lösung liefern wird ebenso gezeigt.

3.1 Diskretisierung in der Zeit

Aus Definition 2.1.6 und Satz 2.1.7 ist uns bekannt, dass wir zu einem Kontrollsystem eine diskrete Euler-Approximation konstruieren können, die zu jedem Anfangswert $x \in \mathbb{R}^d$ und zu jeder diskreten Kontrollfunktion $u_h : h\mathbb{Z} \rightarrow U$ eine zeitdiskrete approximative Lösung $\tilde{\Phi}_h(t, x, u_h)$ liefert. Für ein gegebenes kontinuierliches optimales

Steuerungsproblem betrachten wir nun das zu dieser Euler-Approximation gehörige diskrete optimale Steuerungsproblem gegeben durch

$$\tilde{v}_h(x) := \sup_{u_h \in \mathcal{U}_h} \tilde{J}_h(x, u_h)$$

mit

$$\tilde{J}_h(x, u_h) := h \sum_{j=0}^{\infty} (1 - \delta h)^j g(\tilde{\Phi}_h(jh, x, u_h), u_h(jh)). \quad (3.1)$$

Folgender Satz zeigt, dass die Approximation $\tilde{v}_h(x)$ gegen $v(x)$ konvergiert, wenn Zeitschritt $h \rightarrow 0$.

3.1.1 Satz

Betrachte ein optimales Steuerungsproblem aus Definition 2.2.1 sowie das zugehörige Euler-diskretisierte optimale Steuerungsproblem (3.1). Wir nehmen an, dass das zugrundeliegende Kontrollsystem die Voraussetzungen von Satz 2.1.4 und Satz 2.1.7 erfüllt. Dann gelten für die optimalen Wertefunktionen v und \tilde{v}_h mit passender Konstante $K > 0$.

$$(i) \quad v(x) \leq \tilde{v}_h(x) + K(h^{\frac{\gamma}{2}} + h) \text{ und}$$

$$(ii) \quad \tilde{v}_h(x) \leq v_h(x) + K(h^\gamma + h)$$

$\forall h \in [0, \frac{1}{\delta}[$ und alle $x \in \mathbb{R}^d$. $\gamma \in]0, 1]$ sei die Hölder-Konstante aus Satz 2.3.1.

Man kann also für alle $x \in \mathbb{R}^d$ schreiben:

$$|v(x) - \tilde{v}_h(x)| \leq \tilde{K} h^{\frac{\gamma}{2}} \quad (3.2)$$

mit passendem $\tilde{K} > 0$.

Beweis: Der konvexe Fall eines optimalen Steuerungsproblems ist in [13] nachzulesen. Für den Beweis im nicht-konvexen Fall sei auf [10] verwiesen.

3.1.2 Bemerkung

Nach Satz 2.2.1 erfüllt \tilde{v}_h das Optimalitätsprinzip

$$\begin{aligned} \tilde{v}_h(x) = \sup_{u_h \in \mathcal{U}_h} \left\{ h \sum_{i=0}^k \beta^i g(\tilde{\Phi}_h(ih, x, u_h), u_h(ih)) \right. \\ \left. + \beta^{k+1} \tilde{v}_h(\tilde{\Phi}_h((k+1)hx, u_h)) \right\} \end{aligned} \quad (3.3)$$

Da $\tilde{\Phi}_h$ und g stetig in $(u_h(0), \dots, u_h(k))$ sind und \tilde{v}_h stetig ist, können wir das Supremum hier tatsächlich als Maximum schreiben. Insbesondere folgt, dass wir für $k = 0$ zu jedem $x \in \mathbb{R}^d$ mindestens ein $u_x^* \in U$ finden, so dass das Supremum in (3.4) für ein $u_h \in \mathcal{U}_h$ mit $u(0) = u_x^*$ angenommen wird.

3.1.1 Ein Iterationsverfahren

Ausgehend vom Optimalitätsprinzip (3.4) lässt sich für zeitdiskrete und entsprechend für zeitdiskretisierte, optimale Steuerungsprobleme ein Iterationsverfahren definieren, dessen errechnete Funktionenfolge gegen die optimale Wertefunktion v_h konvergiert.

3.1.3 Definition

Wir definieren iterativ Funktionen $v_h^i : \mathbb{R}^d \rightarrow \mathbb{R}$, $i = 0, 1, \dots$ mittels $v_h^0(x) = 0$ und $v_h^{i+1}(x) = T_h(v_h^i)(x) \quad \forall x \in \mathbb{R}^d$, wobei der Operator $T_h : C(\mathbb{R}^d, \mathbb{R}) \rightarrow C(\mathbb{R}^d, \mathbb{R})$ gegeben ist durch

$$T_h(w)(x) := \max_{u \in U} \{ hg(x, u) + \beta w(f_h(x, u)) \} \quad (3.4)$$

mit $\beta = 1 - \delta h$. Hierbei bezeichnet $C(\mathbb{R}^d, \mathbb{R})$ die Menge der stetigen Funktionen von \mathbb{R}^d nach \mathbb{R} .

Nachfolgend wird gezeigt, dass die durch diese Iteration erzeugte Funktionenfolge tatsächlich gegen v_h konvergiert.

3.1.4 Satz

Betrachte das zeitdiskretisierte optimale Steuerungsproblem (3.1) mit optimaler Wertefunktion v_h . Es sei $\delta h < 1$. Dann gilt für die in Definition 3.1.3 definierten Funktionen die Abschätzung

$$\|v_h^i - v_h\|_\infty \leq \beta^i \frac{Mg}{\delta}.$$

Wegen $\beta < 1$ folgt also insbesondere die Konvergenz $v_h^i(x) \rightarrow v_h(x)$ gleichmäßig für alle $x \in \mathbb{R}^d$.

Beweis: Seien $w_1, w_2 : \mathbb{R}^d \rightarrow \mathbb{R}$ zwei beliebige Funktionen. Mit Hilfe von Lemma 2.3.3 folgt:

$$\begin{aligned} |T_h(w_1)(x) - T_h(w_2)(x)| &= \\ &= \left| \max_{u \in U} \{hg(x, u) + \beta w_1(f_h(x, u))\} - \max_{u \in U} \{hg(x, u) + \beta w_2(f_h(x, u))\} \right| \\ &= \left| \sup_{u \in U} \{hg(x, u) + \beta w_1(f_h(x, u))\} - \sup_{u \in U} \{hg(x, u) + \beta w_2(f_h(x, u))\} \right| \\ &\leq \sup_{u \in U} |\beta w_1(f_h(x, u)) - \beta w_2(f_h(x, u))| \\ &= \beta \sup_{u \in U} |w_1(f_h(x, u)) - w_2(f_h(x, u))| \\ &\leq \beta \|w_1 - w_2\|_\infty. \end{aligned}$$

Daraus folgt

$$\|T_h(w_1)(x) - T_h(w_2)(x)\|_\infty \leq \beta \|w_1 - w_2\|_\infty.$$

Mit Hilfe des Optimalitätsprinzips (3.3) für $k = 0$ und (2.3) folgt die Gleichung $v_h = T_h(v_h)$. Damit und mit der Definition von v_h^{i+1} erhalten wir

$$\|v_h - v_h^{i+1}\|_\infty = \|T_h(v_h) - T_h(v_h^i)\|_\infty \leq \beta \|v_h - v_h^i\|_\infty. \quad (3.5)$$

Mit Lemma 2.2.3 und $j = 0$ können wir schreiben

$$\|v_h - v_h^0\|_\infty = \|v_h\|_\infty \leq \frac{Mg}{\delta} = \beta^0 \frac{Mg}{\delta} \quad (3.6)$$

Mit (3.6) als Induktionsanfang und (3.5) als Induktionsschritt ist der Satz nun ganz leicht per Induktion über j zu beweisen.

□

3.1.5 Bemerkung

Zusätzlich lässt sich zeigen, dass die erhaltenen Funktionen v_h^i für alle i Lipschitzstetig sind.

3.1.2 Zustandsraumbeschränkung

Wie bereits angedeutet, lässt sich v_h nicht im ganzen \mathbb{R}^d numerisch berechnen. Deswegen muss man den Definitionsbereich von v_h bzw. die zeitliche Diskretisierung \tilde{v}_h auf eine kompakte Menge $\Omega \subset \mathbb{R}^d$ einschränken. Zusätzlich werden nur die Lösungen betrachtet, die für alle positiven Zeiten in Ω bleiben. In der Praxis ergibt sich die geeignete Menge Ω oftmals aus der Aufgabenstellung z.B. aus dem physikalisch interessanten Bereich. Formal muss man jedoch einiges beachten. Wir unterscheiden hierbei 3 Fälle:

- (i) Ω ist stark invariant, d.h. $\forall x \in \Omega$ und $\forall u \in U$ ist $f_h(x, u) \in \Omega$. Hierbei ergeben sich keine Probleme und die Einschränkung von v_h auf Ω bewirkt bei v_h keine Änderung.
- (ii) Ω ist schwach invariant, d.h. $\forall x \in \Omega$ gibt es mindestens ein $u \in U$ mit $f_h(x, u) \in \Omega$. In diesem Fall optimiert man nur über diese $u \in U$, also über eine kleinere Menge. Folglich wird die Funktion v_h kleiner.
- (iii) Ω ist nicht invariant, d.h. $\exists x \in \Omega$, so dass $\forall u \in U$ gilt: $f_h(x, u) \notin \Omega$. In diesem Fall bietet sich an, $f_h(x, u)$ durch den nächstgelegenen Punkt in Ω zu ersetzen. Jedoch kann die erhaltene Lösung stark von der tatsächlichen optimalen Wertefunktion abweichen.

3.2 Diskretisierung im Raum

Obwohl wir alle benötigten Parameter zur Berechnung der Funktionswerte haben, ist es nicht möglich eine unendliche Zahl von Funktionswerten zu evaluieren. Auch wenn es möglich wäre, so ist eine solche Vorgehensweise nicht sinnvoll. Sogar die kompakte Menge Ω ist immer noch zu groß, um damit effizient arbeiten zu können. Deswegen wird der Raum \mathbb{R}^d auf einem Gitter der Dimension d diskretisiert. Damit verfügt man dann über eine endliche Anzahl von Werten, die die Evaluation von v_h leichter machen.

3.2.1 Funktionen auf Gittern

Der zu betrachtende Funktionenraum muss also endlich sein. Im Folgenden wird die Betrachtung auf $d = 2$ beschränkt, da auch die später untersuchten Beispiele auf \mathbb{R}^2 operieren.

3.2.1 Definition

Sei $\Omega \subset \mathbb{R}^2$ gegeben durch $\Omega = [a_1, b_1] \times [a_2, b_2]$ mit Werten $a_1 < b_1$ und $a_2 < b_2$. Ein regelmäßiges Rechteck Γ auf Ω ist eine Menge von Rechtecken R_i , $i = 0, \dots, P - 1$, $P = P_1 P_2$, mit Kantenlängen $k_1 = \frac{(b_1 - a_1)}{P_1}$ und $k_2 = \frac{(b_2 - a_2)}{P_2}$, so dass

$$\bigcup_{i=0}^{P-1} R_i = \Omega \text{ und } \text{int } R_i \cap \text{int } R_j = \emptyset, \forall i, j = 0, \dots, P - 1, i \neq j.$$

Mit E_i , $i = 0, \dots, N - 1$, $N = (P_1 + 1)(P_2 + 1)$ bezeichnen wir die Eckpunkte (oder Knotenpunkte) des Gitters. Der Wert $k = \sqrt{k_1^2 + k_2^2}$ bezeichnet den maximalen Durchmesser eines Rechtecks.

Die Abbildung 3.1 zeigt solch ein Beispiegelgitter. Wir definieren nun den Funktionenraum, den wir zur Approximation von v_h verwenden wollen.

3.2.2 Definition

- (i) Sei $A \subset \mathbb{R}^2$. Eine Funktion $\omega : A \rightarrow \mathbb{R}$ heisst affin bilinear, falls es Konstanten $\alpha_0, \dots, \alpha_3$ gibt, so dass für alle $x = (x_1, x_2)^T \in A$ die Identität $\omega(x) = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_1 x_2$ gilt.

(ii) Betrachte eine rechteckförmige Menge $\Omega \in \mathbb{R}^2$ mit Rechteckgitter Γ . Wir definieren den Raum der stetigen und stückweise affin bilinearen Funktionen auf Ω bezüglich Γ als

$$\mathcal{W} := \{w : \Omega \rightarrow \mathbb{R} \mid w \text{ ist stetig und } w|_{\mathbb{R}_i} \text{ ist affin bilinear für jedes } i = 0, \dots, P-1\}.$$

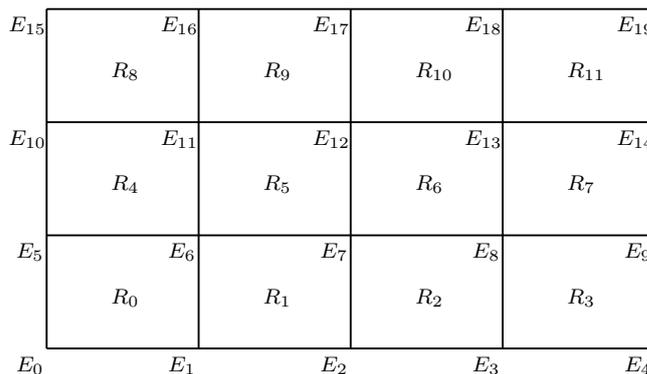


Abbildung 3.1: Beispielgitter

Das folgende Lemma fasst die für uns wichtigen Eigenschaften von \mathcal{W} zusammen.

3.2.3 Lemma

(i) Jede Funktion $w \in \mathcal{W}$ ist eindeutig durch ihre Werte $w(E_i)$ in den Eckpunkten des Gitters bestimmt.

(ii) Für jedes Rechteck $R_i = [c_1, d_1] \times [c_2, d_2]$ mit den Eckpunkten

$$E_{i_0} = (c_1, c_2)^T, \quad E_{i_1} = (d_1, c_2)^T, \quad E_{i_2} = (c_1, d_2)^T, \quad E_{i_3} = (d_1, d_2)^T$$

lässt sich $w|_{\mathbb{R}_i}$ für $x = (x_1, x_2)^T \in \mathbb{R}_i$ schreiben als

$$w(x) = \sum_{j=0}^3 \mu_j(x) w(E_{i_j}) \tag{3.7}$$

mit

$$\begin{aligned} \mu_0(x) &= (1 - y_1(x))(1 - y_2(x)), & \mu_1(x) &= y_1(x)(1 - y_2(x)), \\ \mu_2(x) &= (1 - y_1(x))y_2(x), & \mu_3(x) &= y_1(x)y_2(x) \end{aligned}$$

und

$$y_l(x) = \frac{x_l - c_l}{d_l - c_l} \quad \text{für } l = 1, 2.$$

Insbesondere gilt hierbei $\mu_j(x) \geq 0$ für $j = 0, \dots, 3$ und $\sum_{j=0}^3 \mu_j(x) = 1$.

3.2.2 Die vollständige Diskretisierung

In diesem Abschnitt wird der iterative Algorithmus aus Definition 3.1.3 für Funktionen aus \mathcal{W} formuliert. Wie wir in Lemma 3.2.3 gesehen haben, reicht es, die Werte in den Eckpunkten des Gitters zu berechnen. Der nur in den Eckpunkten E_i , $i = 0, \dots, N-1$ auszuwertende Operator T_h liefert dann eine Folge von Funktionen $\hat{v}_h^j \in \mathcal{W}$ mittels

$$\hat{v}_h^{j+1}(E_i) = \max_{u \in U} \{hg(E_i, u) + \beta \hat{v}_h^j(f_h(E_i, u))\}$$

Schreibt man $V^j = (V_1^j, \dots, V_N^j)^T \in \mathbb{R}^N$ mit $V_i^j = \hat{v}_h^j(E_i)$, so kann diese Iteration auf \mathcal{W} nun als eine Iteration auf N -dimensionalen Vektoren formuliert werden. Die folgende Definition liefert die dazugehörige Vorschrift.

3.2.4 Definition

Betrachte ein zeitdiskretes optimales Steuerungsproblem und ein Rechteckgitter Γ mit P Rechtecken und N Eckpunkten. Zu jedem $u \in U$ und jedem $i = 0, \dots, N-1$ sei $B(i, u)$ der N -dimensionale Zeilenvektor, für den für jedes $w \in \mathcal{W}$ und $W = (w(E_0), \dots, w(E_{N-1}))^T \in \mathbb{R}^N$ mit der üblichen Matrixmultiplikation gilt

$$w(\Phi_h(E_i, u)) = B(i, u)W \tag{3.8}$$

(beachte, dass $B(i, u)$ unabhängig von $w \in \mathcal{W}$ ist und höchstens 4 Einträge $\neq 0$ besitzt, nämlich gerade μ_l aus Lemma 3.2.3 in globaler Nummerierung, welche ≥ 0 sind und sich zu 1 aufsummieren). Desweiteren sei $G(i, u) = hg(E_i, u)$. Dann berechnen wir Vektoren V^j iterativ durch $V^0 := (0, \dots, 0)^T$

und das Gesamtschrittverfahren

$$V_i^{j+1} := \max_{u \in U} \{G(i, u) + \beta B(i, u)V^j\} \quad \text{für } i = 0, \dots, N-1 \quad (3.9)$$

oder das Einzelschrittverfahren

$$V^{j+1} := V^j, \quad V_i^{j+1} := \max_{u \in U} \{G(i, u) + \beta B(i, u)V^{j+1}\} \quad \text{für } i = 0, \dots, N-1. \quad (3.10)$$

3.2.5 Bemerkung

Das für die Berechnung der Daten im Kapitel 6 implementierte C++-Programm verwendet das Einzelschrittverfahren.

Das folgende Lemma gibt ein Abbruchkriterium für diese Iteration.

3.2.6 Lemma

Betrachte die Iterationsvorschrift aus Definition 3.2.4 und sei $\delta h < 1$. Dann konvergieren die Vektoren V^j für $j \rightarrow \infty$ komponentenweise gegen den Vektor V , der eindeutig bestimmt ist durch

$$V_i = \max_{u \in U} \{G(i, u) + \beta B(i, u)V\} \quad \text{für } i = 0, \dots, N-1. \quad (3.11)$$

Für die mit \hat{v}_h^j , $j = 1, \dots, \infty$ und \hat{v}_h bezeichneten zugehörigen Funktionen aus \mathcal{W} gilt darüberhinaus: Falls $|V_i^j - V_i^{j+1}| \leq \epsilon$ für alle $i = 0, \dots, N-1$, so folgt

$$\|\hat{v}_h^j - \hat{v}_h\|_\infty \leq \frac{\epsilon}{h\delta}. \quad (3.12)$$

Beweis: Nach Lemma 2.3.3 folgt für beliebige Vektoren $V, W \in \mathbb{R}^N$ und alle $i = 0, \dots, N-1$ die Ungleichung

$$\left| \max_{u \in U} \{G(i, u) + \beta B(i, u)V\} - \max_{u \in U} \{G(i, u) + \beta B(i, u)W\} \right| \leq \beta \|V - W\|_\infty \quad (3.13)$$

Dabei bezeichnet $\|\cdot\|_\infty$ die L_∞ -Norm im \mathbb{R}^N und $\beta = 1 - \delta h$. Zusätzlich wird $\sum_{k=0}^{N-1} B(i, u)_k = 1$ ausgenutzt. Also definiert (3.13) eine Kontraktion auf dem \mathbb{R}^N bzgl. der L_∞ -Norm, weswegen der Vektor V existiert. Er ist auch eindeutig, denn für jeden weiteren solchen Vektor W gilt mit (3.13)

$$\|V - W\|_\infty \leq \beta \|V - W\|_\infty < \|V - W\|_\infty$$

was zu der Folgerung führt, dass $V = W$. Man sieht auch, dass die Vektoren V^j die Abschätzung

$$\|V^{j+1} - V\|_\infty \leq \beta \|V^j - V\|_\infty$$

erfüllen. Daraus folgt die behauptete Konvergenz.

Ausserdem folgt

$$\begin{aligned} \|V^j - V\|_\infty &\leq \beta \|V^j - V^{j+1}\|_\infty + \|V^{j+1} - V\|_\infty \\ &\leq \epsilon + \beta \|V^j - V^\infty\|_\infty \end{aligned}$$

und daraus

$$\|V^j - V\|_\infty \leq \frac{\epsilon}{1 - \beta} = \frac{\epsilon}{h\delta}$$

Die entsprechende Aussage für die Funktionen \hat{v}_h^j folgt nun leicht mit der Darstellung aus Lemma 3.2.3(ii).

□

Kapitel 4

Berechnung approximativ optimaler Kontrollen

Neben der optimalen Wertefunktion können wir auch die optimale Kontrolle berechnen, die es uns ermöglicht das Maximum einer Wertefunktion zu erreichen. Dieses wird in diesem Kapitel kurz vorgestellt.

Aus dem Abschnitt über die vollständige Diskretisierung [3.2.2] wissen wir, dass

$$\hat{v}_h(x) = \max_{u \in U} \{hg(x, u) + \beta \hat{v}_h(f_h(x, u))\} \quad (4.1)$$

Nun definieren wir eine numerische Kontrollstrategie \hat{u}^* .

4.0.7 Definition

Sei $\hat{u}^* : \Omega \rightarrow U$ eine Abbildung, mit folgenden Eigenschaften:

- zu jedem $x \in \Omega$ existiert ein $u_x \in U$, so dass das Maximum in (4.1) angenommen wird
- dieses u_x existiert, wird aber im Allgemeinen nicht eindeutig sein
- $\hat{u}^*(x) = u_x$

4.0.8 Definition

Sei nun zu jedem Anfangswert x $\hat{u}_h^x \in U^{\mathbb{N}_0}$ eine diskrete Kontrollfunktion mit folgender iterativen Vorschrift:

$$\hat{u}_h^x(0) := \hat{u}^*(x) \quad \hat{u}_h^x(ih) = \tilde{u}^*(\Phi_h(hi, x, \hat{u}_h^x)) \quad \text{für } i = 1, 2, \dots \quad (4.2)$$

4.0.9 Satz

Die in (4.2) definierte diskrete Kontrollfunktion u_h^x ist approximativ optimal für das zeitdiskrete optimale Steuerungsproblem. Genauer gilt $\forall x \in \mathbb{R}^d$ die Abschätzung

$$|J_h(x, \hat{u}_h^x) - v_h(x)| \leq C \frac{k^\gamma}{h^{\gamma+1}}$$

für $\gamma = 1$, falls $\delta > L$, $\gamma \in (0, 1)$ beliebig, falls $\delta = L$ und $\gamma = \frac{\delta}{L}$, falls $\delta < L$, und für eine geeignete Konstante $C > 0$.

Beweis: Ist in [13] zu finden.

Kapitel 5

Optimierungsstrategien

Nun können wir uns mit dem Kern der Arbeit - den effizienten Strategien zur iterativen Berechnung der optimalen Wertefunktion - beschäftigen.

Die bisherige Vorgehensweise des iterativen Verfahrens in der Maximierung über U beinhaltete die Diskretisierung der Menge und die Bestimmung des Maximums über Vergleiche. Dieses Verfahren stößt jedoch bei geforderter höherer Genauigkeit schnell an seine Grenzen, da die Berechnung der Werte für viele diskrete Werte $u \in U$ sehr lange dauert. Deswegen werden wir in diesem Kapitel auf Verfahren eingehen, die auf die Diskretisierung der Menge U verzichten. Es werden drei Verfahren (rekursive Suche, Brent-Verfahren, erweitertes Brent-Verfahren) sein, von denen wir das erste - die rekursive Suche - bei den folgenden Tests nur als Genauigkeitskontrolle der anderen Verfahren benutzen werden. Denn die rekursive Suche ist zwar sehr genau, jedoch zu langsam für unsere Ansprüche. Dabei werden die Strategien in diesem Kapitel hinsichtlich der Voraussetzungen, Funktionsweise, sowie des Aufwands betrachtet. Eine praktische Testauswertung folgt im nächsten Kapitel.

5.1 Rekursive Suche

5.1.1 Bedingungen

Zuerst muss der für den jeweiligen Gitterpunkt x zulässige lokale Bereich für den Steuerparameter u gefunden werden. Dies wird durch die Funktion `admit_intervall()` gemacht. Dabei werden zuerst die ursprünglichen Intervallgrenzen a und b überprüft. Sind diese zulässig, so wird das ganze Intervall als zulässig deklariert und die Funktion gibt $[a, b]$ als zulässigen Bereich zurück. Es wird davon ausgegangen, dass der Bereich zusammenhängend ist.

Ist einer der beiden Endpunkte nicht zulässig, so wird mittels der Bisektion der zulässige zweite Endpunkt $c \in [a, b]$ ermittelt und $[c, b]$ bzw. $[a, c]$ zurückgegeben.

Sind beide Endpunkte unzulässig, so unterteilt man den Bereich in 20 äquidistante Abschnitte und evaluiert die Funktionswerte, bis ein zulässiger Punkt c gefunden ist. Ist dies der Fall, so wird mittels der Bisektion ein Intervall $[d, e]$ ermittelt, so dass $c \in [d, e] \subset [a, b]$. Andernfalls wird das Verfahren abgebrochen.

5.1.2 Vorgehensweise

Wir nehmen gemäß Bemerkung 2.1.2 $U = [a, b]$ als kompaktes Intervall an und gehen folgendermaßen vor:

1. Wir wählen ein $n \in \mathbb{N}$, so dass $n_i \geq 3$ und setzen die Rekursionstiefe $i := 0$. Die zu betrachtenden Intervallgrenzen setzen wir auf $a_0 = a$, $b_0 = b$.
2. Wir setzen $\Delta_i = (b_i - a_i)/n_i$. Die daraus resultierende Menge der Werte $U_i = \{a_i, a_i + \Delta_i, a_i + 2\Delta_i, \dots, a_i + n_i\Delta_i = b_i\}$ liefert ein $x_i^* \in U_i$, so dass $F(x_i^*) = \max_{x \in U_i} F(x)$.
3. Falls die Länge des Intervalls $[a_i, b_i]$ kleiner ist als eine vorher bestimmte Schranke ϵ , beenden wir den Algorithmus und geben x_i^* als approximative Maximalstelle aus.

4. Andernfalls gilt für die Intervallgrenzen: Falls $x_i^* = a_i$, setzen wir die Intervallgrenzen der folgenden Rekursion auf $a_{i+1} := x_i^*$, $b_{i+1} := x_i^* + 2\Delta_i$. Analog dazu setzen wir $a_{i+1} := x_i^* - 2\Delta_i$, $b_{i+1} := x_i^*$ falls $x_i^* = b_i$. Liegt die bisherige Maximalstelle nicht am Intervallrand, so setzen wir $a_{i+1} := x_i^* - \Delta_i$, $b_{i+1} := x_i^* + \Delta_i$, erhöhen i auf $i + 1$ und gehen zu 2.

5.1.3 Aufwandsabschätzung

Nun wollen wir den Aufwand dieses Verfahrens abschätzen. Bei einem Startintervall $U = [a, b]$, der geforderten Genauigkeit $\epsilon > 0$ und der Unterteilung $n \geq 3$ folgt

$$\frac{2^i(b-a)}{n^i} \leq \epsilon$$

$$\Rightarrow i \geq R \log_2 \left(\frac{(b-a)}{\epsilon} \right) \quad \text{mit} \quad R = \frac{1}{\log_2(\frac{n}{2})}$$

Da $i \in \mathbb{N}^+$ gilt

$$i \geq \left\lceil R \log_2 \left(\frac{(b-a)}{\epsilon} \right) \right\rceil$$

Daraus ergibt sich für die Konvergenz eine Schranke von:

$$(n+1)i$$

Funktionsevaluationen.

Bei der äquidistanten Diskretisierung von $U = [a, b]$ benötigt man hingegen $(b-a)/2\epsilon$ Teilintervalle, was $((b-a)/2\epsilon) + 1$ Funktionsevaluationen bedeutet.

Ein kleines Beispiel soll nun den Aufwand beider Verfahren vergleichen. Angenommen $U = [0, 1]$ und das Maximum soll mit einer Genauigkeit von $\epsilon = 0.001$ ermittelt werden. Mit einer äquidistanten Diskretisierung von U benötigt man hierfür 501 Funktionsevaluationen. Mit der rekursiven Suche (unter der Annahme, dass sie konvergiert) mit $n = 5$ benötigen wir höchstens $i = \lceil 7.53 \dots \rceil = 8$ Iterationen und somit 48 Funktionsevaluationen. Man sieht, dass wir bereits bei relativ kleiner Genauigkeit mit weniger als 10% der Funktionsauswertungen auskommen.

5.1.1 Bemerkung

- Die beste Wahl von n , um eine Verkleinerung des Intervalls auf $|b_i - a_i| \leq \epsilon$ mit den wenigsten Funktionsevaluationen zu erreichen, hängt direkt von ϵ ab. Wir bestimmen nun das optimale n für $\epsilon = 0.01$ und $\epsilon = 0.00001$.

$\epsilon = 0.01$:

n	i	$(n+1)i$
3	$\lceil 11.358 \dots \rceil = 12$	48
4	$\lceil 6.644 \dots \rceil = 7$	35
5	$\lceil 5.026 \dots \rceil = 6$	36
6	$\lceil 4.192 \dots \rceil = 5$	35
7	$\lceil 3.676 \dots \rceil = 4$	32
8	$\lceil 3.322 \dots \rceil = 4$	36
9	$\lceil 3.062 \dots \rceil = 4$	40
10	$\lceil 2.861 \dots \rceil = 3$	33
11	$\lceil 2.701 \dots \rceil = 3$	36
12	$\lceil 2.571 \dots \rceil = 3$	39

$\epsilon = 0.00001$:

n	i	$(n+1)i$
3	$\lceil 28.394 \dots \rceil = 29$	116
4	$\lceil 16.609 \dots \rceil = 17$	85
5	$\lceil 12.564 \dots \rceil = 13$	78
6	$\lceil 10.479 \dots \rceil = 11$	77
7	$\lceil 9.190 \dots \rceil = 10$	80
8	$\lceil 8.304 \dots \rceil = 9$	81
9	$\lceil 7.654 \dots \rceil = 8$	80
10	$\lceil 7.153 \dots \rceil = 8$	88
11	$\lceil 6.753 \dots \rceil = 7$	84
12	$\lceil 6.425 \dots \rceil = 7$	91

- Wenn man jedoch sicher gehen will, ein Maximum nicht durch grobe Unterteilung des Intervalls zu "übersehen", sollte die Schrittweite feiner gewählt werden.
- Wenn die Funktion F Lipschitz stetig in u ist, so folgt aus $|x_i^* - x^*| \leq \epsilon$ die Abschätzung $|F(x_i^*) - F(x^*)| \leq L\epsilon$. D.h., dass auch das Maximum selbst gut approximiert wird.

5.2 Brent-Verfahren

Brent¹ hat Anfang der 70er Jahre einen Algorithmus [17] entwickelt, welcher das Maximum einer Funktion f , die auf dem Intervall $[a, b]$ definiert ist, approximiert. Dabei wird je nach Situation zwischen der Methode des Goldenen Schnitts und

¹Brent, Richard P., engl. Mathematiker, *1946

der parabolischen Interpolation gewechselt. Wenn die Zielfunktion in der Nähe des Maximums ein parabolisches Verhalten aufweist, wird eine Interpolation zweiter Ordnung durchgeführt, was eine schnellere Konvergenz des Verfahrens zur Folge hat. Tut sie das nicht, kommen wir mit der Methode des Goldenen Schnitts zwar langsam, aber doch stetig dem gesuchten Maximum näher. Das gefundene Maximum ist lokal und muss nicht dem globalen Maximum entsprechen, wenn die Zielfunktion nicht unimodal ist.

5.2.1 Bedingungen

Um das Verfahren von Brent anwenden zu können, müssen wir zuerst ein Tripel von Werten mit folgender Eigenschaft haben:

$$x_1 < x_2 < x_3 \quad \text{mit} \quad x_1, x_2, x_3 \in [a, b]$$

$$f(x_1) < f(x_2) > f(x_3)$$

Dabei gehen wir folgendermassen vor:

1. Als erstes müssen wir für den jeweiligen Gitterpunkt x den zulässigen lokalen Bereich für den Steuerparameter u bestimmen. Dies wird durch die Funktion `admit_intervall()` gemacht, deren Vorgehensweise bereits im Abschnitt 5.1.1 erläutert worden ist.
2. Sodann wird innerhalb des durch `admit_intervall()` erhaltenen zulässigen Intervalls das benötigte Tripel ermittelt. Dies erreichen wir, indem wir das zu betrachtende Intervall in eine vorher festgelegte Anzahl von Schritten aufteilen und das Maximum sowie die beiden benachbarten Werte als die Startwerte für die Methode festlegen. Innerhalb unseres Programms wird diese Aufgabe durch die Funktion `init_value()` erledigt.
Befindet sich der Maximalwert am Rand des Intervalls, wird zusätzlich der letzte Abschnitt mit einer Bisektion geteilt, um ein eventuelles Maximum, welches sich nahe der Intervallgrenze befindet, aufzuspüren.

Stellt sich jedoch heraus, dass trotz dieser Massnahme der Funktionswert an der Intervallgrenze dem maximalen errechneten Wert entspricht, wird dieser Punkt als das Maximum angesehen und es wird auf das Ausführen des eigentlichen Brent-Verfahrens verzichtet. In allen anderen Fällen jedoch wird `brent()` mit dem errechneten Tripel sowie den Intervallgrenzen a und b , welche für gewöhnlich den beiden äusseren Punkten x_1 und x_3 entsprechen, aufgerufen.

5.2.2 Vorgehensweise

Die Methode behält 6 Punkte a , b , u , v , w , und x im Auge. Diese müssen nicht unbedingt verschieden sein. Die Position dieser Punkte wechselt während des Algorithmus, nicht jedoch ihre Bedeutung.

Wir definieren a und b als Intervallgrenzen, in dem das Maximum liegt, x als den bis dahin höchsten errechneten Funktionswert, w als den zweithöchsten und v als den früheren Wert von w . Der letzte Punkt, dessen Funktionswert gerade bestimmt worden ist wird mit u bezeichnet. Dieser besitzt im ersten Schritt noch keinen Wert.

1. Da am Anfang noch keine anderen Punkte vorliegen, verwenden wir die durch `init_value()` übermittelten Werte wie folgt: $a := x_1$, $b := x_3$, $x := x_2$. Falls $f(x_1) \geq f(x_3)$, $w := x_1$, $v := x_3$. Sonst $w := x_3$, $v := x_1$. Eine mögliche Konfiguration zeigt die Abbildung 1.

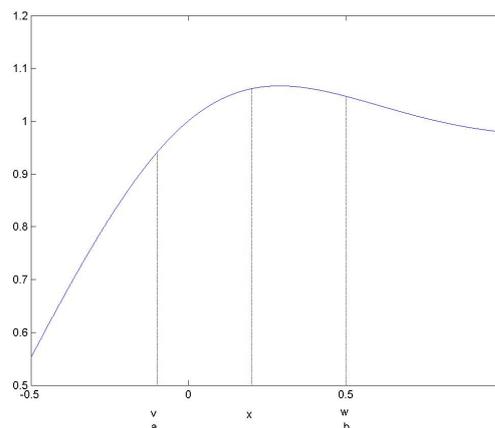


Abbildung 5.1: Mögliche Konfiguration zu Beginn von `brent()`

2. Wenn $|b - a| \leq \epsilon$, dann terminiert der Algorithmus und $f(x)$ wird als das approximierete Maximum zurückgegeben.
3. Andernfalls werden p und q bestimmt, so dass $u = x + \frac{p}{q}$ dem Extrempunkt der Parabel entspricht, welche durch die Punkte $(x, f(x)), (v, f(v))$ und $(w, f(w))$ verläuft. Wenn zwei oder mehr dieser Punkte koinzident sind, liefert $q = 0$.

Die exakten Formeln für p und q lauten:

$$p = \pm[(x - v)^2(f(x) - f(w)) - (x - w)^2(f(x) - f(v))]$$

$$q = \mp 2[(x - v)(f(x) - f(w)) - (x - w)(f(x) - f(v))]$$

4. Sei e der Wert von $\frac{p}{q}$ in der vorletzten Iteration. Wenn $|e| \leq \frac{1}{4}\epsilon$, $q = 0$, $x + \frac{p}{q} \notin (a, b)$ oder $|\frac{p}{q}| \geq \frac{1}{2}|e|$ (da eine Konvergenz angestrebt wird, darf u maximal um diesen Betrag von x entfernt sein), wird die Methode des Goldenen Schnitts ausgeführt. Dies gilt insbesondere für die erste Iteration, da zu diesem Zeitpunkt $e = 0$ ist. Der nächste Wert für u ist dann:

$$u = \begin{cases} (\frac{\sqrt{5}-1}{2})x + (\frac{3-\sqrt{5}}{2})a & : x \geq m \\ (\frac{\sqrt{5}-1}{2})x + (\frac{3+\sqrt{5}}{2})b & : x < m \end{cases}$$

5. Andernfalls wird u mittels der parabolischen Interpolation als $x + \frac{q}{p}$ berechnet, vorausgesetzt die Abstände $|u - x|$, $u - a$ und $b - u$ betragen mindestens $\frac{1}{4}\epsilon$.
6. Abschliessend wird der Funktionswert $f(u)$ berechnet und die Punkte a , b , v , w und x entsprechend ihrer Funktion aktualisiert. Wir gehen zu 2 und eine neue Iteration beginnt.

Man sieht, dass der Algorithmus nie zwei Funktionswerte von Punkten ausrechnet, deren Abstand kleiner als $\frac{1}{4}\epsilon$ ist. Dies bedeutet gleichzeitig, dass der Algorithmus das Maximum gefunden hat, sobald $|b - a| \leq \epsilon$.

Eine typische Konfiguration der Punkte zeigt die Abbildung 5.2.2. Dabei stellt v den in der letzten Iteration gültigen Intervallanfang mit: $|b - v| > \epsilon$. Im der nächsten Iteration wird der neue Punkt u evaluiert und entsprechend des Funktionswertes ($f(u) < f(x)$ und $f(u) > f(v)$) als neuer Intervallanfang a gesetzt. Jetzt gilt $|b - a| < \epsilon$ und das Maximum ist gefunden.

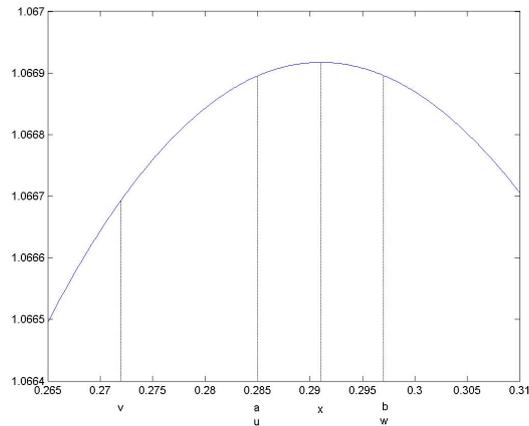


Abbildung 5.2: Typische Konfiguration zum Schluß von *brent()*

5.2.3 Aufwandsabschätzung

Da der Bruch $|\frac{p}{q}|$ nach jeweils zwei Iterationsschritten um einen Faktor ≥ 2 kleiner wird und bei $|e| \leq \epsilon$ der Goldene Schnitt gemacht wird, kommen wir für $[a, b]$ als Anfangsintervall und ϵ als Genauigkeitsschranke zur folgenden Konvergenzaussage:

$$\left(\frac{1}{2}\right)^{\left(\frac{1}{2}x\right)}(b-a) \leq \epsilon$$

Daraus folgt, dass höchstens

$$x \leq \left\lceil 2 \log_2 \left(\frac{b-a}{\epsilon} \right) \right\rceil$$

aufeinanderfolgende Schritte der parabolischen Interpolation vorkommen können. Bleibt die Konvergenzbetrachtung der Methode des Goldenen Schnitts. Ein solcher Schnitt verkleinert das Intervall nicht unbedingt merklich. Wenn z.B. $x = b - \epsilon$ und $f(u) > f(x)$, wird das Intervall $[a, b]$ nur um ϵ verkleinert. Aber zwei Goldene Schnitte müssen das Intervall um mindestens den Faktor $\geq \frac{(1+\sqrt{5})}{2} = 1.618\dots$ verkleinern, was uns zur folgenden Konvergenzaussage bringt:

$$\left(\frac{2}{1+\sqrt{5}}\right)^x (b-a) \leq \epsilon$$

$$\Rightarrow x \leq \lceil K \log_2\left(\frac{b-a}{\epsilon}\right) \rceil \quad \text{mit} \quad K = \frac{1}{\log_2\left[\frac{1+\sqrt{5}}{2}\right]} = 1.44\dots$$

Zusammen betrachtet ergibt sich für die Konvergenz eine Schranke von nicht mehr als:

$$\lceil 2K \left[\log_2\left(\frac{b-a}{\epsilon}\right) \right]^2 \rceil$$

Funktionsevaluationen.

5.2.1 Bemerkung

- Wenn man die obere Schranke für $\text{brent}()$ mit den Beispielzahlen aus 5.1.3 bestimmt, kommt man mit maximal $\lceil 286, 11.. \rceil = 287$ Funktionsevaluationen auf ein Ergebnis, das um fast 50% besser als die Methode der äquidistanten Diskretisierung ist.
- Die Tests im Kapitel 6 werden zeigen, dass die durchschnittliche Zahl der benötigten Funktionsauswertungen weniger als 10% der oberen Schranke beträgt.
- Wenn die Funktion C^2 in der Nähe des Maximums ist, so ist die Konvergenz höher, da ab einem bestimmten Zeitpunkt nur noch parabolische Interpolationen durchgeführt werden.
- Das Verfahren neigt dazu unausgeglichene Punkt-Tripel zu erzeugen, d.h. $|x - a| \ll |b - x|$ oder umgekehrt.

5.3 Erweitertes Brent-Verfahren

Ein Problem bleibt jedoch. Bei der oben angegebenen Vorgehensweise in der Belegung der Punkte x , v und w kommt es zu einer unausgewogenen Verteilung was ein Problem bei der exakten Evaluation des nächsten Funktionswertes darstellt. Es kommt zur numerischen Ungenauigkeit bei der parabolischen Interpolation, da zwei Punkte nah beieinander sind, wogegen sich der dritte relativ weit weg befindet. Die Folge ist eine Parabel, welche nur bedingt den wahren Verlauf der Zielfunktion interpoliert.

Wenn man annimmt, dass die tatsächliche Maximalstelle in der Umgebung $d = |x - u|$ des bereits berechneten bisherigen Extremums x liegt, kann man versuchen das Intervall $[a, b]$ durch gezielte Funktionsberechnungen auf beiden Seiten gleichmäßig zu verkleinern und somit eine schnellere Konvergenz zu erreichen. Das Verfahren von Brent verändert die Intervallgrenze leider nur an einer Seite. Eine Anregung diesen Mißstand zu ändern hat Jack Crenshaw [3] durch seine Erweiterung des Verfahrens von Brent gegeben.

5.3.1 Bedingungen

Um das erweiterte Verfahren von Brent - *erw-b()* - anwenden zu können, müssen wir wie zuvor bei Brent zuerst ein Tripel von Werten mit folgender Eigenschaft haben:

$$x_1 < x_2 < x_3 \quad \text{mit} \quad x_1, x_2, x_3 \in [a, b]$$

$$f(x_1) < f(x_2) > f(x_3)$$

Zur genauen Beschreibung sei auf den Abschnitt 5.2.1 verwiesen. Zudem werden noch zwei weitere Punkte $x_0 < x_1$ und $x_4 > x_3$ mit den dazugehörigen Funktionswerten aufgenommen, so dass aus dem Tripel ein 5-Tupel entsteht.

5.3.2 Vorgehensweise

Ein typischer Schritt in der Methode behält 6 Punkte x_0, x_1, x_2, x_3, x_4 und u im Auge.

x_1 und x_3 entsprechen den Intervallgrenzen, in dem das Maximum liegt, x_2 besitzt den bis dahin höchsten errechneten Funktionswert, x_0 mit $x_0 < x_1$ und x_4 mit $x_4 > x_3$ liegen außerhalb des Intervalls, werden aber zur Stabilisierung des Verfahrens benötigt. u ist der letzte Punkt, dessen Funktionswert bestimmt worden ist. Er ist im ersten Schritt noch nicht definiert.

Eine mögliche Konfiguration zeigt die Abbildung 5.3.2.

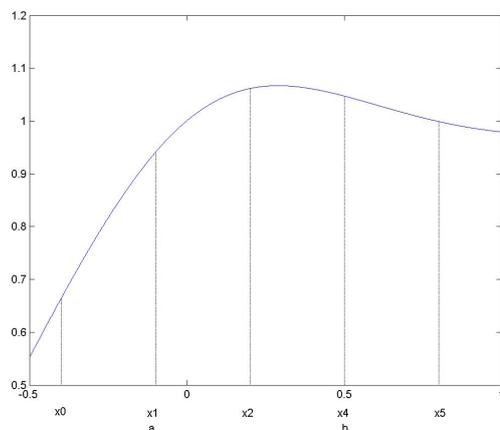


Abbildung 5.3: Mögliche Konfiguration zu Beginn von *erw-b()*

Dabei werden im Verlauf eines Iterationsschrittes folgende Routinen verwendet.

- Die Funktion, welche am Anfang jedes Iterationsschrittes steht ist *para_fit()*. Sie legt eine Parabel durch die inneren 3 Punkte $[x_1, x_2, x_3]$ mittels der schon bei Brent gezeigten Formel. Dabei wird ein neuer Punkt u bestimmt sowie sein Abstand $d = x_2 - u$ vom bisherigen Maximum.

Ist $f(u) > f(x_2)$, wird ein neuer Punkt $u_1 = u + d$ bestimmt sowie sein Funktionswert $f(u_1)$ evaluiert, welcher meistens kleiner als $f(u)$ ist. Daraus ergibt sich das neue innere Tripel $[x_2, u, u_1]$ bzw. $[u_1, u, x_2]$, welches gegenüber dem Tripel bei *brent()* ausbalancierter ist und dessen Intervalllänge radikal reduziert worden ist. Wir nennen diese Funktion *twostep()*, da wir den gleichen Schritt d zweimal in die gleiche Richtung gemacht haben.

Ist $f(u) < f(x_2)$, wird ein neuer Punkt $u_1 = x_2 - d$ bestimmt sowie sein Funktionswert $f(u_1)$ evaluiert, welcher auf der anderen Seite von x_2 liegt als u . Das Maximum wird durch das neue Tripel $[u, x_2, u_1]$ bzw. $[u_1, x_2, u]$ eingeschlossen, wenn $f(u_1) < f(x_2)$. Diesmal haben wir den Schritt nicht in der gleichen Richtung wiederholt, sondern ihn in der entgegengesetzten ausgeführt. Deswegen nennen wir diese Funktion *flip()*. Wie bereits bei *twostep()* haben wir ein ausbalanciertes Tripel mit stark verkleinerter Intervalllänge.

Anschliessend wird das Array der 5 Punkte $[x_0, x_1, x_2, x_3, x_4]$ entsprechend der Vorgabe aktualisiert.

- Die Funktion `wide_fit()` legt eine Parabel durch die äußeren 3 Punkte $[x_0, x_2, x_4]$ mittels der bereits bekannten Formel. Der neue Punkt u wird nur betrachtet, wenn er innerhalb des inneren Tripels $[x_1, x_2, x_3]$ liegt. Ist es der Fall, wird $f(u)$ evaluiert und das 5 Punkte Array aktualisiert. Ansonsten wird er verworfen und der Iterationsschritt ist beendet.
- Die Funktion `line_fit()` legt jeweils eine Gerade durch die zwei rechten bzw. linken Punkte des Arrays und berechnet den Schnittpunkt. Voraussetzung dafür ist eine bestimmte Form der Funktionswerte des Arrays, nämlich eine Dach-Form, d.h. $f(x_0) < f(x_1) < f(x_2) > f(x_3) > f(x_4)$. Nur wenn die gegeben ist, macht es Sinn diese Funktion aufzurufen. Dadurch sollen Funktionen, deren Verhalten in der Nähe des Maximums nicht parabolisch ist, (z.B. die Betragsfunktion) schneller approximiert werden. Der Schnittpunkt der zwei Geraden bildet dann den neuen Punkt u , der wiederum nur betrachtet wird, wenn er innerhalb des inneren Tripel liegt und dann in das 5 Punkte Array eingefügt wird. Nach dieser Funktion ist der Iterationsschritt auf jeden Fall zu Ende.

5.3.1 Bemerkung

Da die Anbindung von `line_fit()` bei den im Kapitel 6 betrachteten Funktionen schlechtere Ergebnisse lieferte als ohne, haben wir diese Funktion aus dem erweiterten Brent-Verfahren herausgenommen und nur die Funktionen, `para_fit()`, `twostep()`, `flip()` und `wide_fit()` verwendet. Sie ist als herauskommentierter Teil des Quellcodes jedoch jederzeit für andere Beispiele aktivierbar.

5.3.3 Abbruchkriterien

Bei diesem Verfahren haben wir 2 Konvergenzkriterien, welche kombiniert als Abbruchkriterium verwendet werden.

- $[x_0, x_2] < \epsilon$
- $|y_{max} - f(x_{max})| < \epsilon^2$, d.h. der erwartete Funktionswert y_{max} ist sehr nahe am tatsächlichen Funktionswert $f(x_{max})$

Da sowohl bei *para_fit()* als auch bei *wide_fit()* eine parabolische Interpolation, sprich eine Interpolation vom Grad 2, durchgeführt wird, reicht es, wenn für das zweite Konvergenzkriterium eine Schranke von ϵ^2 angenommen wird. Beide Abbruchkriterien werden durch eine oder-Verknüpfung verbunden.

Kapitel 6

Numerische Tests

Nachfolgend werden die theoretischen Schranken mittels praktischer Beispiele überprüft. Dabei werden wir das erweiterte Brent-Verfahren, das ursprüngliche Brent-Verfahren sowie die rekursive Suche der bisher verwendeten äquidistanten Diskretisierung gegenüberstellen und diese nach der Zahl der benötigten Funktionsevaluationen, dem Durchschnitt für eine Maximierung sowie dem errechneten Maximum und Minimum vergleichen.

6.1 Bezeichnungen

Da wir in diesem Kapitel einige Abkürzungen für Kurzformen von Verfahren und ihre Eigenschaften, unter welchen sie ausgewertet werden, benötigen, sollen sie an dieser Stelle kurz eingeführt werden. Sie sollen die Übersichtlichkeit der Tabellen verbessern.

Als erstes wollen wir die Bezeichnungen für die Verfahren angeben:

- diskr* : Äquidistante Diskretisierung
- brent* : Brent-Verfahren
- rekur* : Rekursive Suche
- erw – b* : Das erweiterte Brent-Verfahren

Sie werden nach folgenden Eigenschaften ausgewertet:

- Intervall : Anzahl der Teilintervalle, in die ein Intervall eingeteilt wird
 -bei *brent* und *erw-b* bezüglich der Schritte in *init_value()*
 -bei *diskr* in Bezug auf das Ausgangsintervall
 -bei *rekur* bezüglich jeder Rekursionsstufe
- Gesamt : Gesamtzahl der Funktionsevaluationen über alle Gitterpunkte
- Iter : Anzahl der Iterationen für das gesamte Gitter
- Min : Das Minimum über alle Funktionswerte
- Max : Das Maximum über alle Funktionswerte
- Schnitt : Durchschnittliche Zahl von Funktionsevaluationen
 für eine Maximierung eines Gitterpunktes
- Fehler : Durchschnittsdifferenz zwischen den ermittelten
 und den exakten Funktionswerten

Die verwendeten Parameter:

- h : Schrittweite
- δ : Diskontrate
- ϵ : Genauigkeitsschranke
- $n[0]$: Anzahl der Unterteilungen in $x[0]$ -Richtung
- $n[1]$: Anzahl der Unterteilungen in $x[1]$ -Richtung
- u : Bereich, in dem die Steuervariable ausgewertet wird

6.1.1 Bemerkung

- n Unterteilungen bedeuten $n + 1$ Funktionswerte in eine Richtung. Dies ist bei der Ermittlung der durchschnittlichen Zahl von Funktionsevaluationen für eine Maximierung zu beachten.
- Gleiches trifft bei der Tabellenspalte mit der Bezeichnung Intervall zu. Auch hier sind bei n Intervallen jedesmal $n + 1$ Funktionswerte zu bestimmen.
- Zur Abkürzung der Bildunterschriften werden nur das benutzte Verfahren sowie in [] der Wert aus der Spalte Intervall benutzt. Dies sollte die problemlose Zuordnung der Graphik zur Tabellenzeile gewährleisten.

- Aus Gründen der Übersichtlichkeit werden nicht alle in den Tabellen berechneten Funktionen visualisiert. Die dazu notwendigen Datensätze sind aber selbstverständlich auf der beiliegenden CD-ROM im Ordner `\Arbeit\Daten` enthalten.

6.2 Einfaches Modell

Als Einstieg soll uns ein einfaches Modell dienen um zu sehen, ob bereits solche Systeme effektiver mit den neuen Optimierungsstrategien zu maximieren sind. Das zeitkontinuierliche optimale Steuerungssystem

$$\begin{aligned}\dot{x}_1(t) &= x_1 + ux_2 \\ \dot{x}_2(t) &= -x_2\end{aligned}$$

und die dazugehörige Ertragsfunktion

$$g(x_1, x_2, u) = -1.0 + x_1 - 0.3u$$

bilden ein relativ einfach lösbares System.

Die bei der Auswertung verwendeten Parameter

h	δ	ϵ	$n[0]$	$n[1]$	u
0.05	0.5	0.0001	50	50	[0;1]

liefern für unsere Verfahren folgende Ergebnisse:

Verfahren	Intervall	Gesamt	Iter	Min	Max	Schnitt
<i>brent</i>	3	4503521	247	-1.9977	-1.2040	7.01
<i>brent</i>	6	6430839	247	-1.9977	-1.2040	10.01
<i>diskr</i>	10	7066917	247	-1.9977	-1.2040	11.00
<i>rekur</i>	6	41759055	247	-1.9977	-1.2040	65.00
<i>rekur</i>	15	52680654	247	-1.9977	-1.2040	82.00
<i>erw - b</i>	3	4504808	247	-1.9977	-1.2040	7.01
<i>erw - b</i>	6	6431612	247	-1.9977	-1.2040	10.01

Bei diesem Beispiel ist die Zahl der Funktionsevaluationen bei *brent* und *erw-b* fast gleich und alle Verfahren kommen auf die gleichen Werte für das Minimum und das Maximum der Funktion. Die nachfolgenden Abbildungen zeigen die Wertefunktion sowie die optimale Steuerung, welche durch *brent*[3] ermittelt worden sind. Der Sprung in der optimalen Steuerung findet sich als ein Knick in der Wertefunktion des Modells wieder. Die Funktion ist somit entlang dieser Linie - genannt Skiba-Linie - nicht differenzierbar.

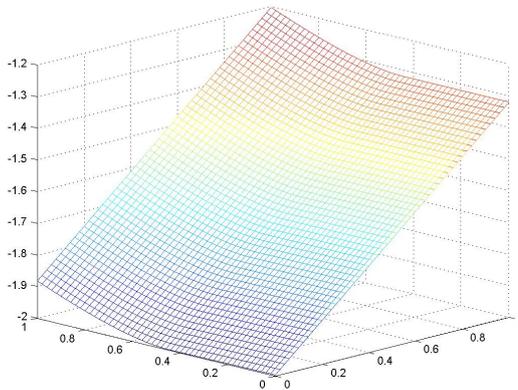


Abbildung 6.1: Wertefunktion
für das einfache Modell

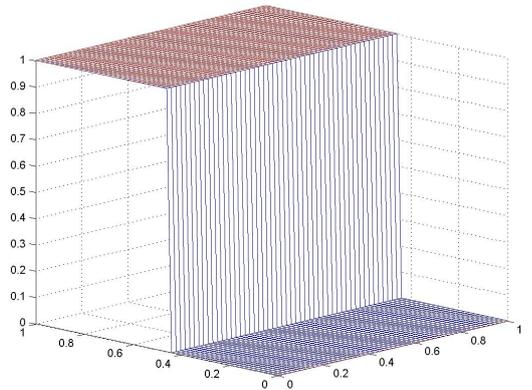
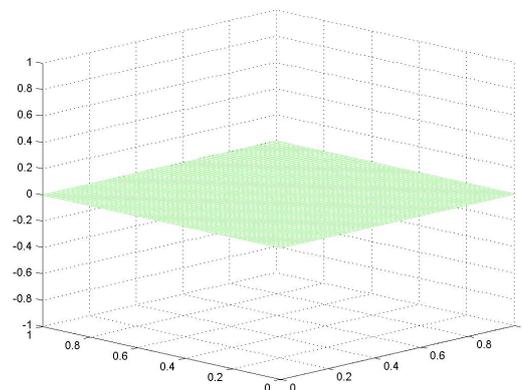


Abbildung 6.2: Optimale Steuerung
für das einfache Modell

Jetzt vergleichen wir den Unterschied zwischen den einzelnen Funktionswerten der Wertefunktionen von *brent*[3] sowie *diskr*[10]. Wie vermutet ist das zugrundeliegende Modell zu einfach gewesen, um einen Vorteil durch *brent* zu haben. Da die optimale Steuerung jeweils den Rändern des Definitionsbereiches für die Steuervariable u entspricht, ist es unerheblich, wieviel Zwischenwerte innerhalb des Intervalls ausgewertet wurden. Daher ist es nicht überraschend, dass $diskr[10](i) - brent[3](i) = 0 \quad \forall i = 0, \dots, N$, was auf folgender Abbildung zu sehen ist. Um die Stärken von *brent* zu zeigen bedarf es eines komplizierteren Modells.

Abbildung 6.3: $diskr[10] - brent[3]$

6.3 Investitionsmodell

Das zweite Beispiel könnte bereits unseren Anforderungen entsprechen. Auch hier wenden wir auf das Modell unsere Optimierungsstrategien an und untersuchen anschliessend die Ergebnisse hinsichtlich der errechneten Funktionswerte und der Zahl der Funktionsevaluationen, die zu diesen geführt haben. Doch zuerst werden wir das Beispiel, welches eine Modellierung der Kapitalströme darstellt, näher vorstellen. Es stammt aus [12].

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

Dabei sollen wir den diskontierten cash flow, der mittels der Ertragsfunktion

$$g(x_1, x_2, u) = k_1 \sqrt{x_1} - \frac{x_1}{1+k_2 x_1^4} - c_1 x_2 - \frac{c_2}{2} x_2^2 - \frac{\alpha}{2} u^2$$

gemessen wird, maximieren. Wir setzen: $\sigma = 0.25$, $k_1 = 2$, $k_2 = 0.0117$, $c_1 = 0.75$, $c_2 = 2.5$ und $\alpha = 12$.

Die übrigen Parameter sowie die resultierenden Ergebnisse sind den folgenden Tabellen zu entnehmen.

h	δ	ϵ	$n[0]$	$n[1]$	u
0.05	0.04	0.001	50	50	[-1;1]

Verfahren	Intervall	Gesamt	Iter	Min	Max	Schnitt
brent	3	58902691	1836	12.962	30.551	12.33
brent	6	69262275	1837	12.961	30.551	14.49
diskr	10	52844517	1847	12.835	30.470	11.00
diskr	100	481793634	1834	12.958	30.546	101.00
rekur	6	247397430	1834	12.960	30.550	51.86
erw-b	3	56519780	1838	12.958	30.559	11.82
erw-b	6	68899431	1834	12.959	30.548	14.44

Zwei Ergebnisse sind ersichtlich. Bei einer kleinen Anzahl an Unterteilungen bekommen wir bei *diskr* schnell Ergebnisse, jedoch erreichen sie nicht die Güte von *brent*. Bei hoher Unterteilung hat *diskr* einen wesentlich grösseren Aufwand als die anderen vorgestellten Strategien. Die Ergebnisse kommen näher an das Optimum heran, aber auch nicht ganz. Der Unterschied zwischen *erw-b* und *brent* ist recht klein. Eine genauere Aussage darüber, welches Verfahren Vorteile bei diesem Beispiel hat, kann erst bei der Betrachtung der Abbildungen 6.8 und 6.9 getroffen werden.

Die folgenden Abbildungen zeigen die mittels *brent[6]* errechnete Wertefunktion sowie die dazugehörige optimale Steuerung. Dabei kann man feststellen, dass das optimal gesteuerte System zwei stabile optimale Gleichgewichte besitzt, deren Einzugsbereiche durch eine Kurve getrennt werden, welche in der Wertefunktion einen Knick darstellt, also nicht differenzierbar ist. Solche Kurven werden Skiba-Kurven genannt.

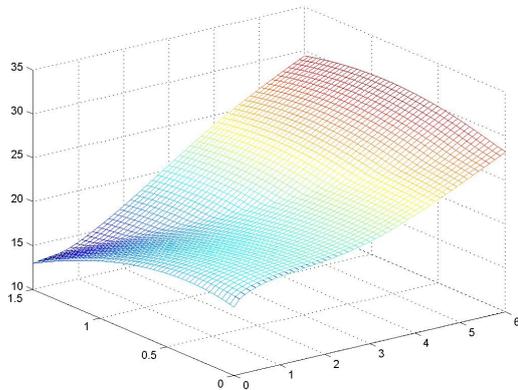


Abbildung 6.4: Wertefunktion
für das Investitionsmodell

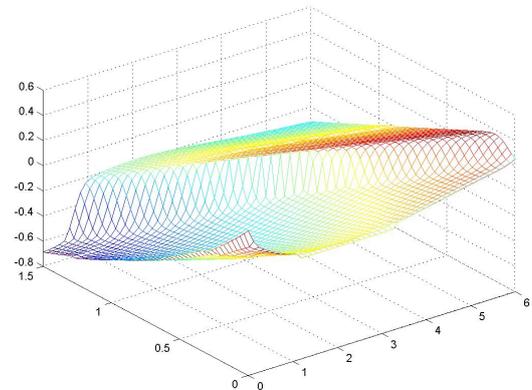


Abbildung 6.5: Optimale Steuerung
für das Investitionsmodell

Mit Hilfe der folgenden Abbildungen können wir nun näher auf folgende Fragen eingehen.

Wo differieren nun die einzelnen Lösungen des Modells? Sind es bestimmte Stellen an denen der Unterschied besonders gross ist oder ist die eine Wertefunktion gleichmässig schlechter als die andere?

Die erste Abbildung bezieht sich auf den Unterschied zwischen der Diskretisierung mittels 10 Abschnitte und $brent[6]$. Wie man deutlich erkennen kann, ist er ungleichmässig und relativ hoch, insbesondere entlang des Knicks.

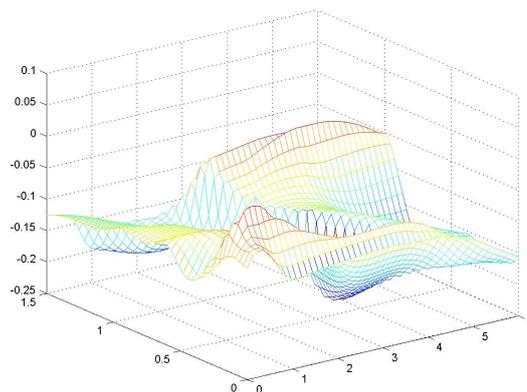


Abbildung 6.6: $diskr[10] - brent[6]$

Die zweite zeigt den Unterschied zwischen der Unterteilung in 100 Abschnitte und *brent*[6]. Das Ergebnis ist erwartungsgemäß deutlich besser als in der vorherigen Abbildung. Jedoch ist der Rechenaufwand so gross, dass eine andere Optimierungsstrategie vorzuziehen wäre. Die einzelnen Spitzen entlang der Skiba-Kurve beruhen auf einem leicht voneinander abweichend ermittelten Verlauf des Knicks innerhalb der Wertefunktionen. Während bei *brent* durch eben diese Punkte die Skiba-Kurve verläuft, befinden sich diese bei *diskr* schon jenseits der Linie, also bei Punkten mit aufsteigenden Funktionswerten.

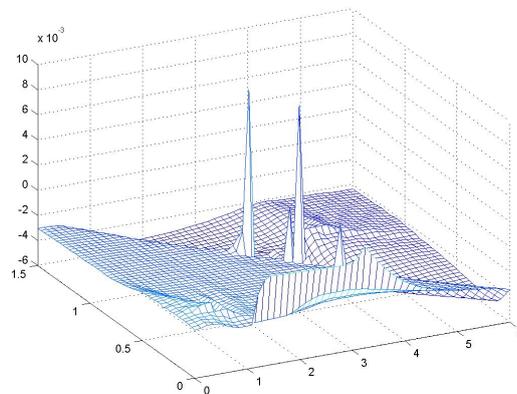
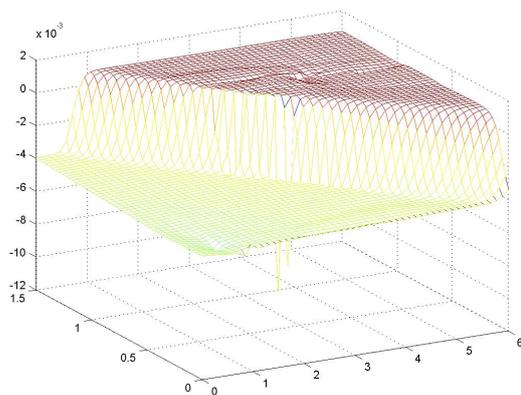
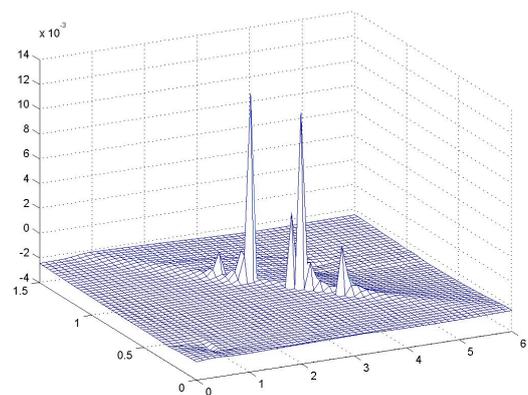


Abbildung 6.7: *diskr*[100] – *brent*[6]

Und wie verhalten sich die zwei Arten des Brent-Verfahrens zueinander? Nachfolgend zeigt die Abbildung die Differenz der Wertefunktionen von *brent* und *erw-b* bei 3 respektive 6 Intervallen. Bei *erw-b*[3] – *brent*[3] kann die Abbildung in einen Bereich vom Anfang des Auswertungsbereichs bis zur Skibalinie und einen Bereich darüber eingeteilt werden. Während im ersten Bereich das Ergebnis von *brent* gleichmäßig um ca. 0.004 besser als bei *erw-b* ist, so drehen sich im zweiten Bereich die Rollen um und *erw-b* liefert die höheren Werte. Die Differenz liegt jedoch nur in der Größenordnung von $\epsilon = 0.001$. Bei *erw-b*[6] – *brent*[6] nähern sich die Ergebnisse der beiden Verfahren noch mehr an. Bis auf ein paar Spitzen entlang der Skibalinie sind die von *brent* ermittelten Werte um 0.002 besser als die von *erw-b*. Die Gründe für die Spitzen sind gleich denen in der Abbildung 6.7. Der Knick verläuft in Wirk-

lichkeit genau zwischen den beiden Gitterpunkten und die Verfahren müssen sich für einen entscheiden. Ein feineres Gitter würde dieses Problem beheben.

Abbildung 6.8: $erw-b[3] - brent[3]$ Abbildung 6.9: $erw-b[6] - brent[6]$

6.4 Makroökonomisches Modell

Das zeitkontinuierliche optimale Steuerungssystem

$$\begin{aligned}\dot{x}_1(t) &= u - 0.55x_1 + \frac{x_1^2}{1+x_1^2} \\ \dot{x}_2(t) &= 0\end{aligned}$$

und die dazugehörige Ertragsfunktion

$$g(x_1, u) = 2\sqrt{u} - \frac{\lambda}{2}x_1^2 \quad \lambda \geq 0$$

bilden das makroökonomische Modell von W. Brock und D. Starret, welches in [13] zu finden ist. Man kann es folgendermaßen interpretieren: Wir nehmen an, dass ein Unternehmen ein Werk in der Nähe eines Sees betreibt und durch die Produktion phosphathaltiges Abwasser entsteht, das in den See eingeleitet wird. Die Variable x_1 beschreibt den Phosphatgehalt des Sees und die Gleichung

$$\dot{x}(t) = 0.55x + \frac{x^2}{1+x^2}$$

die Selbstreinigungskraft des Sees, d.h. sie gibt an, wie schnell das Phosphat abgebaut werden kann. Die Kontrollvariable u beschreibt die Höhe der Produktion des Werks. Der Einfachheit halber ist sie so normiert, dass die Menge der Phosphateinleitung gerade gleich u ist.

Die Ertragsfunktion besteht hier aus zwei Komponenten. Einerseits geht die Produktion positiv in die Gleichung ein. Andererseits hat die Verschmutzung des Sees negative Auswirkungen auf den Gewinn. Beide Komponenten werden durch Faktoren je nach Bedarf gewichtet.

Dieses Testbeispiel ist als ein eindimensionales Modell konzipiert worden. Deswegen haben wir für unsere Berechnungen die zweite Variable $x_2 = 0$ gesetzt. Die verwendeten Parameterwerte

h	δ	ϵ	$n[0]$	$n[1]$	u	λ
0.05	0.1	0.001	50	50	[0;1]	0.8

liefern folgende Ergebnisse:

Verfahren	Intervall	Gesamt	Iter	Min	Max	Schnitt
brent	3	23384364	638	-6.504	5.909	14.09
brent	6	23475148	546	-6.458	5.871	16.53
brent	25	47564343	546	-6.458	5.871	33.49
diskr	10	14048001	491	-6.427	5.679	11.00
diskr	20	29167614	534	-6.460	5.823	21.00
diskr	100	142909344	544	-6.457	5.863	101.00
rekur	6	72705906	546	-6.458	5.871	51.19
erw-b	3	29261607	642	-6.506	5.910	17.52
erw-b	6	33224103	543	-6.456	5.867	23.52
erw-b	25	48401244	546	-6.458	5.871	34.08

Es ist zu beobachten, dass *brent* bei geringerem Aufwand pro Maximierung - 16.53 bei *brent*[6] im Vergleich zu 23.52 bei *erw-b*[6] und 21.00 bei *diskr*[20] - bessere Ergebnisse liefert. Sie sind vergleichbar mit *rekur*, welches jedoch mit 51.19 Funktionsevaluationen pro Maximierung viel langsamer ist. Ebenso sieht man, dass die

Werte in der Spalte Intervall nicht zu klein sein dürfen. Sowohl *brent*[3] als auch *diskr*[10] und *erw-b*[3] liefern Werte, die durch eine feinere Einteilung nicht bestätigt werden konnten. Die durch *brent*[6] erzeugte Wertefunktion bzw. optimale Steuerung zeigen folgende Abbildungen.

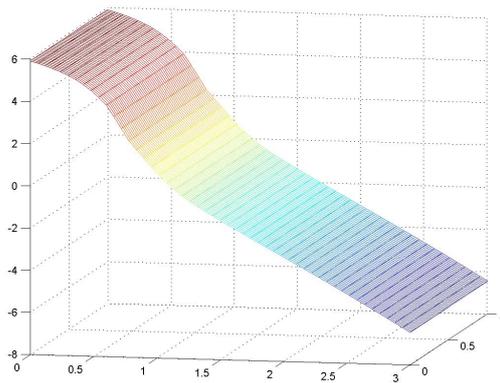


Abbildung 6.10: Wertefunktion für das makroökonomische Modell

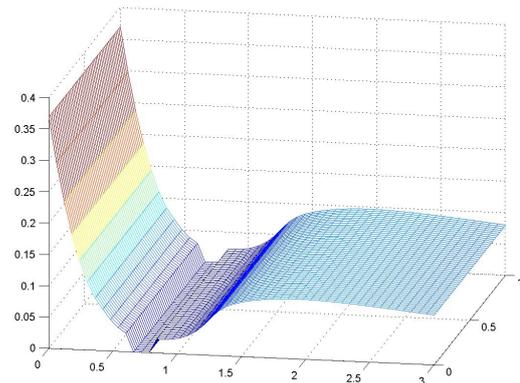


Abbildung 6.11: Optimale Steuerung für das makroökonomische Modell

Wie sehen nun die Unterschiede in den Wertefunktionen im Einzelnen aus? Da wir speziell *brent* mit dem bisher angewandten Verfahren *diskr* vergleichen wollen, haben wir in der folgenden Abbildung die Wertefunktionendifferenz *diskr*[100] – *brent*[25] dargestellt.

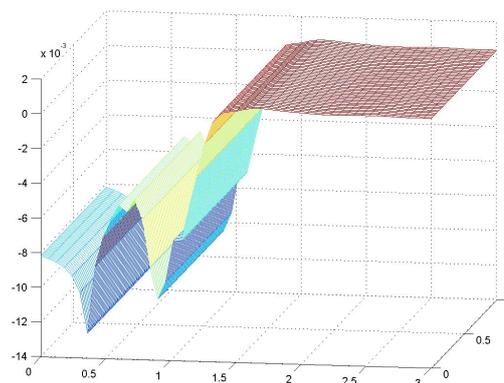


Abbildung 6.12: $diskr[100] - brent[25]$

Die Differenz der beiden auf der parabolischen Interpolation beruhenden Verfahren liefert folgendes Ergebnis.

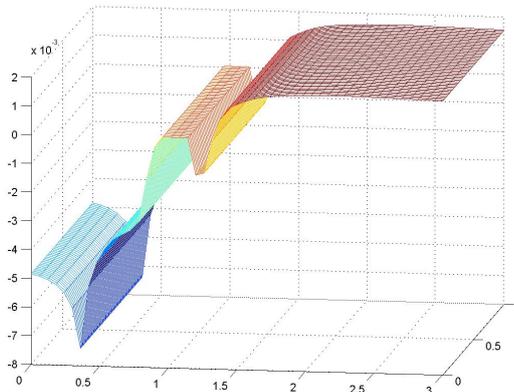


Abbildung 6.13: $erw-b[6] - brent[6]$

Da wir die zweite Dimension x_2 in unserem Beispiel nicht besetzt haben und $\lambda \geq 0$ ein relevanter Parameter innerhalb des Modells ist, setzen wir nun $x_2 = \lambda$ und werten ihn im Bereich $[0; 1]$ aus. Die sich daraus ergebende Ertragsfunktion lautet nun:

$$g(x_1, x_2, u) = 2\sqrt{u} - \frac{x_2}{2}x_1^2$$

Diese Massnahme ändert die Ergebnisse nicht; sie entsprechen jetzt Linien entlang des jeweiligen Wertes für $x_2 = \lambda$. Für die weiteren Tests des Modells werden wir die gleichen Parameterwerte wie oben verwenden. Es ergeben sich folgende Werte:

Verfahren	Intervall	Gesamt	Iter	Min	Max	Schnitt
brent	6	32746375	874	-9.378	18.077	14.40
brent	10	41007194	874	-9.297	18.077	18.03
brent	25	73166972	874	-9.257	18.077	32.18

Verfahren	Intervall	Gesamt	Iter	Min	Max	Schnitt
diskr	10	24548238	858	-9.787	18.019	11.00
diskr	13	31170384	856	-9.589	18.010	14.00
diskr	20	47738754	874	-9.452	18.077	21.00
diskr	100	229600674	874	-9.262	18.077	101.00
rekur	9	118655988	874	-9.257	18.077	52.19
erw-b	6	38130616	874	-9.378	18.077	16.77
erw-b	10	43497367	874	-9.298	18.077	19.13
erw-b	25	72537287	874	-9.257	18.077	31.90

Wir beobachten auch hier, dass *rekur* zwar die besten Werte liefert, jedoch der Aufwand sehr gross ist. Die äquidistante Diskretisierung kommt an *brent*, trotz der Steigerung der Intervalle von 10 auf 20, nicht heran. Erst die Erhöhung auf 100 Intervalle kommt dem Ergebnis von *brent*[25] nahe. Jedoch mit dem 3-fachen Aufwand. *erw-b* und *brent* sind relativ ähnlich im Ergebnis. Benötigt anfänglich *brent* - 6 Intervalle - im Durchschnitt zwei Funktionsevaluation weniger pro Maximierung, holt *erw-b* mit der Erhöhung der Intervallzahl - auf schliesslich 25 - auf und benötigt zum Schluss sogar weniger Funktionsberechnungen als *brent*. Dies lässt vermuten, dass *erw-b* bei kleineren Ausgangsintervallen effizienter ist als *brent*.

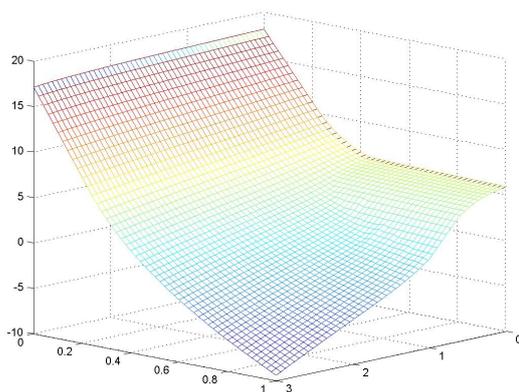


Abbildung 6.14: Wertefunktion für das makroökonomische Modell

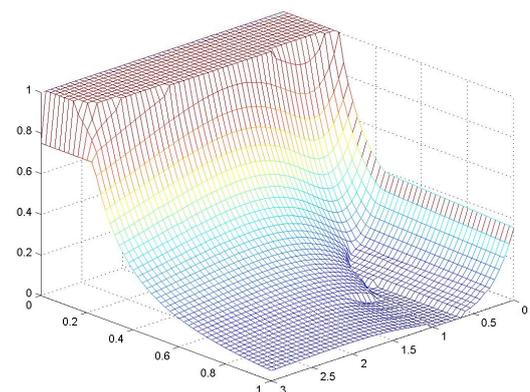


Abbildung 6.15: Optimale Steuerung für das makroökonomische Modell

Man kann sagen, dass unter Berücksichtigung des Aufwands wiederum *brent* die besten Resultate in diesem Beispiel liefert. Dies zeigen auch die nachfolgenden Abbildungen, welche die Wertedifferenz zwischen *diskr*[100] und *brent*[25]

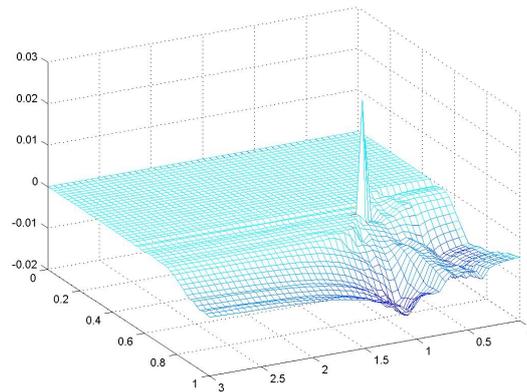


Abbildung 6.16: $diskr[100] - brent[25]$

sowie *erw-b* und *brent* bei 6 bzw. 25 Intervallen darstellen.

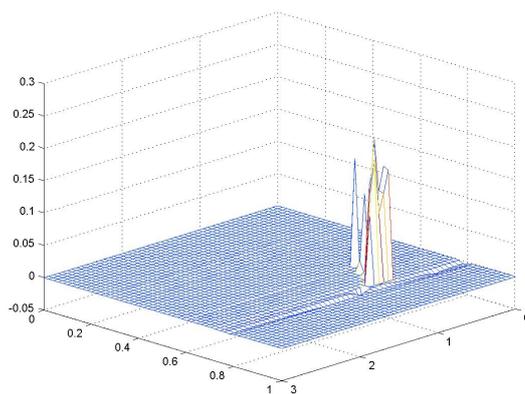


Abbildung 6.17: $erw-b[6] - brent[6]$

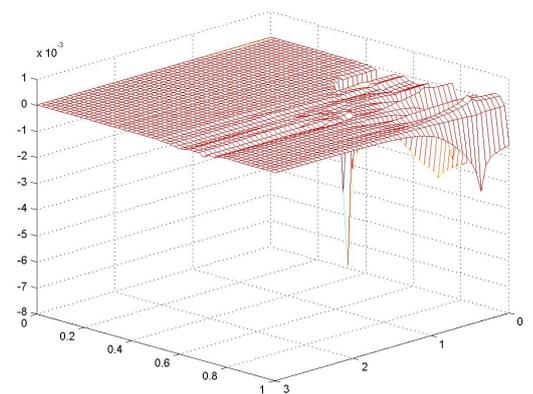


Abbildung 6.18: $erw-b[25] - brent[25]$

6.5 Ökonomisches Wachstumsmodell

Die Performance von *brent* beim Ökonomischen Wachstumsmodell soll nun untersucht werden. Dieses zeitdiskrete optimale Steuerungssystem - in abgewandelter Form entnommen aus [14] - ist gegeben durch

$$\begin{aligned}x_1(t+1) &= Ae^{x_2}x_1^\alpha - u \\x_2(t+1) &= \rho x_2\end{aligned}$$

und die dazugehörige Ertragsfunktion

$$g(x_1, x_2, u) = \ln u$$

wobei $\rho, \alpha, A \in \mathbb{R}^+$. In dieser Arbeit betrachten wir die Werte $\rho = 0.9$, $\alpha = 0.34$, $A = 5$ und $\beta = 1 - \delta h = 0.95$. Die exakte Lösung für dieses Modell ist bekannt und lautet:

$$V(x) = B + C \ln x_1 + D x_2$$

mit

$$B = \frac{\ln((1-\beta\alpha)A) + \frac{\beta\alpha}{1-\beta\alpha} \ln(\beta\alpha A)}{1-\beta}, \quad C = \frac{\alpha}{1-\alpha\beta}, \quad D = \frac{1}{(1-\alpha\beta)(1-\rho\beta)}$$

Die optimale Steuerung für dieses Modell erhält man durch:

$$u_{opt}(x) = (1 - \alpha\beta)Ae^{x_2}x_1^\alpha$$

Sie helfen uns dabei die Ergebnisse im Hinblick auf ihre Differenz zur exakten Lösung zu untersuchen. Die Parameter

h	δ	ϵ	$n[0]$	$n[1]$	u
1	0.05	0.00001	50	50	[0.1;13.5]

und die generierten Ergebnisse sind den Tabellen zu entnehmen.

Verfahren	Intervall	Gesamt	Iter	Min	Max	Schnitt	Fehler
exakt				23.7298	34.1921		0.00
brent	3	19928041	153	23.72397	34.18973	50.07	0.0034
brent	6	21399109	156	23.72398	34.18973	52.74	0.0034
brent	30	29318704	149	23.72397	34.18972	75.65	0.0034
diskr	10	5321646	186	21.55441	34.01367	11.00	0.2649
diskr	50	19367046	146	23.67337	34.18219	51.00	0.0156
diskr	100	35201934	134	23.71483	34.18543	101.00	0.0102
rekur	6	46752822	153	23.72397	34.18973	117.48	0.0034
erw-b	3	25502971	157	23.72398	34.18973	64.45	0.0034
erw-b	6	27363748	158	23.72398	34.18973	66.58	0.0034
erw-b	30	33907834	152	23.72399	34.18973	85.76	0.0033

Der Fehler ist mit Hilfe der Formel:

$$Fehler = \frac{\sum_{i=0}^N |exakt(i) - Verfahren(i)|}{N}$$

ermittelt worden. Auch hier hat *brent* Vorteile gegenüber den anderen Optimierungsverfahren. Zudem ist sichtbar, dass bei einem großen Anfangsintervall $-[0.1; 13.5]$ - und der hohen Genauigkeit $-\epsilon = 0.00001$ - wie in diesem Fall viele Funktionsauswertungen pro Maximierung unumgänglich sind. Auch hier nähert sich die durchschnittliche Zahl der Funktionsberechnungen pro Maximierung von *erw-b* mit der Verfeinerung des Intervalls immer näher an die von *brent* an. War die Vermutung aus dem vorhergehenden Beispiel etwa begründet und *erw-b* wird mit der Verfeinerung des Ausgangsintervalls im Vergleich zu *brent* immer effizienter? Die Antwort liegt in der maximalen Zahl an Funktionsevaluationen pro Maximierung. Während bei *brent* diese stets < 100 ist, beträgt sie bei *erw-b*[3] 3744 bei *erw-b*[6] 2254 und bei *erw-b*[30] immer noch 755. Dies bedeutet wiederum, dass das beidseitige Eingrenzen des Maximums, wie es im Kapitel 5.3 beschrieben worden ist, in der Praxis nicht garantierbar ist. Es folgen die mittels *brent*[6] ermittelte Wertefunktion bzw. optimale Steuerung.

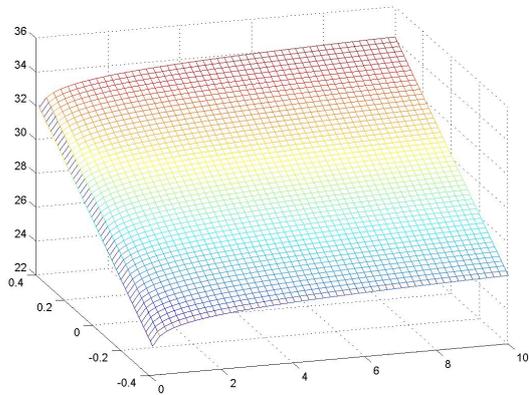


Abbildung 6.19: Wertefunktion für das ökonomische Wachstumsmodell

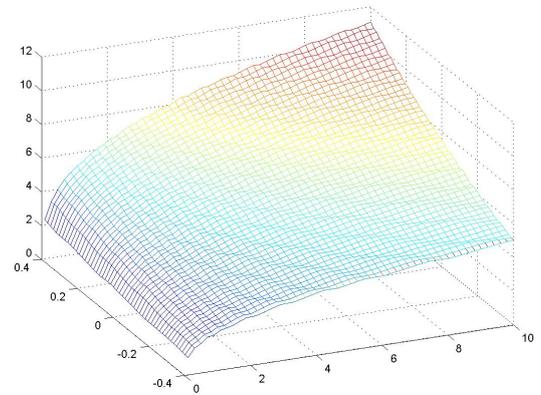


Abbildung 6.20: Optimale Steuerung für das ökonomische Wachstumsmodell

Da die exakte Lösung dieses Beispiels bekannt ist, können wir unsere Ergebnisse direkt damit vergleichen. Dabei werden wir sowohl auf die Differenz der Wertfunktionen, als auch auf die der optimalen Steuerungen eingehen.

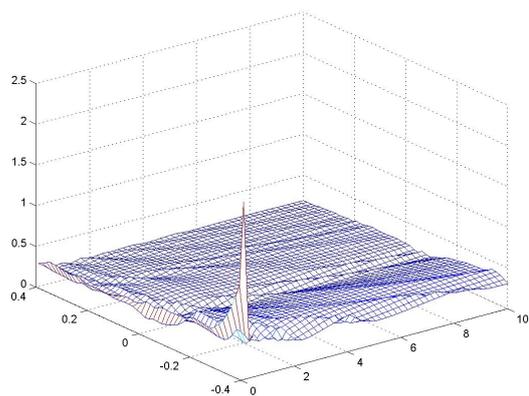


Abbildung 6.21: $exakt - disk[10]$

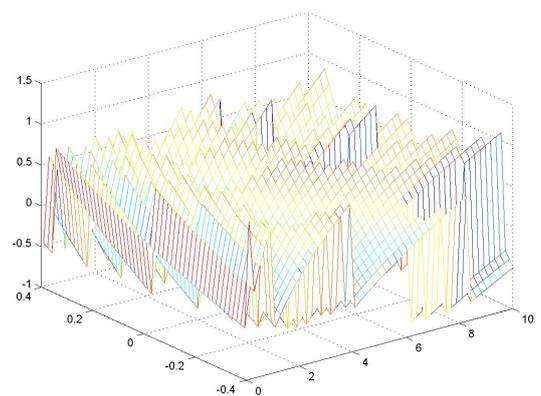
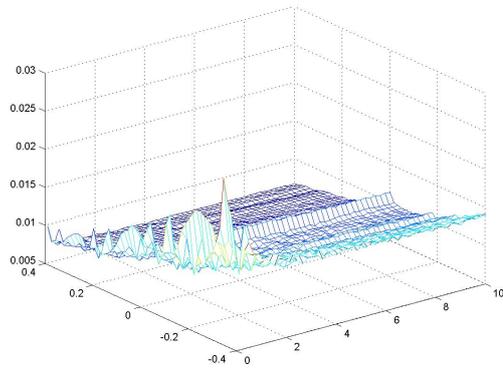
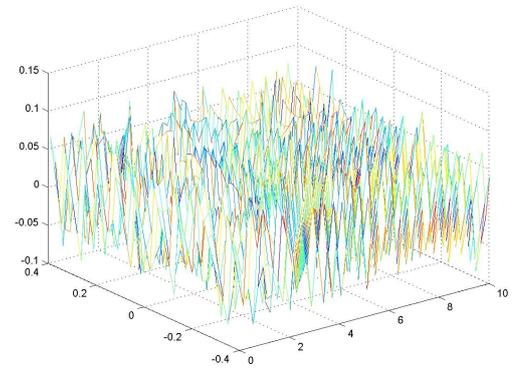
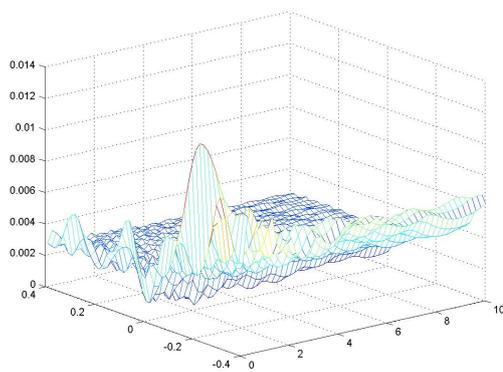
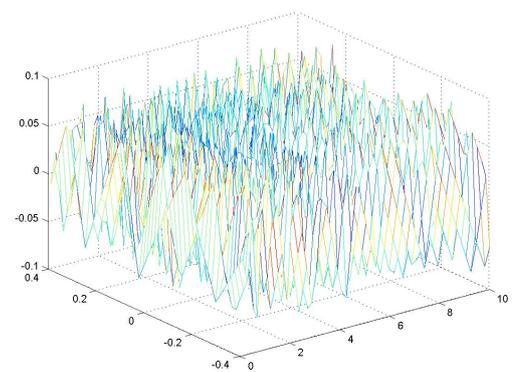
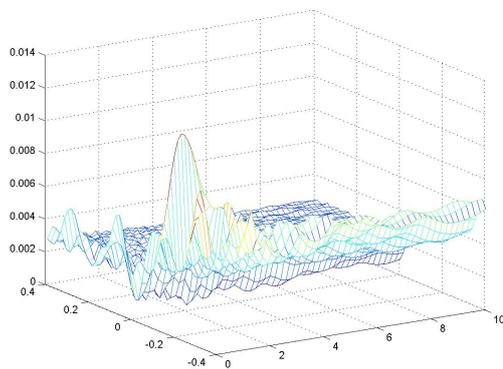
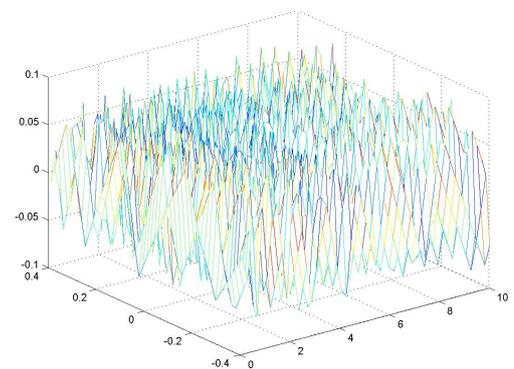
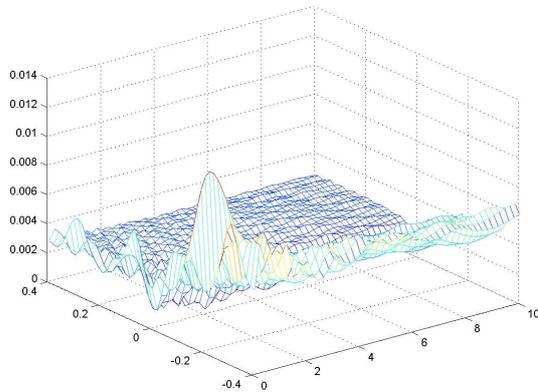
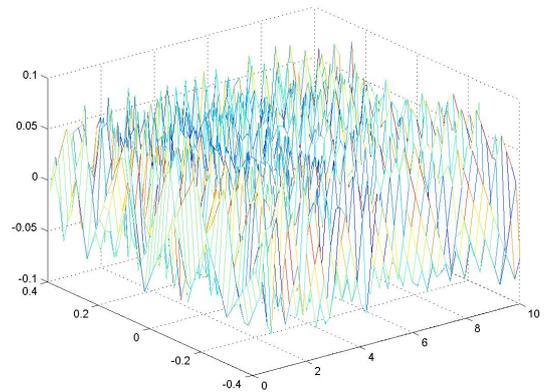
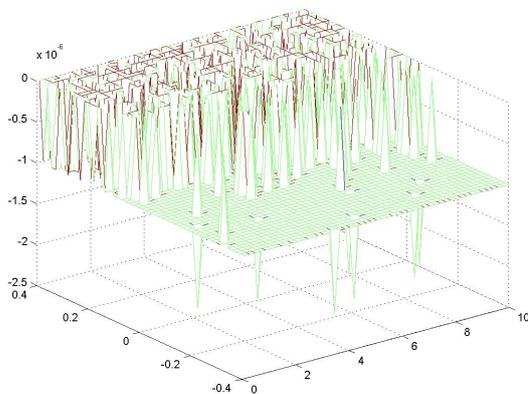
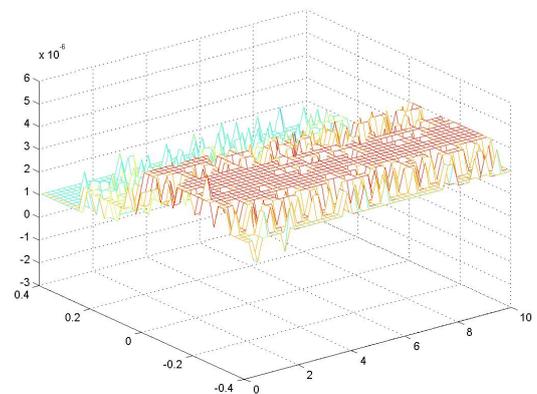


Abbildung 6.22: $u_{exakt} - u_{diskr}[10]$

Abbildung 6.23: $exakt - disk[100]$ Abbildung 6.24: $u_{exakt} - u_{disk}[100]$ Abbildung 6.25: $exakt - rekur[6]$ Abbildung 6.26: $u_{exakt} - u_{rekur}[6]$ Abbildung 6.27: $exakt - erw-b[6]$ Abbildung 6.28: $u_{exakt} - u_{erw-b}[6]$

Abbildung 6.29: $exakt - brent[6]$ Abbildung 6.30: $u_{exakt} - u_{brent[6]}$

Es zeigt sich, dass sowohl *brent* als auch *erw-b* sowie *rekur* ähnliche Ergebnisse bei den Wertefunktionen und der optimalen Steuerung aufweisen. Deswegen schauen wir uns diese noch genauer an.

Abbildung 6.31: $rekur[6] - brent[6]$ Abbildung 6.32: $erw-b[6] - brent[6]$

Bei genauerem Hinsehen sind keine bedeutenden Unterschiede auszumachen, zumal wir nur mit einer Genauigkeit von $\epsilon = 0.00001$ gerechnet haben. Deswegen wird hier zusätzlich die benötigte Anzahl an Funktionsberechnungen als Vergleichsgröße herangezogen. Bei durchschnittlich 52.7 Funktionsberechnungen pro Maximierung

stellt *brent* hier deutlich vor *erw-b* mit 66.9 und *rekur* mit 117.5 das schnellste Verfahren.

Die obigen Zahlen beziehen sich auf die Gesamtfunktion. Doch wie kommt es zu den Zahlen? Wie läuft eine Maximierung im Detail ab? Kann man die im Kapitel 5 gemachten Annahmen bezüglich des Ablaufs einer Maximierung bestätigen oder waren sie zu optimistisch? Wir betrachten dazu exemplarisch die Maximierung des Knotenpunktes x_{368} in der Iteration (Iter=100).

admit_intervall() benötigt, um aus dem Ausgangsintervall $[0.1; 13.5]$ das zulässige Intervall, nämlich $[0.1; 4.859766]$, für diesen Punkt zu bestimmen 23 Funktionsevaluationen.

init_value() benötigt zusätzliche 7 Funktionsberechnungen (Intervall=6) um das Starttripel

x_1	x_2	x_3
2.479883	3.273177	4.066472

zu bestimmen, welches für die Anfangswerte bei *brent()* verwendet wird. Die dabei ermittelten Werte der relevanten Punkte während der Maximierung sind der nachfolgenden Tabelle zu entnehmen.

Iter	Art	a	x	b	u	x-a	b-x	Intervall
Start		2.479883	3.273177	4.066472		0.793294	0.793294	1.586588
1	2	2.970166	3.273177	4.066472	2.970166	0.303011	0.793294	1.096306
2	1	3.273177	3.322589	4.066472	3.322589	0.049411	0.743882	0.793294
3	1	3.273177	3.300392	3.322589	3.300392	0.027214	0.022197	0.049411
4	1	3.273177	3.300392	3.301627	3.301627	0.027214	0.001234	0.028449
5	2	3.273177	3.289997	3.300392	3.289997	0.016819	0.010394	0.027214
6	1	3.273177	3.285441	3.289997	3.285441	0.012263	0.004555	0.016819
7	1	3.285441	3.285473	3.289997	3.285473	0.000031	0.004523	0.004554
8	1	3.285473	3.285480	3.289997	3.285480	0.000006	0.004517	0.004523
9	1	3.285477	3.285480	3.289997	3.285477	0.0000025	0.004517	0.004519
10	1	3.285477	3.285480	3.285482	3.285482	0.0000025	0.0000025	0.000005

6.5.1 Bemerkung

Art bezeichnet die jeweils durchgeführte Methode der Intervallreduzierung.

1: *Parabolische Interpolation*

2: *Goldener Schnitt*

brent() hat zusätzliche 10 Funktionsevaluationen benötigt. Zusammen kommen wir dann auf 40 Funktionsberechnungen für eine Maximierung. Der Wert liegt unter dem Durchschnitt von 52.74. Es war jedoch eine sehr schnelle Maximierung, da zu 80% parabolische Interpolationen angewendet werden konnten. Dies kann nicht immer garantiert werden, wie der folgende Gitterpunkt x_{369} innerhalb der gleichen Iteration (Iter=100) zeigt.

admit_intervall() benötigt, um aus dem Ausgangsintervall [0.1; 13.5] das zulässige Intervall, nämlich [0.1; 5.002325], für diesen Punkt zu bestimmen 23 Funktionsevaluationen.

init_value() benötigt zusätzliche 7 Funktionsberechnungen (Intervall=6) um das Starttripel

x_1	x_2	x_3
2.551162	3.368217	4.185271

zu bestimmen. Diese Werte werden als Anfangswerte bei *brent()* verwendet. Der detaillierte Verlauf der Maximierung ist der nachfolgenden Tabelle zu entnehmen.

Iter	Art	a	x	b	u	x-a	b-x	Intervall
Start		2.551162	3.368217	4.185271		0.817054	0.817054	1.634108
1	2	3.056130	3.368217	4.185271	3.056130	0.312086	0.817054	1.129141
2	1	3.368217	3.412324	4.185271	3.412324	0.044106	0.772947	0.817054
3	1	3.368217	3.412324	3.563330	3.563330	0.044106	0.151005	0.195112
4	2	3.368217	3.412324	3.470003	3.470003	0.044106	0.057679	0.101786
5	1	3.412324	3.428603	3.470003	3.428603	0.016279	0.041399	0.057679
6	1	3.412324	3.424539	3.428603	3.424539	0.012215	0.004064	0.016279
7	1	3.412324	3.421438	3.424539	3.421438	0.009113	0.003101	0.012215

Iter	Art	a	x	b	u	x-a	b-x	Intervall
8	2	3.412324	3.417956	3.421438	3.417956	0.005632	0.003481	0.009113
9	1	3.413935	3.417956	3.421438	3.413935	0.004021	0.003481	0.007502
10	2	3.416420	3.417956	3.421438	3.416420	0.001535	0.003481	0.005017
11	1	3.417956	3.419140	3.421438	3.419140	0.001183	0.002297	0.003481
12	1	3.417956	3.418604	3.419140	3.418604	0.000647	0.000536	0.001183
13	1	3.417956	3.418559	3.418604	3.418559	0.000602	0.000045	0.000647
14	2	3.417956	3.418329	3.418559	3.418329	0.000372	0.000230	0.000602
15	2	3.418186	3.418329	3.418559	3.418186	0.000142	0.000230	0.000372
16	2	3.418186	3.418329	3.418417	3.418417	0.000142	0.000087	0.000230
17	2	3.418274	3.418329	3.418417	3.418274	0.000054	0.000087	0.000142
18	1	3.418274	3.418329	3.418351	3.418351	0.000054	0.000022	0.000076
19	2	3.418308	3.418329	3.418351	3.418308	0.000020	0.000022	0.000042
20	1	3.418308	3.418329	3.418333	3.418333	0.000020	0.000004	0.000025
21	2	3.418321	3.418329	3.418333	3.418321	0.000008	0.000004	0.000012
22	1	3.418321	3.418329	3.418331	3.418331	0.000008	0.000002	0.000010
23	2	3.418326	3.418329	3.418331	3.418326	0.000003	0.000002	0.000005

brent() hat jetzt zusätzliche 23 Funktionsevaluierungen benötigt. Zusammen kommen wir dann auf 53 Funktionsberechnungen für eine Maximierung. Der Anteil der parabolischen Interpolation an der Maximierung ist auf $\frac{12}{23}$ gefallen. Er kann sogar auf knapp 20%, wie beim Knoten x_{89} (siehe Anhang D), fallen. Dies hängt davon ab, wo auf dem Gitter sich das Maximum befindet. Ist es innerhalb der durch die Gitterpunkten entstehenden Segmente, so funktioniert *brent()* recht gut. Befindet sich der Extrempunkt jedoch auf der Verbindungslinie zwischen zwei Knotenpunkten, sprich auf der Kante des Segments, so kann die parabolische Interpolation nicht die gewünschten Ergebnisse liefern, da zwischen den Segmenten oft Knicke vorkommen. Dadurch steigt die Anzahl der weniger effektiven Goldenen Schnitte innerhalb der Maximierung.

Nachdem wir nun das Brent-Verfahren genauer betrachtet haben, wollen wir uns nun *erw-b()* zuwenden. Auch hier werden wir den detaillierten Maximierungs-

vorgang anhand zweier Knotenpunkte zeigen. Zum besseren Vergleich werden wir die gleichen Punkte wie bei *brent()* verwenden.

Die Vorkonditionierung ist exakt gleich. Somit können wir gleich mit dem eigentlichen Maximierungsvorgang des Knotens x_{368} beginnen.

Iter	Art	a	x	b	u	x-a	b-x	Intervall
Start		2.479883	3.273177	4.066472		0.793294	0.793294	1.586588
1	1	3.273177	3.284509	4.066472	3.284509	0.011331	0.781963	0.793294
2	3	3.273177	3.284509	3.295840	3.295840	0.011331	0.011331	0.022662
3	4	3.284498	3.284509	3.295840	3.284498	0.000011	0.011331	0.0011342
4	2	3.284509	3.284520	3.295840	3.284520	0.000011	0.0011320	0.011331
5	1	3.284520	3.285481	3.295840	3.285481	0.000961	0.010359	0.011320

6.5.2 Bemerkung

Art bezeichnet die jeweils durchgeführte Methode der Intervallreduzierung.

- 1: *Parabolische Interpolation durch die inneren drei Punkte. Dabei bekommt man einen Punkt der dx vom bisherigen Maximum entfernt ist.*
- 2: *Die Auswertung des Punktes, der $-dx$ vom bisherigen Maximum entfernt ist.*
- 3: *Die Auswertung des Punktes, der $2dx$ vom bisherigen Maximum entfernt ist.*
- 4: *Parabolische Interpolation durch die äußeren drei Punkte. Auch hier bekommt man einen Punkt, der dx vom bisherigen Maximum entfernt ist.*

erw-b() hat zusammen $23 + 7 + 5 = 35$ Funktionsberechnungen für diese Maximierung benötigt. Dies ist das schnellste Ergebnis, dass wir ermittelt haben, weil das Abbruchkriterium $|y - y_{guess}| \leq \epsilon^2$ rasch unterschritten worden ist: $9.01 * 10^{-11} < 1.0 * 10^{-10}$. Für die anderen Knoten ist das Verfahren jedoch nicht so effektiv. Wie bereits bei *brent()* nehmen wir als Beispiel den Knotenpunkt x_{369} . Auch hier sind die Vorkonditionierungswerte wie bei *brent()*. Die Maximierung ergibt:

Iter	Art	a	x	b	u	x-a	b-x	Intervall
Start		2.551162	3.368217	4.185271		0.817054	0.817054	1.634108
1	1	3.368217	3.377609	4.185271	3.377609	0.009392	0.807662	0.817054
2	3	3.377609	3.387001	4.185271	3.387001	0.009392	0.798269	0.807662
3	4	3.387001	3.417247	4.185271	3.417247	0.030245	0.768024	0.798269
4	3	3.387001	3.417247	3.447492	3.447492	0.030245	0.030245	0.060490
5	1	3.387001	3.417247	3.423622	3.423622	0.030245	0.006375	0.036620
6	2	3.410871	3.417247	3.423622	3.410871	0.006375	0.006375	0.012750
7	4	3.410871	3.417247	3.423620	3.423620	0.006375	0.006372	0.012748
8	2	3.410874	3.417247	3.423620	3.410874	0.006372	0.006372	0.012745
9	1	3.417247	3.419390	3.423620	3.419390	0.002143	0.004229	0.006372
10	3	3.417247	3.419390	3.421534	3.421534	0.002143	0.002143	0.004287
11	4	3.417247	3.419282	3.419390	3.419282	0.002035	0.000108	0.002143
12	3	3.417247	3.419174	3.419282	3.419174	0.001927	0.000108	0.002035
13	1	3.417247	3.418677	3.419174	3.418677	0.001430	0.000497	0.001927
14	3	3.417247	3.418179	3.418677	3.418179	0.000932	0.000497	0.001430
15	4	3.417930	3.418179	3.418677	3.417930	0.000248	0.000497	0.000746
16	2	3.418179	3.418428	3.418677	3.418428	0.000248	0.000248	0.000497
17	1	3.418179	3.418426	3.418428	3.418426	0.000246	0.000002	0.000248
18	3	3.418179	3.418423	3.418426	3.418423	0.000244	0.000002	0.000246
19	4	3.418179	3.418332	3.418423	3.418332	0.000152	0.000091	0.000243
20	3	3.418241	3.418332	3.418423	3.418241	0.000091	0.000091	0.000182
21	1	3.418241	3.418332	3.418349	3.418349	0.000091	0.000016	0.000107
22	2	3.418316	3.418332	3.418349	3.418316	0.000016	0.000016	0.000033
23	4	3.418316	3.418332	3.418346	3.418346	0.000016	0.000014	0.000030
24	2	3.418318	3.418332	3.418346	3.418318	0.000014	0.000014	0.000028
25	1	3.418318	3.418332	3.418335	3.418335	0.000014	0.000003	0.000017
26	2	3.418318	3.418330	3.418332	3.418330	0.000012	0.000002	0.000014
27	4	3.418318	3.418327	3.418330	3.418327	0.000009	0.000003	0.000012
28	3	3.418325	3.418327	3.418330	3.418325	0.000002	0.000003	0.000005

Die benötigte Zahl der Funktionsevaluationen ist nun auf $23 + 7 + 28 = 58$ gestiegen. Das Wertetripel konnte nicht wie angenommen ausbalanciert gehalten werden. Dadurch ist das Intervall nur langsam verkleinert worden. Zwischen Iteration 6 und 8 beispielsweise wurde das Intervall lediglich um 0.000005 verkleinert. $\min(|y - y_{guess}|) = 7.59 * 10^{-8} > 1.0 * 10^{-10}$

Eine Übersicht der Maximierungsvorgänge in den ersten 500 Knotenpunkten während dieser Iteration (Iter=100) findet sich im Anhang D. Aufgrund der großen Menge an Daten sind dort zu jedem Knotenpunkt nur die verwendeten Methoden der Intervallreduzierung und die daraus resultierende Anzahl an Funktionsbestimmungen dargestellt.

Diese Daten sind auch auf der beigelegten CD-ROM im Verzeichnis `\Arbeit\Daten` zu finden.

6.6 Räuber-Beute-Modell

Als letztes wollen wir ein Modell betrachten, in dem die optimale Ausbeute aus einem sich gegenseitig beeinflussenden System von Faktoren betrachtet wird. Es stammt aus der Arbeit [13]. Das optimale Steuerungsproblem ist gegeben durch

$$\begin{aligned}\dot{x}_1(t) &= (a_0 - a_2x_2 - a_1x_1 - u)x_1 \\ \dot{x}_2(t) &= (b_1x_1 - b_0 - b_2x_2 - u)x_2\end{aligned}$$

und die dazugehörige Ertragsfunktion lautet

$$g(x_1, x_2, u) = \frac{1}{1 + x_1u}x_1u + \frac{1}{1 + x_2u}x_2u - \frac{u}{2}$$

Man könnte das Modell folgendermaßen interpretieren: Zwei Fischarten, x_1 - ein Beutefisch und x_2 - ein Räuberfisch, leben im gleichen Areal. Ein Fischer versucht nun durch die richtigen Fangraten sein Gewinn u zu maximieren.

$a_0 = 1.04$, $a_1 = 0.01$, $a_2 = 0.07$, $b_0 = 1.01$, $b_1 = 0.2$ und $b_2 = 0.01$ sind die hier verwendeten Parameter. Die restlichen Parameter lauten:

h	δ	ϵ	$n[0]$	$n[1]$	u
0.05	5	0.00001	50	50	[0;3]

Die dadurch generierten Ergebnisse sind der nachstehenden Tabellen zu entnehmen:

Verfahren	Intervall	Gesamt	Iter	Min	Max	Schnitt
brent	3	1341767	30	0.00	0.31528	17.19
brent	6	1509976	30	0.00	0.31532	19.35
brent	10	1789639	30	0.00	0.31533	22.93
brent	20	2471655	30	0.00	0.31533	31.67
diskr	10	858330	30	0.00	0.31296	11.00
diskr	100	7881030	30	0.00	0.31531	101.00
diskr	1000	78108030	30	0.00	0.31533	1001.00
rekur	6	6783060	30	0.00	0.31533	86.92
erw-b	3	5339505	30	0.00	0.31528	68.42
erw-b	6	1976732	30	0.00	0.31532	25.33
erw-b	10	2063206	30	0.00	0.31533	26.44
erw-b	20	2681660	30	0.00	0.31533	34.36

Es zeigt sich, dass ab einer gewissen Zahl der Intervalle bei *brent* das Ergebnis sich nicht mehr verbessert. Es ist also eine angemessene Anzahl von Teilintervallen zu wählen. Der relativ hohe Durchschnittswert bei *erw-b*[3] gründet wieder in der hohen Maximalzahl an Funktionsberechnungen pro Maximierung. Sie liegt bei 2278. Es zeigt sich einmal mehr, dass statt der erhofften schnellen Eingrenzung des Maximums, *erw-b* immer wieder in eine Art äquidistante Schrittfunktion mit einer sehr kleinen Schrittweite verfällt. Die nachfolgenden Abbildungen zeigen die Wertefunktion sowie die optimale Steuerung für das Räuber-Beute-Modell, welche mit *brent*[6] ermittelt worden sind.

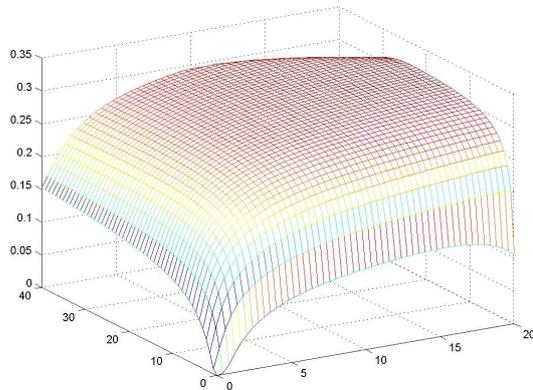


Abbildung 6.33: Wertefunktion
für das Räuber-Beute-Modell

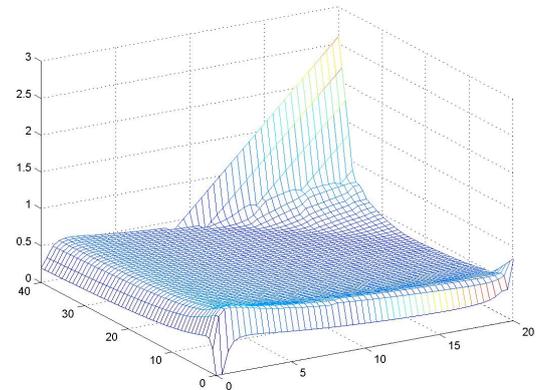


Abbildung 6.34: Optimale Steuerung
für das Räuber-Beute-Modell

Auch diesmal kommt die lineare Diskretisierung im Ergebnis nicht an *brent* heran. Die 10 Teilintervalle liefern für eine möglichst optimale Steuerung zu wenig Funktionswerte.

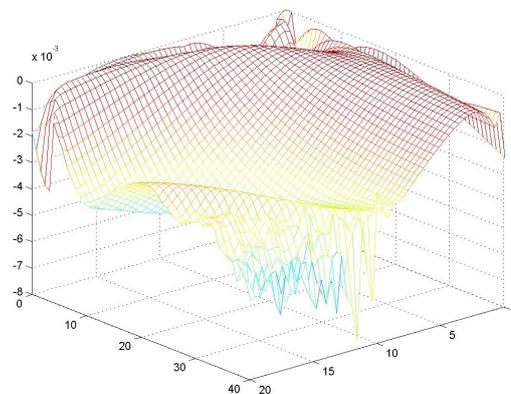


Abbildung 6.35: *diskr*[10] – *brent*[6]

Die Erhöhung der Intervalldichte auf 100 bewirkt die bereits in den vorherigen Beispielen beobachtete Ergebnisse. Das Resultat verbessert sich im Vergleich zu 10 Intervallen, jedoch steigt der Aufwand auf ein Maß, wodurch dieses Verfahren mit

den anderen nicht mehr konkurrieren kann. Sogar das Verfahren der rekursiven Suche braucht dann weniger Funktionsevaluationen.

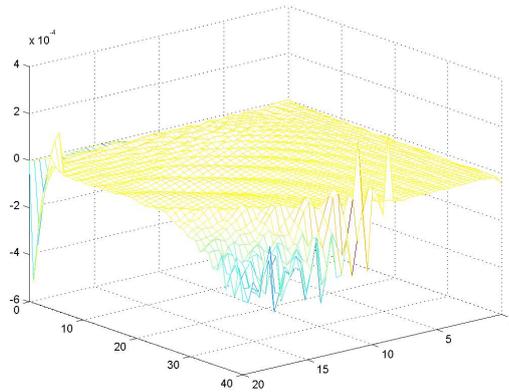


Abbildung 6.36: `diskr[100] - brent[6]`

Als letztes schauen wir uns nochmal den Unterschied zwischen `erw-b` und `brent` an. Beide Wertefunktionen sind exakt gleich. Aber mit durchschnittlich 3.5 Funktionsberechnungen pro Maximierung weniger als `erw-b` kann `brent` erneut als das bessere Verfahren angesehen werden.

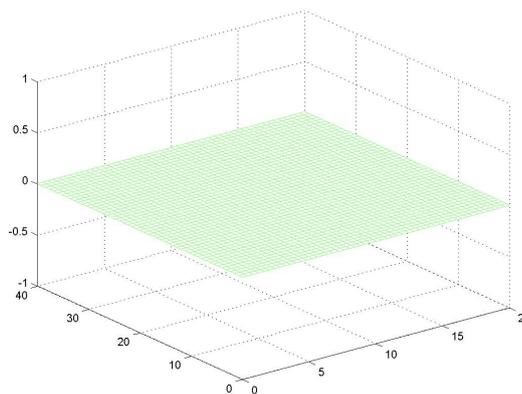


Abbildung 6.37: `erw-b[10] - brent[10]`

Kapitel 7

Schluss

Die vorgestellten Optimierungsstrategien sind nun an 5 Beispielen getestet worden. Mit den erhaltenen Datensätzen kann nun die Frage angegangen werden, ob die parabolische Interpolation tatsächlich effektiver als die Methode der äquidistanten Diskretisierung ist.

Die Ergebnisse zeigen, dass *brent()* und *erw-b()* bei ähnlichem Aufwand an Funktionsberechnungen deutlich bessere Ergebnisse als *diskr* erzielen. Steigert man die Anzahl der äquidistanten Unterteilungen bei *diskr*, so verbessert sich das Ergebnis erwartungsgemäß jedoch nicht soweit, als dass es die Ergebnisse der parabolischen Interpolation übertrifft.

Anzumerken ist jedoch, dass sowohl *brent()* als auch *erw-b()* Startwerte benötigen, welche bisher nur mit einem recht hohen Aufwand zu ermitteln sind. Die entsprechenden Funktionen arbeiten mit einer Bisektion, was eine zielführende, aber nicht die schnellste Methode darstellt. Dieser Teil des Algorithmus kann sicher noch effektiver implementiert werden und damit den Gesamtaufwand für eine Maximierung weiter verringern.

Eine weiterer Punkt ist die geeignete Wahl des Parameters *Intervall* bei den Verfahren der parabolischen Interpolation. Ein zu kleiner Wert macht das Ergebnis nicht befriedigend. Ein großer Wert verbraucht zusätzliche Funktionsevaluationen ohne das Ergebnis positiv zu beeinflussen. In der Testphase hat sich der Wert 6 als

recht günstig erwiesen.

Die in den Vergleichen der Verfahren auftretenden Spitzen entlang der Skiba-Kurven könnten durch ein dichteres Gitternetz eliminiert werden.

Man kann sagen, dass die Interpolation der Diskretisierung vorzuziehen ist.

Doch welches der beiden Verfahren - *brent()* oder *erw-b()* - ist nun besser?

Das erweiterte Brent-Verfahren hat innerhalb der Testphase deutliche Schwächen gezeigt, welche es dem eigentlichen Brent-Verfahren oft unterlegen macht. Der Verzicht auf den Goldenen Schnitt zugunsten einer zweiten parabolischen Interpolation fördert ein Verfallen in ein Verhalten, wie beim *stepper*, sprich es wird fortlaufend nur ein Wert, der nahe am derzeitigen Maximum liegt als neues Maximum errechnet. Dies verursacht vor allem im Beispiel 4 und Beispiel 5 viele kleine Schritte auf dem Weg zur Ermittlung der tatsächlichen Maximalstelle und fordert bis zu 3744 bzw. 2278 Funktionsevaluationen pro Maximierung. Auch wenn dies nicht oft vorkommt, so verlangsamt es das Gesamtverfahren und erhöht die durchschnittliche Evaluationsszahl unnötig.

Dass *brent()* nicht, wie in der Theorie erhofft, die erwartete Konvergenzgeschwindigkeit aufweist, liegt an der Struktur der betrachteten Modelle. Durch die Diskretisierung im Raum entstanden auf der gesamten Fläche einzelne Segmente, welche oft an den Kanten nicht differenzierbare Stellen hatten. Lag nun ein Maximum direkt auf solch einer Kante, so musste das Verfahren oft den Goldenen Schnitt anwenden und konnte nicht in der gewohnten Geschwindigkeit auslaufen. Lag es innerhalb eines Segments, so verhielt es sich wie in der Theorie.

Vor allem das ökonomische Wachstumsmodell mit der vorhandenen exakten Lösung zeigt, dass die durch *brent()* ermittelten Ergebnisse auch die tatsächlichen Werte recht gut approximieren. Somit kann mit *brent()* nicht nur die Zahl der Funktionsberechnungen verringert, sondern auch das Ergebnis exakter bestimmt werden.

Anhang A

Beispiele

Beispiel 1: Einfaches Modell

Das zeitkontinuierliche optimale Steuerungssystem

$$\begin{aligned}\dot{x}_1(t) &= x_1 + ux_2 \\ \dot{x}_2(t) &= -x_2\end{aligned}$$

und die dazugehörige Ertragsfunktion

$$g(x_1, x_2, u) = -1.0 + x_1 - 0.3u$$

Beispiel 2: Investitionsmodell

Dieses Beispiel stammt aus [12]. Es ist definiert durch das zeitkontinuierliche optimale Steuerungssystem

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

und die Ertragsfunktion

$$g(x_1, x_2, u) = k_1\sqrt{x_1} - \frac{x_1}{1+k_2x_1^4} - c_1x_2 - \frac{c_2}{2}x_2^2 - \frac{\alpha}{2}u^2.$$

Die verwendeten Parameter: $\sigma = 0.25$, $k_1 = 2$, $k_2 = 0.0117$, $c_1 = 0.75$, $c_2 = 2.5$ und $\alpha = 12$.

Beispiel 3: Makroökonomisches Modell

Das zeitkontinuierliche optimale Steuerungssystem

$$\begin{aligned}\dot{x}_1(t) &= u - 0.55x_1 + \frac{x_1^2}{1+x_1^2} \\ \dot{x}_2(t) &= 0\end{aligned}$$

und die dazugehörige Ertragsfunktion

$$g(x_1, u) = 2\sqrt{u} - \frac{\lambda}{2}x_1^2 \quad \lambda \geq 0$$

bilden das makroökonomische Modell von W.Brock und D. Starret. Es ist in [13] zu finden.

Beispiel 4: Ökonomisches Wachstumsmodell

Dieses zeitdiskrete optimale Steuerungssystem ist gegeben durch

$$\begin{aligned}x_1(t+1) &= Ae^{x_2}x_1^\alpha - u \\ x_2(t+1) &= \rho x_2\end{aligned}$$

und die dazugehörige Ertragsfunktion

$$g(x_1, x_2, u) = \ln u$$

wobei $\rho, \alpha, A \in \mathbb{R}^+$. In dieser Arbeit betrachten wir die Werte $\rho = 0.9$, $\alpha = 0.34$, $A = 5$ und $\beta = 0.95$.

Dieses Beispiel stellt eine abgewandelte Version des ökonomischen Wachstumsmodells aus [14] dar.

Beispiel 5: Räuber-Beute-Modell

Das optimale Steuerungsproblem, vorgestellt in [13], ist gegeben durch

$$\begin{aligned}\dot{x}_1(t) &= (a_0 - a_2x_2 - a_1x_1 - u)x_1 \\ \dot{x}_2(t) &= (b_1x_1 - b_0 - b_2x_2 - u)x_2\end{aligned}$$

und die dazugehörige Ertragsfunktion lautet

$$g(x_1, x_2, u) = \frac{1}{1 + x_1u}x_1u + \frac{1}{1 + x_2u}x_2u - \frac{u}{2}$$

$a_0 = 1.04$, $a_1 = 0.01$, $a_2 = 0.07$, $b_0 = 1.01$, $b_1 = 0.2$ und $b_2 = 0.01$ sind die hier verwendeten Parameter.

Anhang B

Programmbedienung

- Wählbare Parameter

- Beim Programmaufruf

Mit dem Programm können 5 Beispiele gemäß Anhang A mit 4 verschiedenen Maximierungsverfahren berechnet werden.

Nach dem Programmstart kommt erst eine Abfrage nach dem Beispiel, das man bearbeiten möchte. Es soll eine Zahl zwischen 1 und 5 eingegeben werden.

Eine zweite Abfrage befasst sich mit den zur Verfügung stehenden Verfahren. Hier soll eine Zahl zwischen 1 und 4 eingegeben werden. Als letztes wird noch nach der Zahl der Intervalle gefragt. Bei der Diskretisierung ist es die Zahl der Teilintervalle im Ausgangsintervall $[a, b]$. Bei der rekursiven Suche die auf jeder Rekursionsstufe vorgenommene Aufteilung des aktuellen Intervalls. Bei *brent()* bzw. *erw-b()* ist es die Zahl der Zwischenschritte bei der dem eigentlichen Verfahren vorangehenden Funktion zur Bestimmung eines geeigneten Anfangstripels. Bei der Diskretisierung soll dabei eine Zahl x , $x \geq 1$, $x \in \mathbb{N}$, eingegeben werden. Bei den übrigen Verfahren muss $x \geq 3$ betragen.

- Im Quellcode

Im Quellcode können zusätzlich folgende Parameter geändert werden:

In der Funktion *beispiel()* befinden sich unter der jeweiligen Beispielnnummer folgende Parameter:

h - Schrittweite

δ - Diskontrate

ϵ - Genauigkeitsschranke, Abbruchkriterium

$n[0]$ - Anzahl der Gitterpunkte in $x[0]$ -Richtung

$n[1]$ - Anzahl der Gitterpunkte in $x[1]$ -Richtung

$a[0]$ - Beginn des Intervalls in $x[0]$ -Richtung

$b[0]$ - Ende des Intervalls in $x[0]$ -Richtung

$a[1]$ - Beginn des Intervalls in $x[1]$ -Richtung

$b[1]$ - Ende des Intervalls in $x[1]$ -Richtung

$u[0]$ - Beginn des Intervalls für die Steuervariable u

$u[1]$ - Ende des Intervalls für die Steuervariable u

In den Funktionen *Fkt_f()* und *Fkt_g()* stehen die jeweiligen Beispielfunktionen. Diese können dort jederzeit modifiziert werden.

- **Ausgabe**

- **Während des Programms**

Die erreichte Iterationszahl

Der aktuelle Wert des Abbruchfehlers gemäß Bemerkung 3.2.6.

- **Nach einem erfolgreichen Programmende**

Die Gesamtzahl der getätigten Funktionsevaluationen

Das Minimum und das Maximum der ermittelten Wertefunktion

Bei *erw-b()* wird zusätzlich die Maximalzahl der Funktionsberechnungen pro Maximierung, über das gesamte Verfahren gesehen, ausgegeben.

- **Speicherung der Daten**

Der ermittelte maximale Funktionswert sowie die dazugehörige optimale Steuerung für jeden Gitterpunkt werden in einer Datei abgelegt. Sie besteht aus 4 Spalten und $(N+1)$ Zeilen.

1. Zeile: $n[0]$ - Anzahl der Gitterpunkte in $x[0]$ -Richtung
 $n[1]$ - Anzahl der Gitterpunkte in $x[1]$ -Richtung
 N - Gesamtzahl der Gitterpunkte
 N - Gesamtzahl der Gitterpunkte

folgende Zeilen: $x[0]$ -Koordinate des Gitterpunktes
 $x[1]$ -Koordinate des Gitterpunktes
 Funktionswert am Gitterpunkt
 Optimale Steuerung am Gitterpunkt

Abhängig vom gewählten Verfahren werden die Datensätze unter dem Dateinamen `werte-brent.asc`, `werte-diskretisierung.asc`, `werte-rekursiv.asc` bzw. `werte-erweitert-brent.asc` gespeichert. So sind spätere Vergleiche der mittels verschiedener Verfahren ermittelten Ergebnisse (z.B. mit Matlab) möglich.

Zusätzlich wird eine Datei `werte-methode.asc` angelegt, die den detaillierten Verlauf der Maximierung bei `brent()` bzw. `erw-b()` dokumentiert, sprich wieviele Iterationsschritte und welche Methoden der Maximierung verwendet wurden. Diese Matrix hat die Grösse $N \times 100$, da `brent()` durch eine im Quellcode vorgenommene Einstellung spätestens nach dem 100. Iterationsschritt die Maximierung beendet. Obwohl `erw-b()` manchmal mehr als 100 Iterationsschritte für die Bestimmung eines Maximums benötigt (siehe **Ausgabe**), werden aus Speichergründen nur die ersten 100 Schritte pro Maximierung dokumentiert. Dies kann jedoch jederzeit bei der Allokation des Speicherplatzes für die Matrix `methode[] []` im Quellcode geändert werden. Weitere Informationen zu `werte-methode.asc` sind im Anhang D nachzulesen.

Anhang C

Verwendete Funktionen

Die Aufgaben der im Programm verwendeten Funktionen sollen zur besseren Orientierung hier kurz erläutert werden.

- Rahmenfunktionen:

- `main()`:

Hier kann das gewünschte Beispiel sowie das Optimierungsverfahren gewählt werden. Anschliessend wird der benötigte Speicherplatz für globale Variablen allokiert. Am Ende werden die ermittelten Daten gemäß Anhang B gespeichert.

- `algorithmus()`:

Das Iterationsverfahren wird hier durchgeführt. Dabei wird innerhalb der Iterationsschritte die zu Beginn gewählte Optimierungsstrategie benutzt.

- `beispiel()`:

In dieser Funktion sind die für die Beispiele relevanten Parameter enthalten. Sie können dort vor dem Start nach Bedarf geändert werden.

- `Fkt_f()`:

Die den 5 Beispielen zugrundeliegenden Steuerungssysteme sind hier enthalten.

- `Fkt_g()`:
Analog zu `Fkt_f()` befinden sich hier die Ertragsfunktionen der Beispiele.
- `solution()`:
Für das ökonomische Wachstumsmodell (Beispiel 4) werden hier die exakte Wertefunktion sowie die optimale Steuerung berechnet.
- Vorkonditionierung:
 - `admit_intervall()`:
Hier wird das für den jeweiligen Gitterpunkt zulässige Intervall für die Steuerungsvariable u bestimmt.
 - `init_value()`:
Die Startpunkte für `brent()` bzw. `erw-b()` werden in dieser Funktion ermittelt.
- Brent-Verfahren:
 - `brent()`:
Diese Funktion berechnet das Maximum innerhalb eines gegebenen Intervalls. Dabei wechseln sich je nach Bedarf die parabolische Interpolation und der Goldene Schnitt ab.
 - `funktion()`:
Hier wird der Funktionswert des durch `brent()` ermittelten Punktes berechnet.
- Erweitertes Brent-Verfahren:
 - `erweitert_brent()`:
Eine Funktion, die zwischen `para_fit()`, `wide_fit()` und `line_fit()` als nächsten Maximierungsschritt entscheidet.
 - `para_fit()`:
Hier wird die parabolische Interpolation durch die inneren 3 Punkte gemacht.

- `flip()`:
Es wird der Punkt, der dx entfernt auf der anderen Seite von x als u liegt, berechnet.
 - `twostep()`:
Es wird der Punkt, der $2dx$ entfernt auf der gleichen Seite von x als u liegt, berechnet.
 - `wide_fit()`:
Hier wird die parabolische Interpolation durch die äusseren 3 Punkte gemacht.
 - `line_fit()`:
Diese Funktion legt durch die jeweils zwei rechten bzw. linken Punkte eine Gerade und ermittelt den Schnittpunkt.
 - `v_form()`:
Diese Funktion prüft, ob das vorliegende 5-Tupel eine "Dachform" aufweist. Nur dann ist das Ausführen von `line_fit()` sinnvoll.
 - `insert()`:
Jeder ermittelte Punkt wird hier entsprechend seines Funktionswertes in das 5-Tupel eingefügt.
- Sonstige Funktionen:
 - `Euler()`:
Der Euler-Schritt wird hier durchgeführt.
 - `Koordinaten()`:
Aus dem globalen Eckenindex werden die Koordinaten der Ecke berechnet.
 - `FindeRechteckKoord()`:
Ausgehend von den Koordinaten des Punktes wird der Rechteckindex, in dem sich der Punkt befindet, ermittelt.

- `Gitter()`:
Ermittelt aus dem globalen Rechteckindex und dem lokalen Eckenindex den globalen Eckenindex.
- `Wert()`:
Liefert den approximierten Wert v im Punkt x durch eine affin bilineare Funktion.

Anhang D

Auswertungstabellen

In diesem Anhang wird exemplarisch das Verhalten der auf der parabolischen Interpolation basierenden Verfahren während einer Maximierung des ökonomischen Wachstumsmodells (Beispiel 4) in der 100. Iteration gezeigt. Dabei entspricht jede Zeile einem Knoten i mit $i = 0, \dots, N$. Die Zahl der Einträge in der Zeile gibt an, wieviele Iterationen des Verfahrens eine Maximierung beansprucht hat. Von den 2600 Knoten haben wir jeweils die ersten 500 dargestellt. Die Informationen zu den übrigen Knoten sowie die Auswertungsdateien zu den anderen in dieser Arbeit betrachteten Beispielen sind auf der beigelegten CD-ROM im Verzeichnis `\Arbeit\Daten` zu finden. Die jeweiligen Dateien beinhalten statt *-werte-* das Wort *-methode-* im Dateinamen.

Die erste Tabelle zeigt das Verhalten des Brent-Verfahrens bei diesem Beispiel. Die auftretenden Zahlen stehen für die verwendete Methode innerhalb der Funktion:

- 1 : Parabolische Interpolation
- 2, 3 : Goldener Schnitt

D.0.1 Bemerkung

- 2 tritt auf, wenn im Vergleich zur vorletzten Funktionsevaluation $|x - u|_{neu} > \frac{1}{2}|x - u|_{alt}$ oder u ausserhalb des zulässigen Intervalls ist.
- 3 tritt auf, wenn während der vorangegangenen Funktionswertevaluation der Abstand $|x - u| \leq \epsilon$. Dies ist im ersten Schritt immer der Fall, da noch kein neuer Punkt u bestimmt worden ist.

Knoten	
0	3 1 1 2 1 2 1 1 2 1 1 2 1 2 1 2 1 1 2 1
1	3 1 2 1 2 1 1 1 2 1 2 2 1 1 2 2 2 2 1 2 1 1
2	3 1 1 1 2 1 1 1 2 1 2 1 1 2 1 1 2 1 1 1 3 1
3	3 1 1 2 1 1 2 2 2 1 1 2 2 2 2 1 1 2 2 2 2 2 2 2 1 2
4	3 1 1 2 1 2 1 1 2 1 1 2 2 2 1 2 1 1 2 2 2 1 2
5	3 1 1 2 1 2 1 1 2 2 2 2 1 2 1 1 1 1 1 2 2 1 1 3 2 1
6	3 1 1 1 1 1 1 1
7	3 1 1 1 2 1 2 1 1 2 1 1 2 1 1 2 1 1 2 2 2 2 1 1
8	3 1 1 2 1 2 1 1 2 1 1 2 1 2 1 1 1 3 2 2 1
9	3 1 1 2 1 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2
10	3 1 1 1 1 1 1 1
11	3 1 1 1 2 2 2 2 1 2 2 2 2 1 2 1 2 1 2 2 2 2 1
12	3 1 1 2 1 1 2 1 1 2 1 1 1 2 1 1 1
13	3 1 1 2 1 2 1 1 1 1 2 2 1 1 1 2 2 2 2 1 1 2 1 1
14	3 1 1 2 1 2 2 2 1 2 2 2 2 2 2 2 1 2 2 1 1 2 2 2 2
15	3 1 1 2 1 1 1 1 1 1
16	3 1 1 1 1 1 1 1 1 1
17	3 1 2 1 1 1 1 2 2 2 2 1 2 1 2 1 1 2 2 1 1 3 2 1 1
18	3 1 1 2 1 1 1 1 1 2 1 1 2 1 2 1 2 1
19	3 1 1 2 1 1 1 2 1 1 1 2 2 1 1 2 1 1 2 1 1 2
20	3 1 1 2 1 2 2 1 2 2 1 2 2 2 2 2 2 1 2 1 1 2 2 2 2 2
21	3 1 1 2 1 1 1 1 1 1
22	3 1 1 1 2 1 1 1 1 1
23	3 1 1 2 1 1 2 1 1 1 1 1
24	3 1 1 2 1 1 2 2 2 1 2 1 1 2 2 1 2 1 1 2 2 1 2
25	3 1 1 1 1 2 1 1 2 1 1 1 1 1 1 1 1
26	3 1 1 2 1 2 1 1 2 1 1 2 2 1 2 1 2 2 2 2 1 2 1 2
27	3 1 1 2 1 1 1 1 2 2 2 1 1 2 2 2 2 2 1 1 2 2 2 2 2 2 2
28	3 1 1 2 1 1 1 1 1 1
29	3 1 1 1 1 1 1 1 1 1
30	3 1 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
31	3 1 1 2 2 1 1 2 2 2 2 1 1 1 2 2 1 1 2 2 2 2 2 1
32	3 1 1 2 1 1 1 2 2 1 1 2 1 1 2 1 1 2 1 1 1
33	3 1 1 2 1 1 1 1 1 1 1 2 1 2 1 1
34	3 1 1 2 1 1 1 1 2 1 1 2 1 2 1 1 2 1 1 2 1 2
35	3 1 1 2 1 1 2 1 2 2 1 2 1 2 1 2 1 1 2 2 1 1 2 2 1 2
36	3 1 1 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1
37	3 1 1 1 1 1 1 1 1 1
38	3 1 1 1 1 1 1 1 1 1
39	3 1 1 1 1 1 1 1 1 1
40	3 1 1 1 1 1 1 1 1 1
41	3 1 1 2 2 1 2 2 1 2 2 1 1 2 2 2 2 2 2 2 2 2 2
42	3 1 1 2 1 1 2 1 1 2 2 2 2 1 1 2 2 2 2 1 2 1 2 1 2
43	3 1 1 2 1 1 2 1 1 2 1 2 1 1 2 1 1 2 1 2 1 1 1
44	3 1 1 2 1 1 1 2 1 1 2 1 1 1 1 3 1 1 3 1
45	3 1 1 1 1 2 1 1 2 1 1 2 1 1 1 2 1 2 1 1

Knoten	
46	3 1 1 2 1 1 2 2 1 2 1 1 1 2 2 1 1 1 1 3 1 1 2
47	3 1 1 2 1 1 2 1 1 1 2 1 2 1 1 1 2 2 2 2 1 2 1 2 1 2 2
48	3 1 1 2 1 2 2 1 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2
49	3 1 1 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
50	3 1 1 2 1 1 1 1 1 1
51	3 1 2 1 2 1 2 1 1 1 1 1 1 1 2 1 1 2 1
52	3 1 1 2 1 2 1 2 1 1 2 2 1 2 2 2 2 1 1 1 2 2 1 1
53	3 1 2 1 1 2 1 1 1 2 1 2 1 1 2 2 1 1
54	3 1 1 2 1 1 1 1 1 1
55	3 1 2 1 1 1 1 2 1 1 2 1 1 2 1 2 2 1 1 3 1 2 2 1 2
56	3 1 1 2 1 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2
57	3 1 1 1 1 1 1 1
58	3 1 2 1 1 1 1 1 2 1 1 1 1 2 1 1 3 1 1 1
59	3 1 1 2 1 2 1 2 1 1 2 2 1 1 2 2 1 2 1 1 1 2 2 2 2 1
60	3 1 1 2 1 1 1 1 1 1
61	3 1 1 1 2 1 1 1 1 1
62	3 1 2 1 1 1 1 2 1 2 1 1 2 2 2 2 1 1 1 2 2
63	3 1 1 2 1 2 1 2 1 2 1 1 1 1 2 1 1
64	3 1 1 2 1 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 1 1 2 2 1
65	3 1 1 2 1 1 1 1 1 1
66	3 1 1 1 1 1 1 1 1 1
67	3 1 2 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
68	3 1 1 2 1 1 1 1 1 1 2 2 1 1 1 2 1 1
69	3 1 1 2 1 1 2 1 1 2 1 1 1 2 2 1 2 1 2 1 2
70	3 1 1 2 1 1 2 2 1 1 2 2 1 1 2 2 2 2 2 1 2 2 2 2 1
71	3 1 1 2 1 1 1 1 1 1
72	3 1 1 1 2 1 1 1 1 1
73	3 1 1 2 1 1 2 1 1 2 2 2 1 2 2 2 2 2 1 1 1 2 2 2
74	3 1 1 2 1 1 2 2 1 2 1 2 1 1 1 1 2 1 1 2 1
75	3 1 1 2 1 1 1 1 2 1 1 2 2 1 1 1 1 3 1
76	3 1 1 2 1 2 2 1 2 1 2 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2
77	3 1 1 2 1 1 1 1 1 1
78	3 1 1 2 1 1 1 1 1 1
79	3 1 1 1 1 1 1 1
80	3 1 1 1 2 2 2 2 2 2 2 1 1 2 2 2 1 1 2 2 2
81	3 1 1 2 1 2 1 2 2 1 2 1 2 1 1 1 2 1 2 1 2
82	3 1 1 1 2 1 2 1 2 2 1 1 2 1 1 1 2 1
83	3 1 1 2 1 2 1 1 1 1 2 1 1 2 1 1 2 1 1 3
84	3 1 1 2 2 1 1 2 1 2 1 2 1 1 2 1 1 2 1 1
85	3 1 1 2 1 1 2 1 1 1 2 2 2 2 1 2 2 2 2 1 2 1 2 1 2 2 1
86	3 1 1 2 1 1 2 2 1 1 2 1 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2
87	3 1 1 2 1 1 1 1 1 1
88	3 1 1 2 1 1 1 1 1 1
89	3 1 1 2 1 1 1 1 1 1
90	3 1 1 2 1 1 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2

Knoten	
91	3 1 1 2 1 1 2 2 1 1 1 1 2 2 2 1 2 1 2 2 1 1 2 2 1
92	3 1 1 1 2 2 1 1 1 1 2 2 1 1 1 1 1 2 1 1 2 1 1 2
93	3 1 1 1 1 1 2 1 1 2 1 2 1 2 1 1 1 1 1 2 1
94	3 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 3 1 1
95	3 1 1 2 1 1 2 1 1 1 1 2 1 2 2 1 1 2 2 1 1 3
96	3 1 1 2 1 1 2 1 1 1 2 1 1 1 1 2 2 1 1 2 2 2 2 1 1 1
97	3 1 1 2 1 2 2 1 1 1 2 2 2 2 2 2 2 2 1 1 1 2 2 2 2 2
98	3 1 1 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
99	3 1 1 2 1 1 1 1 1 1
100	3 1 1 2 1 1 1 1 1
101	3 1 1 1 2 1 1 1 1 1
102	3 1 2 1 2 1 1 1 2 1 2 2 2 2 1 1 1 2 2 1 2
103	3 1 1 2 1 2 2 1 2 2 2 2 2 2 2 2 2 2 1 1 1 2 2 2
104	3 1 1 1 1 1 1 2 1 2 1 1 1 1 2
105	3 1 1 2 1 1 1 1 1 1
106	3 1 2 1 1 1 2 1 1 1 2 1 1 2 1 1 1
107	3 1 1 2 1 1 1 1 1 1
108	3 1 1 1 1 2 2 2 1 2 2 2 2 1 2 1 1 2 2 2 1
109	3 1 1 2 1 1 1 1 1 2 1 1 1 2 1
110	3 1 1 2 1 2 2 2 2 1 2 2 2 2 2 2 2 1 1 2 2 2 1 1 3 2
111	3 1 1 1 1 1 1 1 1
112	3 1 1 1 2 2 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2
113	3 1 1 1 2 2 1 1 1 2 1 2 1 1 2 1 2 1 2
114	3 1 1 2 1 1 1 2 1 1 2 2 2 2 1 1 2 1 2 1 1 1 2 1
115	3 1 1 2 1 1 2 2 2 1 2 2 1 1 2 2 2 2 1 1 2 2 2 2 2
116	3 1 1 2 1 1 1 1 1 1 1
117	3 1 1 2 1 1 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2
118	3 1 1 2 1 1 1 2 1 1 2 1 1 1 2 2 1 1 2 1 1
119	3 1 1 2 1 2 1 1 2 2 2 1 1 1 1 2 2 1 1 2 1 1
120	3 1 1 2 1 1 1 1 2 2 2 1 2 2 2 2 2 2 2 2 1 2 2 2 2
121	3 1 1 2 1 1 1 1 1 1
122	3 1 1 1 2 1 1 1 1 1
123	3 1 1 2 1 2 1 2 2 1 2 1 2 1 2 2 2 2 1 2 1 2
124	3 1 1 2 1 1 2 1 2 1 1 2 1 2 1 2 1 1 2 1 2 1
125	3 1 1 2 1 1 1 1 2 1 1 2 1 2 1 1 2 1 1 2 1 2
126	3 1 1 2 1 2 2 2 2 1 2 2 1 2 1 1 2 2 2 2 1 1 1 2 2 1
127	3 1 1 2 1 1 1 1 1 1 1
128	3 1 1 2 1 1 1 1 1 1
129	3 1 1 1 1 1 1 1
130	3 1 1 2 2 1 1 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2
131	3 1 1 1 2 1 1 2 2 2 2 1 2 1 2 2 2 2 1 2 1 2 1
132	3 1 1 1 2 2 1 1 1 1 1 1 1 2 1 2 1 2 1
133	3 1 1 1 2 1 1 2 1 1 1 1 1 1 1 3 1 2
134	3 1 1 2 1 2 1 2 1 2 1 1 1 1 2 2 1 1 1 2 2 2 2 1 2
135	3 1 1 2 1 2 1 2 2 1 2 2 1 2 2 1 1 2 2 1 1 2 2 2 1

Knoten	
136	3 1 1 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2
137	3 1 1 1 1 1 1 1
138	3 1 1 1 1 1 1
139	3 1 1 1 2 1 1 1 1
140	3 1 1 1 2 1 2 1 1 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2
141	3 1 1 1 2 2 1 2 2 1 2 1 1 1 1 1 2 2 1 2 1
142	3 1 1 2 1 1 2 1 1 1 1 1 1 1 2 1 2 1 1
143	3 1 1 2 1 1 1 1 1 2 1 1 1 1 1 1 2
144	3 1 1 2 1 2 1 1 2 1 1 2 1 1 2 1 1 2 2 2 2 1 2 1
145	3 1 1 2 1 1 2 2 1 2 1 1 2 2 2 2 2 2 1 1 2 1 2 2 2 1 2 2
146	3 1 1 2 1 1 1 2 1 1 1 2 2 2 2 2 2 2 2 2 1 1 2 1
147	3 1 1 2 1 1 1 1 1 1 1
148	3 1 1 1 1 1 1 1
149	3 1 1 1 2 1 1 1 1
150	3 1 1 1 2 1 2 2 1 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2
151	3 1 1 1 2 1 2 2 1 2 2 2 1 1 1 1 2 2 1 1 2 2 2 2 1
152	3 1 1 1 2 2 1 1 1 2 1 1 2 1 2 1 2 1 1 2
153	3 1 2 1 2 2 1 2 1 1 2 2 1 2 1 1 2 2 1 1 2 2 1 2 2
154	3 1 1 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
155	3 1 1 2 1 1 1 1 2 1 1 1 1 2 1
156	3 1 1 2 1 1 1 1 1
157	3 1 1 2 1 2 1 2 1 1 2 2 1 1 1 1 2 1 2
158	3 1 1 2 1 1 1 1 1
159	3 1 1 1 2 2 2 2 1 2 1 2 1 2 1 2 2 2 2 1 1
160	3 1 1 2 1 2 1 2 1 1 2 1 1 2 1 1 2 1 1 2 1 2
161	3 1 1 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
162	3 1 1 1 1 1 1 1 1
163	3 1 2 1 1 1 1 2 1 1 1 2 1 2 2 1 1 2 1 1 1 3 1
164	3 1 1 2 1 1 1 1 1 2 1 1 2 1 2 1 1 2 1
165	3 1 1 2 1 1 2 2 1 1 2 2 2 1 1 2 2 2 2 2 2 2 2 2 1
166	3 1 1 1 1 1 1 1 1
167	3 1 1 1 1 1 2 2 1 2 2 2 2 1 1 2 2 2 2 2 2 2 2
168	3 1 1 2 1 1 1 1 1 1 2 1 2 1 2 1 1 2 1 1 3
169	3 1 1 2 1 2 1 1 2 1 1 2 1 1 1 2 1 1 2 1 2 1 1 1 2
170	3 1 1 2 1 1 1 1 1 1 1
171	3 1 1 2 1 1 1 1 1
172	3 1 1 1 2 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2
173	3 1 1 2 1 2 1 2 1 2 1 1 1 2 2 2 1 1 1 1 2 2 1
174	3 1 1 2 1 1 2 2 1 1 1 1 2 1 1 3 1 1 2 1
175	3 1 1 1 1 2 1 1 1 1 2 1 1 2 1 1 2 1
176	3 1 1 2 1 1 2 1 1 1 2 2 1 2 1 1 2 2 1 1 2 2 1 1 2 2 2
177	3 1 1 2 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
178	3 1 1 2 1 1 1 1 1 1 1
179	3 1 1 1 1 1 1 1 1
180	3 1 2 1 1 1 2 1 2 1 1 2 2 2 2 1 2 2 2 1 2 2 2 1

Knoten	
181	3 1 1 1 2 1 2 2 2 2 1 2 1 1 2 1 2 1 1 2 2 1 2 1
182	3 1 1 2 1 1 1 1 1 2 1 2 1 2 1 2 1 1
183	3 1 1 1 1 2 1 1 2 1 2 1 1 1 1 1 2 1 2 1
184	3 1 1 2 1 2 1 2 1 2 1 2 2 1 1 2 1 1 1 1 2 2 1 2 1
185	3 1 1 2 1 1 2 2 1 2 2 2 2 1 2 2 2 2 1 2 1 1 3 2 2 1
186	3 1 1 2 1 1 1 1 1 1
187	3 1 1 1 2 1 1 1 1 1
188	3 1 1 1 2 1 1 1 1 1
189	3 1 1 1 2 1 1 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2
190	3 1 1 1 2 2 1 2 2 2 1 1 1 1 2 2 1 2 1 1 1
191	3 1 1 1 1 1 1 1 2 1 1 2 1 1 3 1 1 1
192	3 1 1 2 1 2 1 2 1 2 2 1 1 1 1 2 1 1 1 1 1
193	3 1 1 2 1 1 2 2 2 2 1 2 2 2 2 1 1 2 2 2 1 2 2 2 2 2
194	3 1 1 1 1 1 1 1
195	3 1 1 1 1 1 1 1
196	3 1 1 1 1 1 1 1
197	3 1 1 1 1 2 1 1 1 1 1 1
198	3 1 1 1 2 2 1 2 2 1 1 2 2 1 1 2 2 2 1 2 2 2 2 1
199	3 1 1 1 2 1 1 2 1 1 2 2 2 2 1 2 1 2 1 2 2 2 1
200	3 1 1 1 2 1 2 1 1 1 2 1 1 1 1 2 1 1 2 1 2
201	3 1 1 1 1 1 2 1 1 2 1 1 2 1 1 1 2 1 1
202	3 1 1 1 1 2 1 2 1 2 1 1 1 2 1 2 1 1 2 2 2 2
203	3 1 1 2 1 2 1 1 2 2 1 1 2 2 1 1 2 2 2 2 2 2 1 2 1
204	3 1 2 1 2 2 1 1 2 2 2 1 2 1 2 2 1 1 2 2 1 1 3
205	3 1 1 2 1 1 1 1 1 1
206	3 1 1 2 1 2 1 1 2 1 1 2 1 1 2 1 2 1 1 1 2
207	3 1 1 1 1 1 1 1
208	3 1 1 2 1 2 1 2 1 1 1 2 2 1 2 1 1 2 1 2 1 1 2 2
209	3 1 1 1 1 1 1 1
210	3 1 2 1 1 2 1 1 1 1 2 1 2 1 1 2 1 1
211	3 1 1 2 1 2 1 1 1 1 2 2 1 2 1 1 2 2 1 2 1 2 2 2 2
212	3 1 1 2 1 1 1 1 1 1
213	3 1 1 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
214	3 1 1 2 1 1 1 2 1 1 2 1 1 1 2 1
215	3 1 1 2 1 1 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2
216	3 1 1 2 1 1 1 1 1 1
217	3 1 1 1 2 2 1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 2
218	3 1 1 2 1 1 1 2 1 1 1 1 1 2 1 2 1 2
219	3 1 1 2 1 1 1 1 1 2 2 2 1 2 1 2 1 1 1
220	3 1 1 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2
221	3 1 1 2 1 1 1 1 1 1
222	3 1 1 1 2 1 1 1
223	3 1 1 2 1 2 1 2 1 2 2 1 1 2 1 2 2 2 2 2 2 1 1 3
224	3 1 1 2 1 1 2 1 2 1 2 1 1 2 2 2 1 2 1 2 2 1
225	3 1 1 2 1 1 1 1 1 2 1 1 3 2 2 1 2 1 1

Knoten	
226	3 1 1 2 1 1 2 1 2 2 1 2 1 2 1 2 2 1 1 2 2 1 1 2 2 2 2
227	3 1 1 2 1 1 2 2 2 1 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2
228	3 1 1 2 1 1 1 1 1 1
229	3 1 1 2 1 1 1 1 1
230	3 1 1 2 1 1 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 1 2
231	3 1 1 2 1 1 2 2 2 1 2 1 1 1 1 2 1 1 2 2 2 2 1
232	3 1 1 1 2 1 1 1 1 2 1 1 1 1
233	3 1 1 2 2 1 1 1 1 2 2 1 1 2 1 1 1 1
234	3 1 1 2 1 2 2 1 2 1 2 2 1 1 2 2 1 1 2 2 2 1 2 1
235	3 1 1 2 1 1 1 1 1 1
236	3 1 1 1 1 1 1 1
237	3 1 1 1 1 1 1 1
238	3 1 1 1 2 1 2 2 2 2 1 1 2 2 2 2 1 1 2 2 2 2 2
239	3 1 1 1 2 2 1 1 2 1 1 2 1 1 2 1 1 2 1 1 2 2 2 2 1
240	3 1 1 1 2 1 1 1 1 1 2 1 1 2 1 1
241	3 1 1 2 1 2 1 1 1 2 1 1 2 1 1 1 1 2 2 1 1
242	3 1 1 1 1 2 2 2 2 2 2 2 2 1 1 2 1 2 2 2 2 1 2 2 2
243	3 1 1 2 1 1 1 1 1 1
244	3 1 1 1 1 1 1 1
245	3 1 1 1 1 1 1 1
246	3 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
247	3 1 1 1 1 2 1 1 2 1 1 2 1 1 2 2 2 2 2 1 1 2 2 2 1 2
248	3 1 1 1 2 1 2 1 1 2 1 1 2 1 1 1 1 1 1 1 2
249	3 1 1 1 2 1 1 1 1 1 1 1 1 2 1 2 1
250	3 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1
251	3 1 1 1 1 2 1 2 1 1 2 1 1 1 1 2 1 1 1 1 2 1
252	3 1 1 2 1 2 1 1 2 2 1 2 1 1 1 2 2 2 2 1 1 2 2 1 2 2
253	3 1 1 2 1 2 2 1 1 2 2 2 1 1 2 2 2 2 2 2 2 1 1 3 2
254	3 1 1 1 1 1 2 1 1 1 1 1
255	3 1 2 1 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2
256	3 1 1 2 1 1 1 1 1 1
257	3 1 1 2 1 2 1 2 1 1 2 2 1 2 1 1 2 2 1 2 1 1 2 2 1
258	3 1 1 1 2 2 1 1 2 2 1 1 2 2 2 2 2 1 2 1 1 3 2
259	3 1 1 2 1 2 1 1 2 2 1 1 2 2 2 2 2 1 2 1 1 2 2 2 2 2
260	3 1 1 1 1 1 1 1
261	3 1 2 1 1 1 2 1 1 1 2 1 2 1 1 1 3
262	3 1 1 2 1 2 2 1 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 1
263	3 1 1 2 1 1 1 1 1 1
264	3 1 1 2 1 1 2 1 1 2 1 1 2 1 1 2 1 1 1 1 1 1
265	3 1 1 2 1 1 2 1 2 2 1 1 2 2 1 2 1 1 1 1 2 2 1 1 3 2
266	3 1 1 2 1 1 1 1 1 1
267	3 1 1 1 2 1 2 2 1 1 2 2 2 2 2 2 2 2 2 2 1
268	3 1 1 2 1 2 1 2 2 1 1 2 1 1 2 1 2 1 2 1 1 1
269	3 1 1 2 1 2 1 2 1 2 1 2 1 1 2 1 2 1 2 2 1
270	3 1 1 2 1 1 2 1 1 1 2 2 1 1 2 2 1 1 1 2 2 2 2 1 1 3

Knoten	
271	3 1 1 2 1 1 1 1 1 1
272	3 1 1 1 1 1 1 1
273	3 1 1 2 1 2 1 2 2 2 2 1 2 2 2 1 2 2 2 1 1 2 2
274	3 1 1 2 1 1 2 1 2 1 1 2 1 1 2 1 1 2 1 1 2 1 1 2
275	3 1 1 2 1 1 1 2 1 1 1 1 1 1 2 1 1
276	3 1 1 2 1 2 1 1 2 1 1 2 1 1 1 1 2 2 1 2 1 1 2 1 1 1 1
277	3 1 1 2 1 1 2 2 1 1 2 2 2 2 2 2 1 2 2 1 1 2 2 2 2 1 2
278	3 1 1 2 1 1 1 1 1 1 1
279	3 1 1 1 1 1 1 1 1
280	3 1 1 2 1 1 2 2 1 1 2 2 2 2 2 2 1 2 2 2 1 2 2 2
281	3 1 1 2 1 1 2 1 2 1 1 2 1 2 2 1 1 1 1 2 2 1
282	3 1 1 1 1 1 2 2 1 1 2 1 1 1 1 2 1 1
283	3 1 1 2 1 1 2 2 2 2 1 1 2 1 2 1 1 2 2 1 1 1 2 2 2 2 1
284	3 1 1 2 1 1 1 1 1 1 1
285	3 1 1 1 1 1 1 1 1
286	3 1 1 1 1 1 1 1 1 1
287	3 1 1 1 2 1 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
288	3 1 1 1 2 1 2 2 1 2 1 1 1 2 2 1 2 1 2 1 1
289	3 1 1 1 1 1 2 1 1 2 1 1 1 2 1 2 1 2
290	3 1 1 1 1 2 1 2 1 1 2 1 1 2 1 1 2 1 1 2
291	3 1 1 2 1 1 2 2 1 2 2 1 2 2 1 1 2 2 2 1 2 1 1
292	3 1 1 1 1 1 1 1 1
293	3 1 1 1 1 1 1 1 1
294	3 1 1 1 1 1 1 1 1
295	3 1 1 1 2 1 1 1 1 2 2 2 2 1 1 1
296	3 1 1 1 1 2 2 2 2 1 2 2 1 2 2 1 2 2 1 1 2 2 2 2 2
297	3 1 1 1 2 1 1 2 1 1 2 2 2 2 1 1 2 2 1 2 1 2 1 2 1
298	3 1 1 1 2 1 2 1 2 1 2 1 1 2 2 1 1 1 1
299	3 1 1 1 1 2 1 1 2 1 1 1 2 1 1 1
300	3 1 1 2 2 1 2 1 1 2 1 1 1 2 2 1 1 2 1 2 1
301	3 1 1 1 1 2 1 1 2 2 1 1 2 2 2 1 1 1 1 2 2 1 2 1
302	3 1 1 2 1 2 2 1 2 2 2 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2
303	3 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
304	3 1 1 1 1 1 1 1 1
305	3 1 1 1 1 1 1 1 1 1
306	3 1 2 1 1 1 2 2 2 2 2 2 1 2 2 2 2 1 2 2 2 2 2
307	3 1 1 2 1 1 1 1 1 1 1
308	3 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1 1 2 2
309	3 1 2 1 1 2 1 2 1 1 1 2 1 2 1 1 1 1 2 1 2 1
310	3 1 1 2 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1 2 2
311	3 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 2 2
312	3 1 1 2 1 1 1 2 2 1 2 1 2 1 1 2 1 2 1 1 2 1 1 2
313	3 1 1 2 1 1 1 1 1 1 1
314	3 1 1 1 2 2 1 1 2 2 2 2 2 2 1 1 2 2 2 1 1
315	3 1 1 2 1 2 1 2 1 2 1 2 1 2 1 1 1 2 1

Knoten	
316	3 1 1 2 1 1 1 2 2 2 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2
317	3 1 1 1 1 1 1
318	3 1 1 1 2 2 2 2 1 1 1 2 2 2 2 1 2 1 1 3 2 1
319	3 1 1 2 1 1 2 1 1 2 1 1 1 2 1 1 1
320	3 1 1 2 1 1 2 2 1 2 2 1 1 2 1 2 1 1 2 2 1 2 1 2 2
321	3 1 1 2 1 1 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
322	3 1 1 2 1 1 1 1 1
323	3 1 1 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
324	3 1 1 2 1 1 2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 3 1
325	3 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1
326	3 1 1 2 1 1 2 1 2 2 1 1 1 1 2 2 1 1 1 1 2 1 2 1 2
327	3 1 1 2 1 1 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
328	3 1 1 2 1 1 1 1 1 1
329	3 1 1 1 1 1 1 1 1
330	3 1 1 2 1 1 2 2 2 2 1 1 2 2 2 1 1 2 2 2 1 2 2 2 2
331	3 1 1 1 1 2 1 2 1 1 1 2 1 1 2 1 2 1 1 1 1
332	3 1 1 2 1 2 1 1 2 1 1 2 1 1 1 2 1 2 2 1 2 1
333	3 1 1 2 1 1 1 2 2 2 2 2 2 2 1 2 2 2 2 2 1 2 2 2 1 1
334	3 1 1 1 1 1 1 1 1
335	3 1 1 1 1 1 1 1 1
336	3 1 1 1 2 2 1 2 1 2 2 2 2 2 2 2 2 2 2 2 1 2 2
337	3 1 1 1 2 1 1 2 1 1 1 2 1 1 2 1 1 2 2 1 1 3 2 1 2 1
338	3 1 1 1 2 1 2 1 1 1 2 1 1 2 1 2 1 2 1 2 1
339	3 1 1 1 1 1 1 2 1 2 1 2 1 1 1 1
340	3 1 1 1 1 2 1 1 1 1 2 2 1 2 1 1 2 2 1 1 2 1
341	3 1 1 1 2 1 2 2 2 1 1 2 2 2 1 2 2 1 1 2 2 2 1
342	3 1 1 1 1 1 1 1 1
343	3 1 1 1 1 1 1 1 1
344	3 1 1 1 1 2 1 1 1 1
345	3 1 1 1 2 1 2 1 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1
346	3 1 1 1 2 2 2 2 1 2 2 2 2 1 2 1 2 1 1 2 2 1 1
347	3 1 1 1 2 1 2 1 1 2 1 1 2 1 1 1 2 1 2 1 1 2
348	3 1 1 1 2 1 1 1 1 2 1 1 2 1 1 1 1 1
349	3 1 1 1 1 1 1 2 1 1 2 1 1 1 1 1 2
350	3 1 1 1 1 2 1 2 2 2 1 2 1 1 1 2 2 1 1 1 1 1 3 1
351	3 1 1 2 1 2 1 2 1 2 1 1 1 1 2 1 1 2 2 2 2 1 1 1
352	3 1 1 1 1 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 1 2 2 2
353	3 1 1 1 1 1 1 1 1
354	3 1 1 1 2 1 1 1 1 1
355	3 1 1 1 1 1 1 1 1
356	3 1 1 1 2 1 1 1 1 1
357	3 1 2 1 1 1 1 1
358	3 1 1 1 1 1 1 1 1
359	3 1 1 2 1 1 1 1 1 1
360	3 1 2 1 1 1 1 1 1 2 1 2 1 1 2 1 1

Knoten	
361	3 1 1 2 1 1 1 1 1
362	3 1 1 1 2 2 2 2 1 2 1 2 1 2 1 1 2 2 1 1 2 2 1 2
363	3 1 1 2 1 1 2 2 2 2 1 2 2 1 2 2 1 1 3 2 2 1 2 1 1 3 2
364	3 1 1 1 1 1 3 1 1
365	3 1 1 2 1 1 1 1 1 2 1 2 1 1 1 2 2 1
366	3 1 1 2 1 2 1 2 1 2 1 2 2 1 1 2 1 1 1 2 1 1 1 2
367	3 1 1 2 1 1 1 1 1 1
368	3 1 1 1 2 1 1 1 1 1
369	3 1 1 2 1 1 1 2 1 2 1 1 1 2 2 2 2 1 2 1 2 1 2
370	3 1 1 2 1 1 1 2 1 1 1 2 1 1 1 1 1 1 1
371	3 1 1 2 1 2 2 2 2 1 2 2 2 2 2 1 2 2 2 2 1 1 2 2 1 1
372	3 1 1 2 1 1 1 1 1 1
373	3 1 1 2 1 1 1 1 1 1
374	3 1 1 2 1 1 2 2 2 2 1 1 2 2 1 2 2 2 2 1 1 2 2 1 2 2
375	3 1 1 2 1 1 2 1 1 1 2 1 1 2 1 1 1 2
376	3 1 1 2 1 2 1 1 2 1 1 2 1 1 2 1 2 1 1 1 2 1 2 2 1 2
377	3 1 1 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
378	3 1 1 2 1 1 1 1 1 1
379	3 1 1 1 2 1 1 1 1 1
380	3 1 1 1 2 1 2 2 1 1 2 2 1 1 2 2 2 2 1 1 2 2 1 2 2
381	3 1 1 1 1 2 1 1 2 1 1 2 1 1 1 1 2 1 1
382	3 1 1 1 1 1 1 2 2 1 2 1 2 1 1 2 2 1 1 2 2
383	3 1 1 2 1 1 1 1 1 1
384	3 1 1 1 1 1 1 1 1
385	3 1 1 1 1 2 1 1 1 1 1 1
386	3 1 1 1 2 1 1 2 2 2 1 2 2 2 2 1 2 2 2 2 2 2 2
387	3 1 1 1 2 1 1 1 2 1 1 2 1 1 2 1 1 1 2 1 2 1
388	3 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1
389	3 1 1 2 2 1 2 2 1 1 2 1 1 1 2 2 1 2
390	3 1 1 2 1 1 2 1 2 2 1 1 2 2 2 2 1 1 2 2 2 2 1 1 2 2 2 2
391	3 1 1 2 1 1 2 2 2 2 2 1 2 2 2 2 2 2 2 2 1 1 2 2 2 2
392	3 1 1 2 1 1 1 1 1 1 1
393	3 1 1 1 2 1 1 1 1 1
394	3 1 1 1 1 1 1 1 1
395	3 1 1 1 2 2 2 2 2 1 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2
396	3 1 1 1 2 1 1 2 2 1 1 2 1 2 2 1 1 1 2 1 2 2 1 2
397	3 1 1 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 1 2 1
398	3 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 2 1 1
399	3 1 1 2 2 1 2 1 1 2 1 2 1 1 1 2 2 1 2 1 2
400	3 1 1 1 1 2 2 1 1 2 1 1 1 2 1 2 2 1 1 1 2 1
401	3 1 1 1 2 1 2 1 2 2 1 1 2 2 2 1 1 2 2 2 1 1 2 2
402	3 1 1 1 1 1 2 2 2 2 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2
403	3 1 1 1 2 1 1 1 1 1
404	3 1 1 1 1 1 1 1 1 1
405	3 1 1 1 1 1 1 1 1 1

Knoten	
406	3 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
407	3 1 1 1 2 1 2 2 1 1 2 2 1 2 2 2 2 1 2 1 2 1 2 1 2
408	3 1 1 2 1 1 1 1 1 1
409	3 1 1 1 1 1 1
410	3 1 1 2 1 1 1 1 1 1
411	3 1 2 1 1 1 2 1 1 1 1 2 1 1 1 2 1 1 3
412	3 1 1 2 1 1 1 1 1 1
413	3 1 1 2 1 1 2 1 2 1 1 2 2 1 1 1 2 1
414	3 1 1 2 1 1 1 1 1 1
415	3 1 1 1 2 2 1 2 1 1 2 2 2 2 2 2 2 2 2 2
416	3 1 1 2 1 1 2 1 1 2 1 1 1 2 1 1
417	3 1 1 2 1 2 2 2 2 1 2 1 2 2 1 1 1 2 2 2 2 2 1
418	3 1 1 2 1 1 1 1 1 1
419	3 1 1 1 2 2 2 2 2 2 2 1 2 2 1 1 2 2 2 2 2 2 2 2
420	3 1 1 2 1 1 2 1 2 1 1 1 2 1 1 2 1 1 2 1
421	3 1 1 2 1 2 1 2 1 2 1 1 1 2 1 2 1 1 1 3 1 1 2 2 2
422	3 1 1 2 1 1 1 2 2 2 1 2 2 2 2 1 2 2 2 2 2 2 2 2 2 2
423	3 1 1 1 2 1 1 1 1 1 1
424	3 1 1 2 1 1 2 2 2 2 2 2 2 2 2 1 1 2 2 2 2 1 2 2 2
425	3 1 1 2 1 1 2 1 2 1 1 2 1 1 1 2 1 1 3 1
426	3 1 1 2 1 2 1 1 2 1 1 2 1 1 2 1 1 2 1 2 1 1 1 2
427	3 1 1 2 1 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2
428	3 1 1 2 1 1 1 1 1 1
429	3 1 1 2 1 1 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2
430	3 1 1 2 1 1 2 1 1 2 2 2 2 1 1 1 2 2 1 2 1 2
431	3 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1
432	3 1 1 2 1 2 1 2 1 1 1 1 2 2 1 1 2 2 1 1 2 2 1 1 3 2
433	3 1 1 2 1 1 1 1 1 1 1
434	3 1 1 2 1 1 1 1 1 1 1
435	3 1 1 1 2 1 1 1 1 1 1
436	3 1 1 2 2 1 2 2 2 1 1 1 2 2 2 2 1 2 2 1 1
437	3 1 1 1 2 1 2 1 1 2 1 1 2 1 1 2 1 1 2 1 1 2
438	3 1 1 2 1 1 2 1 1 1 2 1 2 1 1 1
439	3 1 1 2 1 1 2 2 1 1 2 1 1 2 1 2 1 1 1 1 1
440	3 1 1 2 1 2 1 1 2 1 1 2 2 1 1 1 2 2 2 2 1 1 2 2 1 2
441	3 1 1 2 1 1 2 2 2 1 2 2 1 2 2 2 2 1 1 2 2 2 2 2 2 2
442	3 1 1 1 1 1 1 1 1
443	3 1 1 1 1 1 1 1
444	3 1 1 1 2 1 1 1 1 1 1
445	3 1 1 1 2 1 1 2 1 1 2 2 2 2 2 2 1 2 1 1 2 2 2 1
446	3 1 1 1 2 2 1 2 2 1 1 2 1 2 2 1 2 1 2 1 1 1
447	3 1 1 1 1 2 1 2 1 2 1 2 1 1 1 1 1 1
448	3 1 1 1 1 1 1 1 2 1 2 1 1 1 2 1
449	3 1 1 2 2 1 1 1 2 1 1 2 1 1 2 2 2 2 1 1 2 1 1 1
450	3 1 1 2 1 2 1 1 1 2 2 1 1 2 2 1 1 2 2 2 1 2 1 2 2 1

Knoten	
451	3 1 1 2 1 1 1 1 2 2 2 2 2 2 2 2 1 2 2 2 2 2 1 1 2 2
452	3 1 1 1 2 1 1 1 1 1
453	3 1 1 1 1 1 1
454	3 1 1 1 1 1 1 1 1
455	3 1 1 1 2 1 2 2 2 1 2 2 1 2 2 2 2 2 2 2 2 2 1
456	3 1 1 1 2 2 1 1 2 2 1 2 1 1 3 1 1 2 1 1 2 2 2 2 1
457	3 1 1 1 2 1 1 1 1 1 2 1 1 2 1 1
458	3 1 1 1 2 1 1 2 1 2 1 1 2 1 1 2 1 1
459	3 1 1 2 1 1 1 2 1 1 1
460	3 1 1 2 1 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 1
461	3 1 1 1 1 1 1 1 1
462	3 1 1 2 1 2 1 2 1 1 2 1 1 1 2 2 1 1 3 1 1 2
463	3 1 1 2 1 1 1 1 1 1
464	3 1 1 2 1 2 1 2 1 1 2 1 2 1 1 1 2 1 1
465	3 1 1 2 1 1 1 1 1 1
466	3 1 1 1 2 2 2 2 1 1 2 2 2 2 1 2 1 2 1 1 2 2 1 2
467	3 1 1 2 1 2 1 1 2 1 2 1 1 2 1 2 1 2 1 1 2 1
468	3 1 1 2 1 1 1 2 2 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2
469	3 1 1 1 2 1 1 1 1 1
470	3 1 2 1 1 1 1 2 1 1 1 2 1 1 1 2 1 1 1 2 1 1 3
471	3 1 1 2 1 1 1 1 1 1 1 1 2 1 1 2 1 2 1
472	3 1 1 2 1 1 2 2 1 1 2 2 1 2 1 1 2 2 2 1 1 2 2 2 2 2 1
473	3 1 1 2 1 1 1 1 1 1 1
474	3 1 1 2 1 1 2 1 1 1 1 1 1
475	3 1 1 2 1 1 2 1 2 1 2 1 2 1 1 2 1 2 1 2 1 1 3
476	3 1 1 2 1 2 1 1 1 2 1 2 2 1 2 1 2 1 2 1 2 1 1
477	3 1 1 2 1 1 1 1 1 1 1
478	3 1 1 1 1 1 1 1 1
479	3 1 1 1 2 1 1 2 2 2 1 1 2 2 2 2 2 2 1 2 2 2 1
480	3 1 1 1 2 1 2 1 1 2 1 1 2 1 1 2 1 1 2 1 1 1
481	3 1 1 1 1 2 1 1 2 1 2 1 2 1 2 1 1 3 1 1 1
482	3 1 1 2 1 2 1 1 1 2 2 2 2 1 2 1 1 1 2 1 2 2 1 1 1
483	3 1 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
484	3 1 1 1 1 1 1 1 1
485	3 1 1 2 1 1 1 1 1
486	3 1 1 1 2 2 2 2 1 2 1 1 2 2 2 1 2 2 2 2 1 2 2 1
487	3 1 1 1 2 1 2 1 1 1 2 2 1 1 2 1 1 2 2 1
488	3 1 1 1 1 2 1 2 1 1 1 2 1 1 1
489	3 1 1 1 1 2 1 1 1 2 2 2 1 1 1 2 1 2 1 1
490	3 1 1 2 1 2 2 1 2 1 1 2 1 1 2 2 2 2 1 2 1 1 1 1 2 2 2
491	3 1 1 2 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
492	3 1 1 1 1 1 1 1 1 1
493	3 1 1 1 1 1 1 1 1
494	3 1 1 1 2 1 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2 2 1
495	3 1 1 1 2 2 1 1 1 1 2 2 1 2 1 2 1 2 1 1 2 1

Knoten	
90	1 3 4 3 1 2 4 2 1 2 4 2 1 2 4 3 1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 3 1 3...
91	1 3 4 3 1 2 4 3 1 2 4 3 1 2 4 2 1 2 4 2 1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 3
92	1 3 4 3 1 2 1 2 4 2 1 2 4 2 1 3 4 3 1 2 4 2 1 3 4
93	1 3 4 3 1 3 4 3 1 3 4 2 1 2 1 2 4 2 1 3 4
94	1 3 4 3 1 3 4 2 1 3 4 2 1 2 4 2 1 2 4 2 1 2 4 2 1 2 4 2
95	1 3 4 3 1 3 4 3 1 3 4 2 1 2 4 2 1 3 4 3 1 3 4 3 1 2 4 2 1 3 4
96	1 3 4 3 1 3 4 2 1 2 4 2 1 3 4 2 1 3 4 3 1 2 4 3 1 3 4 3 1 2
97	1 3 4 3 1 3 4 2 1 3 4 3 1 2 1 2 1 3 1 2 4 3 1 3 4 3 1 3 4 3 1 3
98	1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 2 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 3 1 2 4 3 1 2 4 3 1 2 4 3 1 2 4 3 1 2 4 3 1 2...
99	1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4
100	1 3 4 3 1 2 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4
101	1 3 4 3 1 2 4 2 1 3 4
102	1 3 4 2 1 2 1 3 4 3 1 3 4 3 1 3 4 3 1 2 4 3
103	1 3 4 2 1 3 4 2 1 3 4 3 1 2 4 3 1 3 4 2 1 3 4 3 1 2 4 3 1 2 4 3
104	1 3 4 2 1 2 4 2 1 2 4 2 1 2 4 2 1 2 4 2 1 3 4 2
105	1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4
106	1 3 4 3 1 2 4 3 1 2 4 2 1 3 4 2 1 3 4
107	1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4
108	1 2 4 2 1 3 4 2 1 3 4 3 1 3 4 3 1 2 4 2 1 2 4 2 1 2 4 3 1 3 4 3 1 2 4 2
109	1 3 4 3 1 2 1 2 1 3 4 3 1 3 4 3 1 3 4 3 1
110	1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 3 1 3 4 2 1 2 4 2 1 3 4 3 1 2 1 3 1 3
111	1 3 4 2 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4
112	1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 3 1 3 4 2 1 3 4 3 1 2 4 2 1 2
113	1 3 4 3 1 2 1 2 1 2 1 2 1 3 4 2 1 3 4 2 1 2 4 3
114	1 3 4 3 1 3 4 3 1 2 4 3 1 2 4 2 1 3 4 3 1 3 4 3 1 3 4 2 1 3 4
115	1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 2 1 3 4 2 1 3 4 3 1 3 4 3 1 2 4 3 1 2 4 2
116	1 3 4 2 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4
117	1 3 4 3 1 2 4 2 1 2 4 3 1 3 4 2 1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 3 1 2 4 3 1 2 4 3 1 2 4 3 1 2 4 2 1 2 4 2 1 2 4 2
118	1 3 4 3 1 2 1 3 4 2 1 3 4 2 1 2 4 2 1 3 4 3 1 3 4 3 1
119	1 3 4 3 1 3 4 3 1 2 4 2 1 3 4 3 1 3 4 3 1 3 1 3 1
120	1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 2 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 2 1 3 1 3
121	1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1
122	1 3 4 3 1 3 4 2 1
123	1 3 4 3 1 2 4 2 1 2 4 2 1 3 4 2 1 3 4 3 1 3 4 3 1
124	1 3 4 3 1 2 1 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 2
125	1 3 4 3 1 3 4 2 1 2 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1
126	1 3 4 3 1 3 4 2 1 2 4 2 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 2 4 3 1 3

Knoten	
244	1 2 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1
245	1 2 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 2 4
246	1 2 4 3 1 2 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 3 1 2 4 2 1 2 4 2 1 2 4 2 1
247	1 2 4 3 1 2 4 2 1 2 4 3 1 3 4 3 1 3 4 3 1 2 4 2 1 2 4 2 1 2 4
248	1 2 4 3 1 2 4 2 1 3 4 3 1 3 4 3 1 2
249	1 2 4 3 1 2 1 3 4 3 1 3 4 2 1 2 1 2 4 2 1 3 4 2 1 2
250	1 2 4 3 1 3 4 3 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 2 1
251	1 2 4 3 1 3 4 3 1 2 4 3 1 3 4 3 1 2 4 2 1 3 4 3 1 3 4 2
252	1 2 4 3 1 3 4 2 1 2 4 3 1 3 4 2 1 2 1 2
253	1 2 4 3 1 3 4 2 1 3 4 3 1 3 1 3 1 2 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 4 3 1
254	1 2 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1
255	1 3 4 2 1 3 4 2 1 3 4 3 1 2 4 2 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4
256	1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1
257	1 3 4 3 1 3 4 3 1 2 4 2 1 2 4 2 1 3 4 3 1 2 4 2 1 3 4 2 1
258	1 3 4 3 1 3 1 2 4 2 1 3 4 2 1
259	1 3 4 3 1 3 4 2 1 2 4 3 1 3 4 2 1 2 4 2 1 3 1
260	1 3 4 3 1
261	1 3 4 3 1 2 1 2 4 2 1 2 4 2 1 2 4 2 1 2 4 2 1 2 4 2 1 3 4 2 1
262	1 3 4 3 1 3 4 2 1 3 4 2 1 3 1 3 1 3 1 3 1 2 1 3
263	1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4
264	1 3 4 3 1 3 4 3 1 2 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1
265	1 3 4 3 1 3 4 2 1 2 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 2 4 3 1 3 4
266	1 3 4 2 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4
267	1 2 4 2 1 2 4 2 1 2 4 2 1 2 4 2 1 2 4 2 1 2 4 2 1 2 4 2 1 2 4 2 1 2 4 2 1 2 4 2 1 2 4 2 1 2 4 2
268	1 3 4 3 1 2 4 3 1 3 4 3 1 3 4 2 1 2 4 2 1 3
269	1 3 4 3 1 2 1 3 4 3 1 2 1 3 4 3 1 2 1 3 4 3 1 3 4 3 1
270	1 3 4 3 1 3 4 2 1 2 4 2 1 3 4 2 1 3 4 3 1 2 4 2 1 3 4 3 1 3 4 2
271	1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4
272	1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4
273	1 3 4 3 1 2 4 2 1 2 4 3 1 3 4 2 1 3 4 3 1 3 4 3 1 2 4 2 1 3 4 2 1 3 4
274	1 3 4 3 1 2 4 2 1 3 4 2 1 2 4 3 1 2 1 2 1 2 1 2 1
275	1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 2 1 2 4 3 1 3 4 2 1 2 4 3
276	1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 2 1 3 4 3 1 2 1 3 4 2
277	1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 2 4 3 1 3 4 3 1 3 4 3 1 2 4 2 1 3 4 3 1 2 4 2
278	1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4
279	1 3 4 3 1 2 4 2 1 3 4
280	1 3 4 3 1 2 4 2 1 2 4 3 1 2 4 3 1 2 4 2 1 2 4 2 1 3 4 3 1 2 4 3 1 2
281	1 3 4 3 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 2 4 2 1 3 4 3 1
282	1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 2 1 2
283	1 3 4 3 1 3 4 2 1 3 4 2 1 2 4 2 1 3 4 2 1 3 4 3 1 3 4

Knoten	
322	1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4
323	1 3 4 3 1 2 4 2 1 2 4 2 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 3 1 2 4 3 1 2 4 3 1 2 4 3 1 2 4 3 1 2 4 3 1 2 4 3 1 2 4 3 1 2 4 3 1 2 4 3 1 2 4 2 1 2 4 2 1 2
324	1 3 4 3 1 2 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 2 4 2
325	1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 2 1 2 4 2 1 3 4 2 1 2 1 2 1 3
326	1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 2 1 3 4 3 1 2 4 2 1 3
327	1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 3 1 3 4 2 1 2 4 3 1 3 4 3
328	1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4
329	1 3 4 3 1 2 4 2 1 3 4 2 1
330	1 3 4 3 1 2 1 3 4 3 1 3 4 2 1 2 4 2 1 2 4 2 1 3 4 2 1 2
331	1 3 4 3 1 3 4 2 1 3 4 2 1 2 4 2 1 3 4 2
332	1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 3 1 3 4 2 1 3 4 3 1 2
333	1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 2 4 3 1 3 4 3 1 2 4 2 1 2 4
334	1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4
335	1 2 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 2 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1
336	1 2 4 3 1 2 4 2 1 2 4 3 1 3 4 2 1 3 4 3 1 3 4 3 1 2 4 3 1 2 4 3 1 2 4 2 1 2 4 2 1 2 4 2 1 2 4 2
337	1 2 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 2 4 3 1 3 4 3 1 2
338	1 2 4 3 1 2 1 3 4 3 1 3 4 3 1 3 4 2 1 3 4 2 1 2 1 3 4
339	1 2 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 2 1 3 4 3 1 2 4 2 1
340	1 2 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 2 1 3 4 2 1 3 1 3 4 2
341	1 2 4 3 1 3 4 2 1 3 4 2 1 2 1 3 1 3 1 3 1 3 1 2 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3
342	1 2 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4
343	1 2 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 2 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1
344	1 2 4 3 1 2 4 2 1 3 4
345	1 2 4 3 1 2 4 2 1 3 4 3 1 2 4 3 1 3 4 3 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 2 4 2 1 2
346	1 2 4 3 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 3 1 3 4 2 1 2 4
347	1 2 4 3 1 2 1 2 1 2 1 2 1 2 1 3 4 3 1 3 4 3 1 3
348	1 2 4 3 1 3 4 3 1 2 4 2 1 2 1 2 1 3 4 3 1 2 4 2 1
349	1 2 4 3 1 3 4 2 1 2 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1
350	1 2 4 3 1 3 4 3 1 2 4 3 1 2 1 3 4 3 1 2 4 2 1 3 4
351	1 2 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 3 1
352	1 2 4 3 1 3 4 2 1 3 4 3 1 3 4 3 1 2 4 2 1 2 4 2 1 2 4 2 1 3 4 3 1 3 4 2 1 2 4
353	1 2 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4
354	1 2 4 3 1 2 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4
355	1 2 4 3 1 2 4 3 1
356	1 2 4 3 1 3 4 3 1
357	1 3 4 2 1 3 4
358	1 3 4 2 1 3 4 2 1
359	1 3 4 2 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1
360	1 3 4 3 1 3 4 2 1 3 4 2 1 3 1 2 4 2 1

Knoten	
361	1 3 4 2 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1
362	1 3 4 2 1 3 4 3 1 2 4 2 1 3 4 2 1 3 4 2 1 3
363	1 3 4 3 1 3 4 2 1 2 4 2 1 3 4 3 1 3 4 3 1 2 4 2 1 3 4 3 1 2
364	1 3 4 2 1 3 4
365	1 3 4 3 1 2 4 3 1 3 4 3 1 2 4 2 1 3 4 3 1 3 4
366	1 3 4 3 1 3 4 3 1 2 4 2 1 2 4 3 1 3 4 2 1 3 4 2 1 3
367	1 3 4 3 1 3 4 2 1 2 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1
368	1 3 4 2 1
369	1 3 4 3 1 2 4 2 1 3 4 3 1 3 4 2 1 3 4 3 1 2 4 2 1 2 4 3
370	1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 3 1 2 1 3 4 3 1 2 4 2 1 3
371	1 3 4 3 1 3 4 2 1 3 1 2 4 3 1 3 4 2 1 3 4 2 1 3 4 3 1 2 4 3 1 2 4
372	1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4
373	1 3 4 3 1 3 4 2 1 3 4 3 1
374	1 3 4 3 1 2 4 3 1 2 4 2 1 2 4 2 1 3 4 2 1 3 4 3 1 3 1 3 1
375	1 3 4 3 1 3 4 3 1 3 4 2 1 2 1 3 4 2 1 2 4 2 1 3 4 2 1 3
376	1 3 4 3 1 3 4 2 1 3 4 3 1 2 4 2 1 2 4 3 1 3 4 3 1 2 4 2 1
377	1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 3 1 2 4 3 1 3 4 2 1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 3 1 3 4 3 1 2 4 2 1 3 4 2
378	1 3 4 3 1 3 4 2 1 2 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4
379	1 3 4 3 1 3 4 2 1 3 4 2 1
380	1 3 4 3 1 2 1 2 4 2 1 3 1 3 1 3 4 2 1 2 1 3
381	1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 3 1 2 4 2 1
382	1 3 4 3 1 3 4 2 1 2 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 2 4 3 1
383	1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4
384	1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 3 1 3 4 2 1
385	1 3 4 3 1 2 4 2 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1
386	1 3 4 3 1 2 4 2 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 2 1 3 1 3 1 2 1 2
387	1 2 4 3 1 2 4 2 1 2 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 2
388	1 2 4 3 1 2 1 3 4 2 1 2 4 2 1 3 4 2 1 2 1 3 4 2 1 2 4 2 1 3 4 3
389	1 2 4 3 1 3 4 2 1 3 4 2 1 3 4 3 1 3 4 3 1 3 4 3
390	1 3 4 3 1 3 4 2 1 2 4 2 1 2 4 2 1 3 4 3 1 3 4 2 1 3 4 3
391	1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 3 1 2 4 3 1 3 4 2 1 3 4 2 1 3 4 3 1 3 4 3 1 3 4 3
392	1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4
393	1 3 4 3 1 2 4 2 1 3 4
394	1 3 4 3 1 2 4 2 1 3 4
395	1 2 4 3 1 3 4 2 1 3 4 3 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 3 1 3 4 3 1 2
396	1 2 4 3 1 2 1 3 4 3 1 3 4 3 1 2 4 2 1 2 4 2 1 3 1 2 1
397	1 2 4 3 1 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 4 2 1 3 1 3
398	1 2 4 3 1 3 4 2 1 2 4 2 1 2 4 2 1 2 4 2 1 2 4 2 1 3 4 2 1 2
399	1 2 4 3 1 3 4 3 1 3 4 2 1 3 4 3 1 3 4 2 1 2 4 2 1 3 4 3 1
400	1 2 4 3 1 3 4 2 1 2 4 2 1 2 1 3 4 3 1 3 4 2 1 2 4 2 1

Anhang E

CD-ROM Inhalt

Auf der dieser Arbeit beigelegten CD-ROM befinden sich folgende Daten:

- **\Programm**
Hier befindet sich der Quellcode des dieser Arbeit zugrundeliegenden C++Programms.
- **\Arbeit**
In diesem Verzeichnis ist die Diplomarbeit als PDF-Datei *Arbeit.pdf* und PostScript-Datei *Arbeit.ps* zu finden. Es liegen folgende Unterverzeichnisse vor:
 - **\Daten**
In diesem Verzeichnis befinden sich die bei dokumentierten Testläufen erhaltene Daten.
 - **\Abbildungen**
Alle in der Arbeit aufgeführten Abbildungen im jpg- und eps-Format sind hier zu finden.
- **\Literatur**
Die im Literaturverzeichnis aufgeführten Internetquellen wurden hier abgelegt.

Literaturverzeichnis

- [1] Bauer F.: *Adaptive Spline-Approximation in der optimalen Steuerung*, Diplomarbeit, Mathematisches Institut, Universität Bayreuth, 2004
- [2] Brent R.P.: *Algorithms for Minimization Without Derivatives*, Dover Publications, Mineola, New York, 2002
- [3] Crenshaw Jack W.: *Ramblin' Jack*, Internetartikel, 2001
<http://www.embedded.com/showArticle.jhtml?articleID=9900084>
- [4] Crenshaw Jack W.: *Worth the Wait*, Internetartikel, 2001
<http://www.embedded.com/showArticle.jhtml?articleID=9900162>
- [5] Crenshaw Jack W.: *Looking at Clouds*, Internetartikel, 2001
<http://www.embedded.com/showArticle.jhtml?articleID=9900205>
- [6] Crenshaw Jack W.: *Making It Work*, Internetartikel, 2001
<http://www.embedded.com/showArticle.jhtml?articleID=9900284>
- [7] Crenshaw Jack W.: *Where Do We Stand?*, Internetartikel, 2001
<http://www.embedded.com/showArticle.jhtml?articleID=9900325>
- [8] Crenshaw Jack W.: *Black Tuesday*, Internetartikel, 2001
<http://www.embedded.com/showArticle.jhtml?articleID=9900357>
- [9] Gerdts M.: *Nichtdifferenzierbare Optimierung*, Vorlesung an der Universität Bayreuth, gehalten im Sommersemester 2003

-
- [10] Gonzáles R. L. V., Tidball M. M.: *On a Discrete Time Approximation of the Hamilton-Jacobi Equation of Dynamic Programming*, INRIA Rapports de Recherche Nr. 1375, 1991
- [11] Grüne L.: *Numerische optimale Steuerung und Stabilisierung*, Diplomarbeit, Institut für Mathematik, Universität Augsburg, 1994
- [12] Grüne L., Semmler W.: *Using Dynamic Programming with Adaptive Grid Scheme for Optimal Control Problems in Economics*, Journal of Economic Dynamics and Control, 28, 2427-2456, 2004
- [13] Grüne L.: *Numerische Dynamik von Kontrollsystemen*, Vorlesung an der Universität Bayreuth, gehalten im Sommersemester 2004
- [14] Grüne L.: *Error estimation and adaptive discretization for the diskrete stochastic Hamilton-Jacobi-Bellman equation*, Numerische Mathematik, 99, 85-112, 2004
- [15] Kropka H.: *LATEX: eine Einführung*, Addison-Wesley, 1991
- [16] Pohl I.: *C++ for C Programmers Second Edition*, The Benjamin/ Cummings Publishing Company, 1994
- [17] Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P.: *Numerical Recipes in C++ Second Edition*, Cambridge University Press, 2002
- [18] Recktenwald G.W.: *Numerical Methods with MATLAB*, Prentice-Hall, Upper Saddle River, New Jersey, 2000
- [19] Sontag E. D.: *Mathematical Control Theory*, Springer Verlag, New York, 2nd Edition, 1998
- [20] Schramm H.: *Eine Kombination von Bundle- und Trust-Region-Verfahren zur Lösung nichtdifferenzierbarer Optimierungsprobleme*, Bayreuther Mathematische Schriften, Heft 30, 1989

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe angefertigt habe. Die verwendete Literatur und sonstige Hilfsmittel sind vollständig angegeben. Die Arbeit wurde in dieser oder ähnlicher Form noch keiner Prüfungskommission vorgelegt.

Bayreuth, 6. Juli 2005

.....

Michael Jarczyk