Universität Bayreuth
Fakultät für Mathematik und Physik

Diplomarbeit

Numerische Lösung dynamischer Gleichgewichtsmodelle

STEPHANIE BECKER

05. Oktober 2005

Betreut durch
PROF. DR. LARS GRÜNE

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Die vorliegende Arbeit beschäftigt sich mit einem Verfahren aus dem Bereich der Perturbations-Methoden. Diese beinhalten Techniken zur lokalen Approximation auf Umgebungen um bestimmte Punkte. Da man die Umgebung dieses Punktes auch als Störung auffassen kann, werden diese Verfahren in deutscher Literatur auch als Störungsmethoden bezeichnet.

Diese Approximationstechnik findet Anwendung in den verschiedensten wissenschaftlichen Disziplinen wie den Natur- und Ingenieurswissenschaften, aber auch in der Ökonomie. Hier ist man insbesondere im Bereich der Makroökonomie an Gleichgewichtsmodellen interessiert, die untersuchen, wie sich Störungen von außen auf ein volkswirtschaftliches Gleichgewicht auswirken. Solche Störungen können z.B. Ölpreiserhöhungen oder Umweltkatastrophen sein. Außerdem haben in diesen Modellen die Marktteilnehmer die Möglichkeit, bestimmte Variablen durch ihr Handeln zu beeinflussen, d.h. zu steuern. Deshalb spielt die Theorie optimaler Steuerungen im Hintergrund eine wichtige Rolle.

In der Makroökonomie ist es üblich, volkswirtschaftliche Modelle durch Linearisierungen zu approximieren, da sie zu komplex sind um analytische Lösungen zuzulassen. Die Idee, ausgehend von einer Gleichgewichts-Lösung in einer Umgebung dieses Punktes bestimmte Entscheidungsregeln zu approximieren, findet sich schon 1982 bei KYDLAND und PRESCOTT [16]. Weil die Autoren die Methoden zur systematischen Untersuchung volkswirtschaftlicher Zusammenhänge grundlegend verbesserten, wird dieser Artikel wird in der Literatur als wegweisende Arbeit angesehen.
Allerdings wird hier laut JIN und JUDD in [13] die Linearisierung eines deterministi-

schen Modells auf ein stochastisches angewandt und Terme höherer Ordnung werden vernachlässigt, wodurch Ungenauigkeiten oder sogar falsche Ergebnisse entstehen.

Diese Arbeit untersucht deshalb ein Verfahren zur quadratischen Approximation der Kontrolle und der Dynamik eines Modells von SCHMITT-GROHÉ und URIBE, das in [23] beschrieben wird. Die Autoren stellen weiterhin Routinen in MATLAB zur Verfügung, die die Methode implementieren.
Ziel dieser Arbeit ist es, mathematische Voraussetzungen und Grundlagen dieser Methode aufzuzeigen, und die einzelnen Schritte zu beschreiben und herzuleiten.
Weiterhin sollen die Programme an einem Beispiel angewandt werden, um zu zeigen, wie gut die Methode zur Approximation der optimalen Steuerungsvariablen u und der optimalen Wertefunktion V geeignet ist. Dazu wird als erstes das approximierte Ergebnis mit der exakten Lösung verglichen, dann erfolgt ein Vergleich mit einem Verfahren aus der dynamischen Programmierung.

Die Arbeit ist wie folgt aufgebaut:
Im nächsten Kapitel wird zuerst allgemein die Idee der Perturbations-Methoden eingeführt. Der Hauptpunkt des Kapitels ist die Herleitung der Gleichgewichtsbedingungen für allgemeine, dynamische, stochastische Gleichgewichtsmodelle. Dazu wird ein optimales Kontrollproblem formuliert und die notwendigen Optimalitätsbedingungen aufgestellt. Danach wird ein Euler-Verfahren zur Diskretisierung von stetigen Kontrollproblemen definiert und anschließend wird auf die Besonderheiten bei diskreten stochastischen Prozessen eingegangen. Der letzte Abschnitt beschreibt das untersuchte Modell und die getroffenen Annahmen.

Im dritten Kapitel erfolgt die Herleitung und Berechnung der Koeffizienten für die lineare Approximation, wobei auch auf die Eindeutigkeit dieser Terme eingegangen wird. Der nächste Teil beschäftigt sich mit der Berechnung der quadratische Approximation. Diese beiden Abschnitte bilden die theoretische Grundlage für die Implementierung. Zum

Schluß werden die Schritte der Methode zusammengefaßt und ein Ausblick auf weitere Anwendungsmöglichkeiten gegeben.

Mit dem vierten Kapitel beginnt der praktische Teil. Hier werden die Routinen von SCHMITT-GROHÉ und URIBE überblickartig vorgestellt. Dabei wird der Aufbau der Programme sowie die benötigten Ein- und Ausgaben beschrieben.

Zur Auswertung der praktischen Ergebnisse wird im fünften Kapitel ein Beispielmodell vorgestellt und für das Verfahren umgeformt. Es wird erklärt, wie das Beispiel in MATLAB implementiert wurde. Danach werden die Ergebnisse der Berechnung aufgeführt und durch verschiedenen Graphiken veranschaulicht und diskutiert. Im letzten Abschnitt wird das Beispiel abgeändert, um den Unsicherheitsfaktor in die Zielfunktion mit einzubeziehen. Auch hier findet eine Auflistung der Ergebnisse statt und die Veränderung wird anhand von Abbildungen gezeigt.

Im sechsten Kapitel findet der Vergleich mit einem von GRÜNE [12] entwickelten Algorithmus statt, der auf dem BELLMANNschen Optimalitätsprinzip basiert. Diese Theorie wird zuerst allgemein vorgestellt. Anschließend werden wichtige Grundlagen und Bausteine des Algorithmus vorgestellt. Der zweite Abschnitt vergleicht und erläutert die Ergebnisse von beiden Beispielen mit denen aus dem vorherigen Kapitel.

Zum Abschluß der Arbeit werden die praktischen Ergebnisse noch einmal zusammengefaßt und ein Ausblick auf Verbesserungsmöglichkeiten gegeben.

An dieser Stelle möchte ich mich bei Herrn Prof. Dr. Grüne für die interessante Aufgabenstellung, die motivierenden Anregungen sowie die umfangreiche Betreuung im Laufe der Arbeit bedanken.
Ebenso danke ich Jürgen Pannek für seine hilfsbereite Unterstützung und für viele interessante Diskussionen.

2 Theorie des Verfahrens

In diesem Kapitel wird der theoretische Hintergrund der Methode von SCHMITT-GROHÉ und URIBE erläutert. Ziel ist es, die mathematischen Voraussetzungen herzuleiten und das Verfahren in die Theorie einzuordnen.
Es wird zuerst die grundlegende Vorgehensweise bei Perturbations-Methoden beschrieben.
Im zweiten Abschnitt werden die theoretischen Grundlagen des Verfahrens definiert. Zuerst wird die Gleichgewichtsbedingung für die Klasse der allgemeinen, dynamischen, stochastischen Gleichgewichtsmodelle eingeführt, da diese die Basis für die Methode bildet. Um die Gleichgewichtsbedingung aufzustellen und um ihren Hintergrund zu verstehen, wird die Theorie der optimalen Kontrollen benötigt. Es wird deshalb ein typisches Kontrollproblem formuliert. Desweiteren werden die notwendigen Optimalitätsbedingungen benannt. Danach wird die zeitliche Diskretisierung von stetigen Kontrollsystem angesprochen. Am Schluß des zweiten Abschnitts wird auf diskrete stochastische Prozesse eingegangen und die dazu benötigten Definitionen erklärt.
Der Modellrahmen für das Verfahren wird im dritten Abschnitt vorgestellt und es wird erklärt, welche Tatsachen das Verfahren ausnutzt, um die Approximation zu erhalten.

2.1 Perturbations Methoden - Einführung

Wie in MURDOCK [19] beschrieben, wurden Perturbations-Methoden ursprünglich für das Studium der Himmelskörperbewegungen entwickelt, um die Gravitationskräfte zwischen den Planeten besser erklären zu können.
In der angewandten Mathematik ist es die Grundidee dieser Methode, ausgehend von einer bekannten Lösung eines Problems, diese zu stören und so auf Werte in ihrer Umgebung schließen zu können. Grundsätzlich bedeutet dies für ein allgemein formuliertes

Problem, daß ein bestimmter Fall mit einer bekannten Lösung gefunden werden muß. Diese wird benutzt, um durch die Störung approximative Werte in einer Umgebung des Punktes zu bekommen. Eine gut verständliche Einführung zu diesem Thema liefert JUDD in [14] und [15].

Bei dem von SCHMITT-GROHÉ und URIBE in [23] eingeführten Verfahren wird zwischen einem deterministischen Modell, welches der ungestörten Lösung entspricht, und einem stochastischen Modell, welches eine Störung von außen enthält, unterschieden.
Hier wird die schon bekannte Lösung durch einen Gleichgewichtspunkt geliefert, der sich in einer Modell-Volkswirtschaft bildet. Ein volkswirtschaftliches Gleichgewicht ist wie in OBERENDER [21] definiert als

> [...] eine Situation, in der kein Wirtschaftssubjekt Veranlassung hat, sein Verhalten zu ändern.

Dieses Gleichgewicht muß vorher berechnet werden und bildet den Ausgangspunkt. Bei der Methode werden also lokale Informationen benutzt, um eine quadratische Approximation der Lösung des deterministischen Modells in der Nähe eines Gleichgewichts zu berechnen. Die Informationen werden dann benutzt, um die Lösung des stochastischen Problems zu approximieren.
Das Resultat einer Perturbations-Methode ist ein Polynom, welches die analytische Lösung in einer Umgebung des speziellen Punktes annähert.
Hierbei können Schwierigkeiten auftreten, die unabhängig vom Verfahren sind und ihre Ursache im zugrundeliegenden Modell haben:

(i) Das Gleichgewicht ist nicht eindeutig, es gibt mehrere Gleichgewichtspunkte. In diesem Fall ist bei einem volkswirtschaftlichen Modell zu überlegen, welches Gleichgewicht unter ökonomischen Gesichtspunkten das Sinnvollere ist.

(ii) Das Gleichgewicht ist kein Punkt, sondern eine periodische Lösung.

(iii) Es existiert kein Gleichgewicht.

(iv) Das Gleichgewicht existiert, ist aber instabil.

2.2 Theoretische Grundlagen

Bemerkung 2.1 *Im Unterschied zu dem Artikel von* SCHMITT-GROHÉ *und* URIBE *werden hier im folgenden die Kontrollvariablen mit u_t anstatt mit y_t benannt, um die Bezeichnung der einschlägigen Literatur über optimale Kontrollen anzupassen.*

Wie in der Einführung beschrieben, wird als Ausgangspunkt eine bekannte Lösung benötigt, wofür hier die Lösung eines volkswirtschaftlichen Gleichgewichtsmodells benutzt wird.
Die **Gleichgewichtsbedingungen** für allgemeine dynamische, stochastische Gleichgewichtsmodelle lauten:

$$E_t f(u_{t+1}, u_t, x_{t+1}, x_t) = 0 \qquad \forall\, t \in \mathbb{N}_0. \tag{2.1}$$

Der Operator E_t stellt den Erwartungswert, abhängig von allen zum Zeitpunkt t vorhandenen Informationen, dar. u_t bezeichnet den Kontrollvektor zum Zeitpunkt t, x_t ist der Zustandsvektor zum Zeitpunkt t.
Um die Funktion f aus der Gleichgewichtsbedingung definieren zu können, müssen vorher einige Überlegungen stattfinden, auf denen f basiert. Diese werden in den folgenden zwei Abschnitten angestellt. Eine Definition von f wird am Ende von Abschnitt 2.2.2 gegeben.

Als wesentlicher Punkt ist bei den obigen Gleichgewichtsbedingungen zu beachten, daß sie bereits die Bedingungen für einen optimalen Punkt enthalten. Deshalb ist die Frage zu klären, warum diese darin schon berücksichtigt sind.
In einer Volkswirtschaft wird davon ausgegangen, daß jedes Individuum das Ziel hat, seinen Nutzen zu maximieren. In VAN DER PLOEG [22] wird diese Tatsache wie folgt beschrieben:

> In einem allgemeinen Gleichgewichtsmodell werden die Zustände einer Volkswirtschaft untersucht, die durch das Optimierungsverhalten der einzelnen Individuen entstehen.

Dies bedeutet, daß ein Optimierungsproblem existiert, welches die Dynamik des Marktes bestimmt. Aus den Annahmen über Optimalität bildet sich -über die Dynamik- als Folge ein volkswirtschaftliches Gleichgewicht. Aus diesen Überlegungen kann gefolgert werden, daß ein Gleichgewicht -Existenz und Annahme vorausgesetzt- einen optimalen Punkt darstellt, da jeder Marktteilnehmer vorher seine optimale Strategie verfolgt hat.

Daraus folgt, daß zum Verständnis der Gleichgewichtsgleichung (2.1) zuerst die Bedingungen für einen optimalen Punkt hergeleitet werden müssen. Dafür werden im nächsten Abschnitt wichtige Definitionen eingeführt und im Anschluß daran die notwendigen Optimalitätsbedingungen formuliert. Dies führt weiter in das Gebiet der optimalen Kontrolltheorie, da die Gleichgewichtsgleichung von der Kontrollvariablen u abhängt.

2.2.1 Ein deterministisches optimales Kontrollproblem

Bemerkung 2.2 *Die Begriffe „Steuerung“ und „Kontrolle“ werden im folgenden synomym verwendet.*

Zu Beginn muß festgelegt werden, welche Werte eine Kontrolle für den untersuchten Prozeß annehmen darf.

Definition 2.3 *Die nichtleere und kompakte Menge $\mathbf{U} \subset \mathbb{R}^m$ wird als Kontrollbereich festgelegt und beschreibt die zulässige Wertemenge für den Kontrollvektor u_t, d.h.*

$$u_t \in \mathbf{U} \quad \forall\, t \in \mathbf{T}$$

mit $\mathbf{T} := h\mathbb{N}_0 := \{hk | k \in \mathbb{N}_0\}$ für $h > 0$.

Bemerkung 2.4 *Im folgenden wird als Schrittweite $h = 1$ gewählt.*

Des weiteren wird der Begriff des Kontrollsytems benötigt:

Definition 2.5 *Ein Kontrollsystem in diskreter Zeit $\mathbf{T}$ im $\mathbb{R}^d$, $d \in \mathbb{N}$, ist gegeben durch die Differenzengleichung*

$$x_{t+1} = \varphi_h(x_t, u_t) \tag{2.2}$$

wobei $\varphi_h : \mathbb{R}^d \times \mathbf{U} \to \mathbb{R}^d$ eine stetige Abbildung ist.

Ein Kontrollsystem legt die Dynamik, also die Bewegung des Systems in der Zeit, fest. Für Kontrollsysteme mit genau festgelegten Eigenschaften ist es möglich, für jeden beliebigen Anfangswert eindeutige Lösungen zu finden.

Definition 2.6 *Eine eindeutige Funktion x_t, die Lösung von (2.2) ist zum Anfangswert $x_0 \in \mathbb{R}^d$ und zur Kontrollfunktion $u_t \in \mathbf{U}$, wird mit $\Phi_h(t, x_0, u_t)$ bezeichnet .*

Die Existenz einer eindeutigen Lösung kann im diskreten Fall durch Induktion gezeigt werden, d.h. zu jedem Anfangswert x_0 und jeder Kontrolle $u_t \in \mathbf{U}$ findet man eine eindeutige Lösung $\Phi_h(t, x_0, u_t)$, $\Phi_h : \mathbf{T} \times \mathbb{R}^d \times \mathbf{U}$, mit

$$\Phi_h(0, x_0, u_t) = x_0 \quad \text{und} \quad \Phi_h(t+1, x_0, u_t) = \varphi_h(\Phi_h(t, x_0, u_t), u_t)$$

Wie im vorherigen Abschnitt beschrieben, spielen Gleichgewichte in dem Verfahren von SCHMITT-GROHÉ und URIBE eine wesentliche Rolle. Deshalb muß formuliert werden, was ein Gleichgewicht ist:

Definition 2.7 *Ein Zustand $\bar{x}$ aus einer Menge $\mathbf{X} \subset \mathbb{R}^d$ und eine Kontrolle $\bar{u} \in \mathbf{U}$ heißen Gleichgewicht eines Kontrollsystems φ_h, falls $\varphi_h(\bar{x}_t, \bar{u}_t) = \bar{x}_t$ für $t \to \infty$.*

Wenn ausschließlich nur ein Kontrollsystem betrachtet wird, kann es unendlich viele Lösungen geben. Da es das Ziel ist, eine optimale Lösung zu finden, muß ein Gütemaß dafür bestimmt werden. Dies geschieht durch eine Zielfunktion.
Dazu wird eine Funktion

$$\psi : \mathbb{R}^d \times \mathbf{U} \to \mathbb{R}$$

betrachtet.
Aufgabe ist es nun, eine Kontrolle $u_t \in \mathbf{U}$ zu finden, die die Summe über diese Funktion entlang einer Trajektorie Φ_h maximiert. Die folgende Definition formuliert deshalb ein optimales Kontrollproblem.

Definition 2.8 *Betrachte ein Kontrollsystem (2.2).*
Für eine Funktion $\psi : \mathbb{R}^d \times \mathbf{U} \to \mathbb{R}$ und einen Parameter $\delta > 0, \delta \in \mathbb{R}$, definieren wir das diskontierte Funktional auf unendlichem Zeithorizont in diskreter Zeit als

$$J_h(x_t, u_t) := h \sum_{t=0}^{\infty} \beta^t \cdot \psi(\Phi_h(t, x_0, u_t), u_t) \tag{2.3}$$

mit $\beta := (1 - \delta h)$. Das optimale Kontrollproblem lautet dann: Bestimme die optimale Wertefunktion

$$V_h(x) := \sup_{u_t \in \mathbf{U}} J_h(x_t, u_t).$$

Dabei gelte

(i) $\mathbf{U}$ *ist kompakt.*

(ii) Die Funktion ψ sei stetig und erfülle

$$|\psi(x, u)| \leq M_\psi \quad \textit{und} \quad |\psi(x_1, u) - \psi(x_2, u)| \leq L_\psi \|x_1 - x_2\|$$

für alle $x, x_1, x_2 \in \mathbb{R}^d$, alle $u \in \mathbf{U}$ und geeignete Konstanten $M_\psi, L_\psi > 0$, mit $M_\psi, L_\psi \in \mathbb{R}$.

Bemerkung 2.9 *Die Formulierung der optimalen Wertefunktion ist natürlich auch für ein Minimierungsproblem möglich. Sie lautet dann*

$$V_h(x) := \inf_{u_t \in \mathbf{U}} J_h(x, u).$$

2.2.2 Notwendige Optimalitätsbedingungen

Mit den im vorhergehenden Abschnitt definierten Grundlagen können nun die notwendigen Bedingungen für ein Optimum hergeleitet werden. Dazu wird die Definition der HAMILTON-Funktion benötigt:

Definition 2.10 *Sei $\lambda_0 \in \mathbb{R}$ und $\lambda \in \mathbb{R}^n$. Dann heißt*

$$H^t(x_t, u_t, \lambda_t) := \lambda_0 \cdot \beta^t \psi(x_t, u_t) + \lambda_{t+1} \cdot \varphi_h(t, x_t, u_t) \tag{2.4}$$

die HAMILTON*-Funktion zum Kontrollproblem aus Definiton 2.8.*
Der Vektor $\lambda \in \mathbb{R}^n$ wird adjungierte Variable oder Kozustand genannt.

Bemerkung 2.11 *λ_0 kann unter zusätzlichen Bedingungen zu $\lambda_0 = 1$ gewählt und tritt daher später nicht in der* HAMILTON*-Funktion auf.*

Die Optimalitätsbedingungen für das optimale Kontrollproblem aus Definition 2.8 ergeben sich aus dem folgenden Satz, der eine Abwandlung des PONTRJAGINschen Maximumprinzips ist.

Satz 2.12 ***(Diskretes Maximumprinzip)*** *Es sei u_t^* eine optimale Entscheidungsfolge und x_t^* die zugehörige Zustandsfolge. Dann existiert eine Kozustandsfolge λ_t, so daß gilt*

$$\boxed{\begin{array}{rcll} x_{t+1}^* & = & \left(\frac{\partial H^t}{\partial \lambda_{t+1}}\right) & \textit{Zustandsgleichung} \\ \lambda_t & = & \left(\frac{\partial H^t}{\partial x_t^*}\right) & \textit{Kozustandsbedingung} \\ 0 & = & \left(\frac{\partial H^t}{\partial u_t^*}\right) & \textit{Maximumbedingung} \end{array}} \tag{2.5}$$

Bemerkung 2.13 *Zum diskreten Maximumprinzip gehören außer den obigen Bedingungen noch die Transversalitätsbedingungen. Dies sind Bedingungen, die den Anfangs- und Endzustand festlegen. Da in dem Verfahren von* SCHMITT-GROHÉ *und* URIBE *der Anfangs- und Endzustand bereits fix ist, werden sie im weiteren nicht benötigt.*

Bemerkung 2.14 *Satz 2.12 gilt unter der Bedingung, daß keine Beschränkung der Steuervariablen und ein unendlicher Zeithorizont vorliegen.*
Für das im folgenden betrachtete Verfahren genügt diese Formulierung; Versionen für stetige Optimalsteuerungsprobleme, für Probleme mit Nebenbedingungen für die Kontrollvariablen und für beschränkten Zeithorizont sind nachzulesen in FEICHTINGER *und* HARTL *[7]. Je nach Problemstellung sind die Bedingungen also anzupassen. In [7] finden ebenfalls weitere Überlegungen zum Aufstellen hinreichender Optimalitätsbedingungen statt.*

Bemerkung 2.15 *Für den Fall, daß H konkav in (x, u) ist, sind die obigen Bedingungen auch hinreichend.*

Bemerkung 2.16 *In dem untersuchten Verfahren wird die Annahme gemacht, daß in einem volkswirtschaftlichen Gleichgewicht die Störung von außen gleich Null ist. Der stochastische Anteil fällt dadurch also weg und es wird ein deterministisches Problem betrachtet. Deshalb reichen hier die Optimalitätsbedingungen für deterministische Kontrollprobleme. Im Gleichgewicht gilt dann also* $E_t f = f = 0$.

In den letzten beiden Abschnitten wurden die Hintergründe zum Verständnis der Gleichgewichtsgleichung (2.1) hergeleitet.
Das Aufstellen der Optimalitätsbedingungen ist ein Vorschritt zum eigentlichen Verfahren. Sie werden benutzt um f zu bilden.

Definition 2.17 *Für die Funktion* f *aus Gleichung (2.1) gilt:*

$$f := \begin{pmatrix} x^*_{t+1} - \frac{\partial H^t}{\partial \lambda_{t+1}} \\ \lambda_t - \frac{\partial H^t}{\partial x^*_t} \\ \frac{\partial H^t}{\partial u^*_t} \end{pmatrix} = 0 \tag{2.6}$$

Im folgenden wird immer davon ausgegangen, daß die Gleichgewichtsbedingungen in der Form aus Gleichung (2.1) ausgedrückt werden, wobei der Erwartungswert im Gleichgewicht wegfällt, da die stochastische Störung hier nicht vorhanden ist.

Zusammenfassend läßt sich sagen, daß vor Beginn des eigentlichen Verfahrens die Bedingungen für ein volkswirtschaftliches Gleichgewicht mit Hilfe der HAMILTON-Funktion aufgestellt werden müssen und daß diese im weiteren Verlauf sowohl als Ausgangspunkt als auch für wesentliche Schritte bei den Berechnungen benutzt werden. Die Lösung für das Gleichgewicht - also der volkswirtschaftliche Gleichgewichtspunkt - muß zusätzlich bekannt sein, da man eine bekannte Lösung für das Verfahren benötigt.

2.2.3 Diskretisierung des optimalen Kontrollproblems

Bisher wurden Kontrollprobleme in diskreter Zeit betrachtet. Wenn ein Kontrollsystem in kontinuierlicher Zeit vorliegt, muß überlegt werden, wie dieses durch ein diskretes System approximiert werden kann.

Definition 2.18 *Die nichtleere und kompakte Menge $\mathbf{U} \subset \mathbb{R}^m$ wird als Kontrollbereich festgelegt und definiert die Werte, die die Kontrolle $u(t)$ für $t \in \mathbb{R}$ annehmen darf.*

Definition 2.19 *Ein Kontrollsystem in kontinuierlicher Zeit $\mathbf{T} = \mathbb{R}$ im $\mathbb{R}^d$,$d \in \mathbb{N}$, ist gegeben durch die gewöhnliche Differentialgleichung*

$$\frac{d}{dt}x(t) = \varphi(x(t), u(t)), \tag{2.7}$$

wobei $\varphi : \mathbb{R}^d \times \mathbf{U} \to \mathbb{R}^d$ ein stetiges Vektorfeld ist.

Auch für stetige Kontrollsysteme ist es unter bestimmten Voraussetzungen möglich, für jeden beliebigen Anfangswert x_0 eindeutige Lösungen zu finden. Dazu sei auf den Existenz- und Eindeutigkeitssatz von CARATHÉODORY in GRÜNE [10] verwiesen.

Definition 2.20 *Eine eindeutige Funktion $x(t)$, die den Satz von* CARATHÉODORY *erfüllt, wird mit $\Phi(t, x_0, u)$ bezeichnet und ist die Lösung zu (2.7) zum Anfangswert $x_0 \in \mathbb{R}^d$ und zur Kontrolle $u(t) \in \mathbf{U}$.*

Für ein Kontrollproblem in stetiger Zeit ergibt sich somit die folgende Definition:

Definition 2.21 *Betrachte ein Kontrollsystem (2.7).*
Für eine Funktion $\psi : \mathbb{R}^d \times \mathbf{U} \to \mathbb{R}$ und einen Parameter $\delta > 0, \delta \in \mathbb{R}$ definieren wir das diskontierte Funktional auf unendlichem Zeithorizont in kontinuierlicher Zeit als

$$J(x, u) := \int_0^\infty e^{-\delta t}\psi(\Phi(t, x_0, u), u(t))dt \tag{2.8}$$

Das optimale Kontrollproblem lautet dann: Bestimme die optimale Wertefunktion

$$V(x) := \sup_{u \in \mathcal{U}} J(x, u)$$

Außerdem gelten die zusätzlichen Bedingungen aus Definition 2.8.

Für die zeitliche Diskretisierung ist als Grundlage die folgende Definition notwendig:

Definition 2.22 *(**EULER-Verfahren für Kontrollsysteme**) Gegeben sei ein kontinuierliches Kontrollsystem φ gemäß Gleichung (2.7). Für einen Zeitschritt $h > 0$, $h \in \mathbb{R}$, und einen Kontrollwert $u \subset \mathbb{R}^m$ definieren wir das* EULER*-Verfahren als das durch die Abbildung*

$$\widetilde{\varphi}_h(x, u) := x + h \cdot \varphi(x, u)$$

definierte zeitdiskrete Kontrollsystem. Die Lösungen werden mit $\widetilde{\Phi}_h(t, x_0, u_t)$ bezeichnet.

In GRÜNE [10] wird gezeigt, daß zu einem Kontrollsystem eine diskrete EULER-Approximation konstruiert werden kann, die zu jedem Anfangswert $x_0 \in \mathbb{R}^d$ und zu jeder diskreten Kontrollfunktion $u_t : h\mathbb{N}_0 \to \mathbf{U}$ eine zeitdiskrete approximative Lösung $\widetilde{\Phi}_h(t, x_0, u_t)$ liefert:

Satz 2.23 *Betrachte ein Kontrollsystem φ, das die Voraussetzungen des Satzes von Carathéodory erfüllt. Dann gilt für das* EULER*-Verfahren aus Definition 2.22 und jede Konstante $R > 0$ die folgende Aussage:*

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

$$\|\widetilde{\Phi}_h(t, x_0, u_t) - \Phi(t, x_0, u)\| \leq K\sqrt{h}\, e^{Lt}$$

gilt für alle $t \in h\mathbf{T}$ für die die Lösungen in $\bar{B}_R(0)$ liegen.
Ist das Kontrollsystem konvex, so gilt die schärfere Abschätzung

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

(ii) Umgekehrt existiert eine von R unabhängige Konstante $K > 0$, so daß für jedes $x_0 \in \bar{B}_R(0)$ und jede diskrete Kontrollfunktion $u_t \in \mathbf{U}$ mit $\|u_t\|_\infty \leq R$ und die durch

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

definierte stückweise konstante Kontrollfunktion die Abschätzung

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

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

Für ein gegebenes kontinuierliches optimales Kontrollproblem wird deshalb das zu dieser EULER-Approximation gehörige diskrete optimale Kontrollproblem betrachtet:

$$\widetilde{V}_h(x) := \sup_{u_t \in \mathbf{U}} \widetilde{J}_h(x, u_t) \quad \text{mit} \quad \widetilde{J}_h(x, u_t) := h \sum_{t=0}^{\infty} \beta^t \psi(\widetilde{\Phi}_h(t, x, u_t), u_t).$$

Bemerkung 2.24 *Das* EULER*-Verfahren für Kontrollsysteme hat die Konvergenzordnung $O(\sqrt{h})$ bzw. $O(h)$ für konvexe Kontrollprobleme.*

Für nähere Untersuchungen über die Konvergenz von $\widetilde{V}_h \to V$ und über die Abschätzung des Diskretisierungsfehlers sei verwiesen auf GRÜNE [10], Kapitel 3.

2.2.4 Diskrete stochastische Prozesse

In dem Modell, das im nächsten Abschnitt eingeführt wird, folgt eine Variable einer stochastischen Dynamik. Da dies bisher noch nicht berücksichtigt wurde, sollen an dieser Stelle die wichtigsten Grundbegriffe geklärt werden.

Optimierungsprobleme unter Unsicherheit weisen bestimmte Eigenschaften auf, die man von deterministischen Optimierungsproblemen nicht kennt. Dies macht die Untersuchung stochastischer Problemen komplizierter. Allerdings ist in vielen Anwendungen die Annahme rein deterministischer Modelle unrealistisch und kann sogar zu falschen Aussagen führen.
Unsicherheit tritt durch zufällige äußere Einflüsse auf. Gründe dafür können z.B. ungenaue Informationen über das System oder nur teilweise Beobachtbarkeit der Variablen sein. Unsicherheit spielt in vielen Modellen eine große Rolle; in der Ökonomie z.B., wenn der Ertrag zweier unterschiedlich riskanter Anlagemöglichkeiten verglichen werden soll.

Systeme, in denen Zufallsvariablen eine Rolle spielen, benötigen dafür einen geeigneten Wahrscheinlichkeitsraum.

Definition 2.25 *Ein Tripel $(\Omega, \mathcal{A}, P)$ heißt Wahrscheinlichkeitsraum, wobei Ω der Ereignisraum ist, $\mathcal{A}$ eine entsprechende $\sigma-$Algebra und P ein geeignetes Wahrscheinlichkeitsmaß.*

Bemerkung 2.26 *Der Vollständigkeit wegen müssen die entsprechenden maßtheoretischen Grundlagen definiert werden. Hierfür sei auf die einschlägige Literatur verwiesen wie u.a.* CAPIŃSKI *und* KOPP *[5] und* ELSTRODT *[6].*

Definition 2.27 *Sei X eine reelle Zufallsvariable auf $(\Omega, \mathcal{A}, P)$. Die Verteilung von $\mathbf{X}$ ist diskret, also getragen von einer abzählbaren Wertemenge $\{x_i | i \in I\}$. Falls die Reihe*

$$\sum_{i \in I} x_i \cdot P(\mathbf{X} = x_i)$$

absolut konvergent ist, so sagt man, der Erwartungswert $E[X]$ existiert und man setzt

$$E[X] := \sum_{i \in I} x_i \cdot P(X = x_i).$$

Definition 2.28 *Es sei X eine reelle Zufallsvariable auf $(\Omega, \mathcal{A}, P)$ mit existierendem zweiten Moment, d.h. $E[X^2] < \infty$. Dann existieren auch die Erwartungswerte $E[X]$ und $E[(X - E[X])^2] =: Var[X]$ und letzterer heißt Varianz von X. Der Parameter $+\sqrt{Var[X]}$ wird als Standardabweichung oder Streuung bezeichnet.*

Zum Beschreiben von dynamischen Systemen unter Unsicherheit benötigt man zuerst die Definition eines stochastischen Prozesses. Es werden ausschließlich diskrete stochastische Prozesse betrachtet, da diese für numerische Verfahren die Hauptrolle spielen.

Definition 2.29 *Für Systeme in diskreter Zeit bedeutet ein stochastischer Prozeß, daß x_{t+1} eine Zufallsvariable ist, die nur von x_t und t abhängt. Der Prozeß kann durch eine stochastische Differenzengleichung ausgedrückt werden:*

$$x_{t+1} = \zeta(x_t, t) + z(x_t, t), \ t \in \mathbf{T}. \tag{2.9}$$

Hierbei ist ζ der bedingte Mittelwert von x_{t+1}, gegeben x_t, und $z(.)$ ist eine d-dimensionale Zufallsvariable mit Mittelwert Null, deren bedingte Verteilung, gegeben x_t, nicht von x_s abhängt für $s < t$.

Einen diskreten stochastischen Prozeß erhält man also, wenn ein nur vom aktuellen Zustand abhängiger Störungsterm zu einer gewöhnlichen Differenzengleichung addiert wird.

Bemerkung 2.30 *Bei der obigen Definition wird von einer exogenen Störung ausgegangen, also nicht von einer Störung durch Meßfehler, bei der die Zustandsvariablen nicht genau beobachtet werden können.*

Die Dynamik eines stochastischen Systems ist also durch eine stochastische Differenzengleichung gegeben.
Unter der Annahme, daß der Störungsterm $z(x_t) \sim N(\mu, \sigma)$ ist, kann die stochastische Variable durch ihre Varianz σ^2 normalisiert werden, so daß $\frac{z(x_t)}{\sigma(x_t)} \sim N(0,1)$-verteilt ist. Daraus folgt

$$z = \sigma(x_t) \cdot \varepsilon_t,$$

wobei ε unabhängig von x ist. $\varepsilon_t, t \in \mathbf{T}$ kann als Folge von unabhängigen und identisch $N(0,1)$-verteilten Zufallsvariablen betrachtet werden. Für die stochastische Differenzengleichung (2.9) gilt dann

$$x_{t+1} = \zeta(x_t) + \sigma(x_t) \cdot \varepsilon_t. \tag{2.10}$$

Definition 2.31 *Die Definition der optimalen Wertefunktion aus Abschnitt 2.2.1 kann leicht auf diskrete stochastische Kontrollprobleme übertragen werden. Sie lautet dann*

$$V(x) = E\left(\sup_{u_t \in \mathbf{U}} \sum_{t=0}^{\infty} \beta^t \cdot \psi(x_t, u_t)\right),$$

mit $\quad x_{t+1} = \varphi(x_t, u_t, z_t)$
und die z_t sind unabhängig und identisch verteilte Zufallsvariablen.

Für ein deterministisches System ist die zukünftige Bewegung durch die Dynamik und den aktuellen Wert des Systems eindeutig festgelegt. Für ein stochastisches System läßt sich dies natürlich nicht fordern. Wären hier alle Kontrollen vorher festgelegt, würde man im Verlauf des Systems feststellen, daß unerwartete Zustände eintreten und daß die ausgewählten Kontrollen bezüglich der Zielfunktion nicht mehr optimal sind.
Aus diesen Gründen wird auch noch einmal die Motivation deutlich, weshalb es so interessant ist, für stochastische Systeme, ausgehend von einem deterministischen System, die Kontrollen in einer bestimmten Umgebung zu approximieren.

2.3 Das Modell

In diesem Abschnitt werden alle Annahmen des Verfahrens dargestellt, um der Methode einen äußeren Rahmen zu geben. Im Anschluß wird das genaue Ziel des Verfahrens formuliert und wie es mathematisch erreicht werden kann.
In dem Modell wird der Zustandsvektor x_t wird nochmal unterteilt, so daß gilt

$$x_t = [x_t^1; x_t^2]^T$$

mit $x_t^1 \in \mathbb{R}^{d_1}, x_t^2 \in \mathbb{R}^{d_2}, d = d_1 + d_2$.
Auf diese Weise wird berücksichtigt, daß ein Teil der Zustände einer rein deterministischen Dynamik folgt, und ein anderer einer stochastischen Dynamik.
Das betrachtete Modell unterliegt verschiedenen wichtigen Voraussetzungen und Annahmen, die im folgenden aufgelistet werden.

Annahmen:

- x_t^1 enthält endogene, d.h. sich aus dem Modell ergebende, Zustandsvariablen.
- x_t^2 folgt einem exogenen stochastischen Prozeß, d.h. mit einer Störung von außen, gegeben durch

  $$x_{t+1}^2 = \widetilde{h}(x_t^2, \sigma) + \widetilde{\eta}\sigma\varepsilon_{t+1}$$

  mit $x_t^2, \varepsilon \in \mathbb{R}^{d_2}, \widetilde{\eta} \in \mathbb{R}^{d_2 \times d_2}, \sigma \geq 0$ und $\widetilde{h} : \mathbb{R}^{d_2} \times \mathbb{R} \to \mathbb{R}^{d_2}$

- Der Vektor ε_t ist unabhängig identisch verteilt mit Erwartungswert Null und Varianz/Kovarianzmatrix I.

- Die Standardabweichung $\sigma \geq 0$ und die Matrix $\widetilde{\eta}$ sind bekannt.

- Es wird angenommen, daß Funktionen $g := \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^m$ und $h := \mathbb{R}^d \times \mathbb{R} \rightarrow \mathbb{R}^d$ existieren, so daß für den Kontrollvektor

$$u_t = g(x_t, \sigma) \tag{2.11}$$

  und für den Zustandsvektor

$$x_{t+1} = h(x_t, \sigma) + \eta \cdot \sigma \cdot \varepsilon_{t+1} \tag{2.12}$$

  gilt.

- $\eta \in \mathbb{R}^{d \times d_2}$ und $\eta := \begin{bmatrix} O \\ \widetilde{\eta} \end{bmatrix}$.

- Die Eigenwerte der Matrix h_x sind $\lambda_i < 1$, $i = 1, ..., d$.
  Dabei bezeichnet h_x die erste Ableitung der Funktion h in x-Richtung.

- Für die Funktion f aus (2.1) gilt dann:

$$f : \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}^d \times \mathbb{R}^d \longrightarrow \mathbb{R}^n \quad \text{mit} \quad n := d + m.$$

- Der Gleichgewichtspunkt $(\bar{x}, \bar{u})$ sei bekannt.

- Im Gleichgewicht soll gelten $\sigma = 0$, d.h. die Störung von außen ist im Gleichgewicht ausgeschaltet und im Gleichgewichtspunkt $(\bar{x}, \bar{u})$ gilt

$$f(\bar{u}, \bar{u}, \bar{x}, \bar{x}) = 0.$$

  Daraus folgt $\bar{u} = g(\bar{x}, 0)$ und $\bar{x} = h(\bar{x}, 0)$.
  Der Erwartungswert spielt also im Gleichgewicht keine Rolle.

- f ist mindestens zweimal bezüglich aller Variablen differenzierbar.

- Die Ableitungen von f sind bekannt.

Bemerkung 2.32 *In* SCHMITT-GROHÉ *und* URIBE *[23] wird die zusätzliche Annahme gemacht, daß ε_t einen beschränkten Trager hat. Sie verweisen dabei auf* JIN *und* JUDD *[13], die an einem Beispiel zeigen, daß unlogische Ergebnisse herauskommen können, wenn keine Restriktionen an die zufälligen Störungen gemacht werden. Hier wird diese Annahme vernachlässigt, da in der Numerik in der Regel nie unbeschränkte Träger zu Grunde gelegt werden, sondern z. B. Intervalle mit einem Konfidenzniveau von 99%.*

Ziel der Methode ist es, die Dynamik h und die Kontrollfunktion g in einer kleinen Umgebung um das Gleichgewicht zu approximieren. Die Größe dieser Umgebung wird durch den Unsicherheitsfaktor σ bestimmt.
Diese Approximation geschieht durch eine mehrdimensionale TAYLOR-Entwicklung, für die der nächste Satz benötigt wird.

Satz 2.33 *Sei $\xi : \mathbb{R}^n \to \mathbb{R}$ und ξ sei C^{k+1}. Dann gilt für $x^0 \in \mathbb{R}^n$ und $x, x^0 \in [a, b]$*

$$\begin{aligned}
\xi(x) \;=\; & \xi(x^0) + \sum_{i=1}^{n} \frac{\partial \xi}{\partial x_i}(x^0)(x_i - x_i^0) \\
& + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial^2 \xi}{\partial x_i \partial x_j}(x^0)(x_i - x_i^0)(x_j - x_j^0) \\
& \vdots \\
& + \frac{1}{k!} \sum_{i_1=1}^{n} \cdots \sum_{i_k=1}^{n} \frac{\partial^k \xi}{\partial x_{i_1} \ldots \partial x_{i_k}}(x^0)(x_{i_1} - x_{i_1}^0) \ldots (x_{i_k} - x_{i_k}^0) \\
& + O(\|x - x^0\|^{k+1}).
\end{aligned}$$

Es muß noch eine weitere Überlegung stattfinden: Für die TAYLOR-Approximation der Funktionen g und h werden verschiedene Ableitungen benötigt. Die Frage ist, wie diese berechnet werden können. Zu diesem Zweck wird definiert:

$$F(x, \sigma) := E_t f(u_{t+1}, u_t, x_{t+1}, x_t) = 0. \tag{2.13}$$

Nun werden die Dynamiken (2.11) und (2.12) in Gleichung (2.13) eingesetzt.
Daraus ergibt sich

$$F(x, \sigma) = E_t f(g(h(x_t, \sigma) + \eta\sigma\varepsilon_{t+1}, \sigma), g(x_t, \sigma), h(x_t, \sigma) + \eta\sigma\varepsilon_{t+1}, x_t) = 0. \tag{2.14}$$

Die Funktion F wurde also als Nullfunktion definiert, d.h.

$$F(x, \sigma) \equiv 0 \quad \forall x \in \mathbb{R}^d, \sigma \geq 0.$$

Daraus läßt sich erkennen, daß auch die höheren Ableitungen von F immer Null sein müssen:

$$F_{x^k \sigma^j} = 0 \quad \forall x \in \mathbb{R}^d, \sigma \in \mathbb{R}, k, j \in \mathbb{N}. \tag{2.15}$$

$F_{x^k \sigma^j}$ ist die k-malige Ableitung in x-Richtung und die j-malige in σ-Richtung.
Diese Tatsache wird in dem untersuchten Verfahren benutzt, um die Ableitungen von g und h zu bestimmen, die für die TAYLOR Approximation gebraucht werden. Man nutzt also die Gleichgewichts-Bedingung, um Aussagen über die Steuerung und die Dynamik in einer Umgebung des Gleichgewichts zu formulieren.
Außerdem benötigt man noch den Gleichgewichtspunkt an sich, also $(\bar{x}, \bar{u})$, der vorher bestimmt wird, sowie die Ableitungen von f an dieser Stelle, die ja gemäß den Annahmen als bekannt vorausgesetzt werden.

Bemerkung 2.34 *Wichtig ist es, sich den genauen Unterschied zwischen den Funktionen F und f klar zu machen. F wird als Nullfunktion definiert, was bedeutet, daß F immer gleich Null sein muß für alle beliebigen Werte von x und σ. Im Gegensatz kann f komplett verschiedene Werte annehmen, allerdings gilt im Gleichgewichtspunkt $f(\bar{u}, \bar{u}, \bar{x}, \bar{x}) = 0$. Hier sind F und f gleich. Deshalb müssen die Ableitungen von f aber nicht unbedingt Null sein.*

3 Herleitung der Terme für die Approximationen

In diesem Kapitel werden die einzelnen Koeffizienten für die TAYLOR-Approximation hergeleitet. Der Inhalt orientiert sich dabei an den Ausarbeitungen von SCHMITT-GROHÉ und URIBE in [23].
Der erste Abschnitt beschäftigt sich mit der linearen Approximation. Hierbei wird zuerst die Herleitung der Koeffizienten gezeigt. Im Anschluß wird auf die Existenz und Eindeutigkeit der Terme eingegangen. Der zweite Abschnitt erklärt die quadratische Approximation. Im letzten Abschnitt wird das theoretische Resultat zusammengefaßt und ein kurzer Ausblick auf weitere Anwendungsmöglichkeiten gegeben.

3.1 Approximation erster Ordnung

3.1.1 Herleitung

O.B.d.A. sei $x \in \mathbb{R}$. Man sucht eine Approximation von g und h um den Entwicklungspunkt $(\bar{x}, 0)$ in der Form

$$g(x,\sigma) = g(\bar{x},0) + g_x(\bar{x},0)(x-\bar{x}) + g_\sigma(\bar{x},0)\sigma \tag{3.1}$$

$$h(x,\sigma) = h(\bar{x},0) + h_x(\bar{x},0)(x-\bar{x}) + h_\sigma(\bar{x},0)\sigma. \tag{3.2}$$

Der jeweils erste Term von (3.1) und (3.2) ist schon bekannt, denn

$$g(\bar{x},0) = \bar{u},$$
$$h(\bar{x},0) = \bar{x},$$

wie in Kapitel 2.3 angenommen wurde.

Um die fehlenden Terme zu berechnen, nutzt man nun Gleichung (2.15) aus, wie am Ende des letzten Kapitels angemerkt:

$$F_\sigma(\bar{x}, 0) = 0 \tag{3.3}$$

$$F_x(\bar{x}, 0) = 0 \tag{3.4}$$

Aus Gründen der Übersichtlichkeit werden bei der Berechnung der Ableitungen Variablen aus der Periode $(t+1)$ mit einem Strich indiziert, bei Variablen aus Periode t entfällt der Index ganz. Somit gilt für (3.3):

$$\begin{aligned} F_\sigma(\bar{x}, 0) &= E_t f_{u'}[g_x(h_\sigma + \eta\varepsilon') + g_\sigma] + f_u g_\sigma + f_{x'} h_\sigma \\ &= f_{u'}[g_x h_\sigma + g_\sigma] + f_u g_\sigma + f_{x'} h_\sigma = 0. \end{aligned}$$

Bemerkung 3.1 *An dieser Stelle wird die Annahme ausgenutzt, daß im Gleichgewicht keine Störung existiert. Nach Voraussetzung ist dann* $E[\varepsilon'] = 0$ *und deshalb fällt der stochastische Anteil weg.*

Durch Umformung erhält man

$$\underbrace{\begin{pmatrix} f_{u'} g_x + f_{x'} & f_{u'} + f_u \end{pmatrix}}_{=: \mathbf{M}} \cdot \begin{pmatrix} h_\sigma \\ g_\sigma \end{pmatrix} = 0. \tag{3.5}$$

Wenn eine eindeutige Lösung existiert, folgt daß

$$\begin{aligned} h_\sigma &= 0, \\ g_\sigma &= 0. \end{aligned}$$

Für die Ableitung in x-Richtung, also für (3.4), gilt:

$$F_x(\bar{x}, 0) = f_{u'} g_x h_x + f_{x'} h_x + f_x + f_u g_x.$$

Es ergibt sich

$$\begin{pmatrix} f_{x'} & f_{u'} \end{pmatrix} \cdot \begin{pmatrix} I \\ g_x \end{pmatrix} h_x = - \begin{pmatrix} f_x & f_u \end{pmatrix} \cdot \begin{pmatrix} I \\ g_x \end{pmatrix}. \tag{3.6}$$

Durch geeignete Zerlegungen kann man mit dieser Gleichung h_x und g_x berechnen, wie im nächsten Abschnitt gezeigt wird. Damit hat man alle Terme erhalten, um die lineare Approximation zu berechnen.

Bemerkung 3.2 *Normalerweise ist $x \in \mathbb{R}^d$ für $d > 1$, was die Schreibweise bei den Ableitungen verkompliziert. Deshalb wurde hier immer von einem eindimensionalen Zustandsvektor ausgegangen. Man berechnet im mehrdimensionalen Fall $[F_x(\bar{x}, 0)]^i_j$, also den Eintrag in Zeile i und Spalte j der Ableitungsmatrix von F nach x. Genauso berechnet man $[F_\sigma(\bar{x}, 0)]^i$, also den i-ten Eintrag des Ableitungsvektors. Zur Berechnung bei den mehrdimensionalen Ableitungen wird die mehrdimensionale Kettenregel benötigt. Für mögliche Schreibweisen vergleiche* SCHMITT-GROHÉ *und* URIBE *[23] oder für die Tensor-Notation* JUDD *[15].*

3.1.2 Existenz und Eindeutigkeit

Um zu verstehen, wie die Methode später implementiert wird, ist es an dieser Stelle wichtig zu zeigen, wie Gleichung (3.6) gelöst wird. Dazu werden einige Begriffe aus der linearen Algebra benötigt, die hier kurz definiert werden.

Definition 3.3 *Man nennt eine lineare Abbildung $F : V \to W$ einen Endomorphismus, wenn V=W ist.*

Definition 3.4 *Sei F ein Endomorphismus des K-Vektorraums V. Ein $\lambda \in K$ heißt Eigenwert von F, wenn es ein $v \in V$ mit $v \neq 0$ gibt, so daß gilt*

$$F(v) = \lambda v.$$

Jedes vom Nullvektor verschiedene $v \in V$ mit $F(v) = \lambda v$ heißt Eigenvektor von F zum Eigenwert λ.

Definition 3.5 *Sei V ein endlich-dimensionaler K-Vektorraum. Ein Endomorphismus $f : V \to V$ heißt diagonalisierbar, wenn es eine aus Eigenvektoren von f bestehende Basis von V gibt; bzgl. einer solchen Basis ist die Matrix von f eine Diagonalmatrix Λ*

und in der Diagonalen stehen die Eigenwerte $\lambda_i, i = 1, ..., n$, die zu den entsprechenden Eigenvektoren gehören.
Eine $n \times n$-Matrix $\mathbf{A}$ heißt diagonalisierbar, wenn sie zu einer Diagonalmatrix ähnlich ist; das bedeutet, daß eine Matrix $\mathbf{X} \in GL(n, K)$ existiert, so daß $\mathbf{XAX^{-1}}$ Diagonalmatrix ist. Die Menge $GL(n, \mathbb{R}) := \{\mathbf{A} : \mathbf{A}$ ist nichtsinguläre reelle $n \times n$-Matrix$\}$ heißt allgemeine lineare Gruppe vom Rang n über $\mathbb{R}$.

Zuerst wird Gleichung (3.5) untersucht. Um hier Eindeutigkeit zu gewährleisten, muß vorausgesetzt werden, daß $\mathbf{M}$ invertierbar ist. Da $\mathbf{M} \in \mathbb{R}^{n \times n}$ ist, ist diese Voraussetzung ohne Probleme möglich. Erst dadurch wird eindeutig, daß $h_\sigma = 0$ und $g_\sigma = 0$. Alle sich daraus ergebenden Folgerungen dürfen also nur aufgrund dieser Voraussetzung gemacht werden.

Für die Lösung von Gleichung (3.6) wird zuerst eine passende Zerlegung benötigt.

$$\underbrace{\begin{pmatrix} f_{x'} & f_{u'} \end{pmatrix}}_{=: \mathbf{A}} \cdot \begin{pmatrix} I \\ g_x \end{pmatrix} \cdot h_x = \underbrace{-\begin{pmatrix} f_x & f_u \end{pmatrix}}_{=: \mathbf{B}} \cdot \begin{pmatrix} I \\ g_x \end{pmatrix} \tag{3.7}$$

$\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}, n = d + m$.
Beide Matrizen sind bekannt, da die Ableitungen von f bekannt sind.

Das nächste Ziel ist es, die Zerlegung (3.7) in ein verallgemeinertes Eigenwertproblem umzuformen. Hierbei wird die nichttriviale Lösung der Gleichung $\mathbf{A}x = \lambda \mathbf{B}x$ gesucht.

Für die Umformung wird die Matrix h_x durch ihre Eigenwerte beschrieben. Sei $\mathbf{P}$ die Eigenvektormatrix von h_x und Λ die Diagonalmatrix mit den Eigenwerten, so daß gilt

$$h_x \mathbf{P} = \mathbf{P}\Lambda, \qquad \mathbf{P} \in \mathbb{R}^{d \times d}, \Lambda \in \mathbb{R}^{d \times d}. \tag{3.8}$$

Gleichung (3.7) wird auf beiden Seiten durch $\mathbf{P}$ erweitert.

$$\begin{pmatrix} f_{x'} & f_{u'} \end{pmatrix} \cdot \begin{pmatrix} I \\ g_x \end{pmatrix} \cdot h_x \cdot \mathbf{P} = -\begin{pmatrix} f_x & f_u \end{pmatrix} \cdot \begin{pmatrix} I \\ g_x \end{pmatrix} \cdot \mathbf{P} \tag{3.9}$$

Außerdem wird definiert

$$\mathbf{X} := \begin{pmatrix} I \\ g_x \end{pmatrix} \mathbf{P}, \quad \mathbf{X} \in \mathbb{R}^{n \times d}. \tag{3.10}$$

Nun kann Gleichung (3.7) in eine äquivalente Form gebracht werden mit Hilfe von (3.8) und (3.10):

$$\mathbf{A}\,\mathbf{X}\,\mathbf{\Lambda} = \mathbf{B}\,\mathbf{X}$$

oder

$$\mathbf{AX} \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_d \end{pmatrix} \;=\; \mathbf{BX}$$

Diese Gleichung kann noch weiter verallgemeinert werden. Das Problem wird verlängert, weil man alle Eigenwerte < 1 sortieren will. Für gegebene quadratische Matrizen $\mathbf{A}$ und $\mathbf{B}$ existieren Matrizen $\mathbf{V_1} \in \mathbb{R}^{n \times d}$ und $\mathbf{V_2} \in \mathbb{R}^{n \times m}$ und Diagonalmatrizen $\mathbf{D_{11}} \in \mathbb{R}^{d \times d}$ und $\mathbf{D_{22}} \in \mathbb{R}^{m \times m}$:

$$\mathbf{A} \begin{pmatrix} \mathbf{V_1} & \mathbf{V_2} \end{pmatrix} \begin{pmatrix} \mathbf{D_{11}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D_{22}} \end{pmatrix} \;=\; \mathbf{B} \begin{pmatrix} \mathbf{V_1} & \mathbf{V_2} \end{pmatrix}$$

Diese Form entspricht dem verallgemeinerten Eigenwertproblem und ist mit MATLAB zu lösen. O.B.d.A. werden alle Eigenwerte < 1 angeordnet, so daß $\mathbf{D_{11}} \equiv \Lambda$.

Es wird definiert

$$\underbrace{\begin{pmatrix} \mathbf{P} \\ g_x \mathbf{P} \end{pmatrix}}_{= \mathbf{X}} \;=:\; \mathbf{V_1} = \begin{pmatrix} \mathbf{V_{11}} \\ \mathbf{V_{12}} \end{pmatrix}.$$

Damit folgt nach (3.8):

$$\boxed{h_x = \mathbf{V_{11}}\,\mathbf{\Lambda}\,\mathbf{V_{11}^{-1}}} \quad \text{und} \quad \boxed{g_x = \mathbf{V_{12}}\,\mathbf{V_{11}^{-1}}} \tag{3.11}$$

Da $\mathbf{V_{11}}$ invertierbar sein muß, folgt, daß $\mathbf{P}$ invertierbar sein muß. Dies ist nur möglich, wenn alle Eigenvektoren von h_x linear unabhängig sind. Daraus folgt, daß h_x diagonalisierbar sein muß.
Es lassen sich folglich zwei wichtige Voraussetzungen festhalten, damit die lineare Approximation um den Punkt $(\bar{x}, 0)$ zu vernünftigen Aussagen führt:

(i) Die Matrix $\mathbf{M}$ aus Gleichung (3.5) muß invertierbar sein.

(ii) Die Matrix h_x aus Gleichung (3.6) muß diagonalisierbar sein.

Während Voraussetzung (i) a priori überprüfbar ist, ist dies bei Voraussetzung (ii) nicht möglich. Deshalb wird später im Algorithmus die Invertierbarkeit der Matrix $\mathbf{P}$ geprüft, was damit gleichbedeutend ist.

3.2 Approximation zweiter Ordnung

Bis jetzt wurde eine lineare Approximation der Kontroll- oder Steuerungsfunktion g und der Dynamik h hergeleitet. Dies reicht aber für viele Modelle nicht aus. Wie in SCHMITT-GROHÉ und URIBE [23] angedeutet, führt dies sogar zu fehlerhaften Aussagen. Oft ging man von der falschen Vermutung aus, daß Approximationen höherer Ordnung viel schwieriger zu berechnen seien als jene erster Ordnung und daß mit steigender Zahl der Variablen auch die Rechenzeit exponentiell ansteigen würde.
Daß dies nicht der Fall, ist wird im nächsten Abschnitt deutlich. Beim Herleiten der Terme für die quadratische Approximation sind nur lineare Gleichnungen zu lösen, wodurch die Rechenzeit überschaubar bleibt.

Bemerkung 3.6 *Zu Beginn muß noch einmal auf die verwendete Schreibweise bei den Ableitungen hingewiesen werden. Da eine quadratische Approximation Ableitungen zweiter Ordnung erfordert, muß vorher überlegt werden, wie die Ableitung einer Matrix notiert werden soll. Im folgenden wird die Notation aus* SCHMITT-GROHÉ *und* URIBE *[23] übernommen. Wie in Bemerkung 3.2 schon erwähnt, bedeutet z.B.* $[h_x(\bar{x}, 0)]^j_a$ *das Element in Zeile* j *und Spalte* a *von Matrix* h_x*. Zur Verdeutlichung dient noch das folgende Beispiel.*

Beispiel 3.7 $[f_u]^i_\alpha\,[g_x]^\alpha_\beta\,[h_x]^\beta_j = \sum_{\alpha=1}^{m}\sum_{\beta=1}^{d}\frac{\partial f^i}{\partial u^\alpha}\,\frac{\partial g^\alpha}{\partial x^\beta}\,\frac{\partial h^\beta}{\partial x^j}$.

3.2.1 Herleitung

Man sucht eine Approximation von g und h um den Punkt $(\bar{x},0)$ der folgenden Form

$$\begin{aligned}[g(x,\sigma)]^i &= [g(\bar{x},0)]^i + [g_x(\bar{x},0)]^i_a[(x-\bar{x})]_a + [g_\sigma(\bar{x},0)]^i[\sigma]\\ &\quad+\frac{1}{2}[g_{xx}(\bar{x},0)]^i_{ab}[(x-\bar{x})]_a[(x-\bar{x})]_b\\ &\quad+\frac{1}{2}[g_{x\sigma}(\bar{x},0)]^i_a[(x-\bar{x})]_a[\sigma]\\ &\quad+\frac{1}{2}[g_{\sigma x}(\bar{x},0)]^i_a[(x-\bar{x})]_a[\sigma]\\ &\quad+\frac{1}{2}[g_{\sigma\sigma}(\bar{x},0)]^i[\sigma][\sigma]\end{aligned}$$

und

$$\begin{aligned}[h(x,\sigma)]^j &= [h(\bar{x},0)]^j + [h_x(\bar{x},0)]^j_a[(x-\bar{x})]_a + [h_\sigma(\bar{x},0)]^j[\sigma]\\ &\quad+\frac{1}{2}[h_{xx}(\bar{x},0)]^j_{ab}[(x-\bar{x})]_a[(x-\bar{x})]_b\\ &\quad+\frac{1}{2}[h_{x\sigma}(\bar{x},0)]^j_a[(x-\bar{x})]_a[\sigma]\\ &\quad+\frac{1}{2}[h_{\sigma x}(\bar{x},0)]^j_a[(x-\bar{x})]_a[\sigma]\\ &\quad+\frac{1}{2}[h_{\sigma\sigma}(\bar{x},0)]^i[\sigma][\sigma]\end{aligned}$$

mit $i=1,\ldots,m$ und $a,b,j=1,\ldots,d$.

Unbekannt sind die Terme $[g_{xx}]^i_{ab}$, $[g_{x\sigma}]^i_a$, $[g_{\sigma x}]^i_a$, $[g_{\sigma\sigma}]^i$, $[h_{xx}]^j_{ab}$, $[h_{x\sigma}]^j_a$, $[h_{\sigma x}]^j_a$, $[h_{\sigma\sigma}]^i$. Die anderen Teile sind entweder über die lineare Approximation bekannt oder weil die Ableitungen von f als bekannt vorausgesetzt wurden.
Wie bei der linearen Approximation wird für die Berechnung der fehlenden Terme ganz allgemein die Tatsache ausgenutzt, daß alle Ableitungen von F Null sind, siehe Gleichung (2.15). Im speziellen wird F zweimal in x- und σ-Richtung abgeleitet.

Bemerkung 3.8 *Im folgenden sind die unbekannten Terme für bessere Übersichtlichkeit grün gefärbt.*

$$
\begin{aligned}
[F_{xx}(\bar{x},0)]^i_{jk} &= \left([f_{u'u'}]^i_{\alpha\gamma}[g_x]^\gamma_\delta[h_x]^\delta_k + [f_{u'u}]^i_{\alpha\gamma}[g_x]^\gamma_k + [f_{u'x'}]^i_{\alpha\delta}[h_x]^\delta_k + [f_{u'x}]^i_{\alpha k}\right) \\
&\quad \cdot[g_x]^\alpha_\beta[h_x]^\beta_j \\
&\quad +[f_{u'}]^i_\alpha[g_{xx}]^\alpha_{\beta\delta}[h_x]^\delta_k[h_x]^\beta_j + [f_{u'}]^i_\alpha[g_x]^\alpha_\beta[h_{xx}]^\beta_{fk} \\
&\quad + \left([f_{uu'}]^i_{\alpha\gamma}[g_x]^\gamma_\delta[h_x]^\delta_k + [f_{uu}]^i_{\alpha\gamma}[g_x]^\gamma_k + [f_{ux'}]^i_{\alpha\delta}[h_x]^\delta_k + [f_{ux}]^i_{\alpha k}\right)[g_x]^\alpha_j \\
&\quad +[f_u]^i_\alpha[g_{xx}]^\alpha_{jk} \\
&\quad + \left([f_{x'u'}]^i_{\beta\gamma}[g_x]^\gamma_\delta[h_x]^\delta_k + [f_{x'u}]^i_{\beta\gamma}[g_x]^\gamma_k + [f_{x'x'}]^i_{\beta\delta}[h_x]^\delta_k + [f_{x'x}]^i_{\beta k}\right)[h_x]^\beta_j \\
&\quad +[f_u]^i_\beta[h_{xx}]^\beta_{jk} \\
&\quad +[f_{xu'}]^i_{j\gamma}[g_x]^\gamma_\delta[h_x]^\delta_k + [f_{xu}]^i_{j\gamma}[g_x]^\gamma_k + [f_{xx'}]^i_{j\delta}[h_x]^\delta_k + [f_{xx}]^i_{jk} = 0 \quad (3.12)
\end{aligned}
$$

mit $i = 1,\dots,n; \quad j,k,\beta,\delta = 1,\dots,d; \quad \alpha,\gamma = 1,\dots,m.$

$$
\begin{aligned}
[F_{\sigma\sigma}(\bar{x},0)]^i &= [f_{u'}]^i_\alpha[g_x]^\alpha_\beta[h_{\sigma\sigma}]^\beta \\
&\quad +[f_{u'u'}]^i_{\alpha\gamma}[g_x]^\gamma_\delta[\eta]^\delta_\xi[g_x]^\alpha_\beta[\eta]^\beta_\phi[I]^\phi_\xi \\
&\quad +[f_{u'x'}]^i_{\alpha\delta}[\eta]^\delta_\xi[g_x]^\alpha_\beta[\eta]^\beta_\phi[I]^\phi_\xi \\
&\quad +[f_{u'}]^i_\alpha[g_{xx}]^\alpha_{\beta\delta}[\eta]^\delta_\xi[\eta]^\beta_\phi[I]^\phi_\xi \\
&\quad +[f_{u'}]^i_\alpha[g_{\sigma\sigma}]^\alpha + [f_u]^i_\alpha[g_{\sigma\sigma}]^\alpha \\
&\quad +[f_{x'}]^i_\beta[h_{\sigma\sigma}]^\beta \\
&\quad +[f_{x'u'}]^i_{\beta\gamma}[g_x]^\gamma_\delta[\eta]^\delta_\xi[\eta]^\beta_\phi[I]^\phi_\xi \\
&\quad +[f_{x'x'}]^i_{\beta\delta}[\eta]^\delta_\xi[\eta]^\beta_\phi[I]^\phi_\xi = 0 \qquad (3.13)
\end{aligned}
$$

mit $i = 1,\dots,n; \quad \alpha,\gamma = 1,\dots,m; \quad \beta,\delta = 1,\dots,d$ und $\phi,\xi = 1,\dots,d_2$.

Bei der folgenden Ableitung wurden alle Terme, die g_σ oder h_σ enthalten, von vornherein weggelassen, da sie gleich Null sind, siehe Kapitel 3.1.

$$
\begin{aligned}
[F_{\sigma x}(\bar{x},0)]^i_{jk} &= [f_{u'}]^i_\alpha[g_x]^\alpha_\beta[h_{\sigma x}]^\beta_j + [f_{u'}]^i_\alpha[g_{\sigma x}]^\alpha_\gamma + [f_u]^i_\alpha[g_{\sigma x}]^\alpha_j + [f_{x'}]^i_\beta[h_{\sigma x}]^\beta_j \\
&= 0 \qquad (3.14)
\end{aligned}
$$

mit $i = 1,\dots,n; \quad j,\beta,\gamma = 1,\dots,d; \quad \alpha = 1,\dots,m.$

Bemerkung 3.9 *Bei allen drei Gleichungssystemen wird die Annahme ausgenutzt, daß im Gleichgewicht keine Störung existiert und folglich ein deterministisches Problem betrachtet wird. Die Verteilung der Zufallsvariablen aus den Annahmen in Kapitel 2.3*

taucht in Gleichungssystem (3.13) auf, denn hier spielt die angenommene Varianz/Kovarianzmatrix I von ε eine Rolle.

Folgerung 3.10

1. *In Gleichung (3.12) gibt es $n \times d \times d$ Unbekannte $n \times d \times d$ Gleichungen. Dieses System ist mit einem linearen Gleichungslöser lösbar und man bekommt die Terme g_{xx} und h_{xx}.*

2. *In Gleichungssystem (3.13) existieren n Unbekannte in n Gleichungen, die ebenso mit einem linearen Gleichungslöser berechnet werden können.*

3. *In der letzten Gleichung (3.14) gibt es $n \times d$ Unbekannte durch die beiden Matrizen $g_{\sigma x}$ und $h_{\sigma x}$ in $n \times d$ Gleichungen. Es läßt sich einfach erkennen, daß -die Existenz einer eindeutigen Lösung vorausgesetzt- gilt*

$$g_{\sigma x} = 0 \quad \textit{und} \quad h_{\sigma x} = 0.$$

3.2.2 Existenz und Eindeutigkeit

Die eindeutige Lösbarkeit der drei Gleichungssysteme (3.12), (3.13) und (3.14), ergibt sich aus folgender Tatsache: In den einzelnen Summanden, welche die unbekannten Koeffizienten enthalten, muß der Multiplikator bzw. das Produkt der Multiplikatoren dieses Koeffizienten vollen Rang haben. Die Aussage, daß es genauso viele Unbekannte wie Gleichungen gibt, reicht also nicht aus. Die Eindeutigkeit ist hier nicht direkt durch Umformungen überprüfbar.
Später bei der Implementierung geschieht dies allerdings indirekt: Dabei wird das dreidimensionale Gleichungssystem in ein zweidimensionales System

$$q = Q \cdot x \tag{3.15}$$

überführt. Man muß sich also jetzt nur noch auf die Lösbarkeit dieser Gleichung zu konzentrieren. Damit sie eindeutig lösbar ist, muß Q invertierbar sein.

Bemerkung 3.11 *Die MATLAB-Routine überprüft dies nicht direkt, aber da die Inverse von Q benutzt wird, würde es zu einer Fehlermeldung kommen, wenn sie nicht existierte. Die Eindeutigkeit wird also indirekt von der Routine überprüft.*

3.3 Zusammenfassung und Ausblick

3.3.1 Theoretisches Resultat

Das beschriebene Verfahren läßt sich in vier Schritte kurz zusammenfassen:

Algorithmus 3.12

Schritt 1: *Gleichgewichtspunkt $(\bar{x}, \bar{u})$ finden.*

Schritt 2: *Berechne $g_x, h_x, g_\sigma, h_\sigma$:*

- *Leite Gleichung (2.14) nach x und σ ab*
- *Umformungen und Berechnung nach (3.11)*

Schritt 3: *Berechne $g_{xx}, h_{xx}, g_{\sigma\sigma}, h_{\sigma\sigma}$:*

- *Leite Gleichung (2.14) nach x und σ zweimal ab*
- *Umformungen und Berechnung mit Hilfe von (3.15)*

Schritt 4: *Setze alle Terme zu einer* TAYLOR*-Approximation (3.12) und (3.12) zur Bestimmung von $h(x, \sigma)$ und $g(x, \sigma)$ zusammen*

Aus der Approximation ergeben sich bestimmte Folgerungen für das stochastische Modell, die im folgenden kurz gezeigt werden sollen.
In den letzten beiden Abschnitten wurden die Terme für eine quadratische Approximation von $g(x, \sigma)$ und $h(x, \sigma)$ hergeleitet. Daraus ergibt sich der folgende

Satz 3.13 *Das Modell sei gegeben durch Gleichung (2.1) mit den Lösungen (2.11) und (2.12). Dann gilt für die Approximation dieser Lösungen*

$$g_\sigma(\bar{x}, 0) = 0, \qquad h_\sigma(\bar{x}, 0) = 0,$$
$$g_{x\sigma}(\bar{x}, 0) = 0, \qquad h_{x\sigma}(\bar{x}, 0) = 0.$$

Nach SCHMITT-GROHÉ und URIBE[23] folgt aus diesem Resultat, daß bei Approximationen erster Ordnung die erwarteten Werte für den Zustandsvektor x_t und den Kontrollvektor u_t mit denen der nicht-stochastischen Gleichgewichtswerte $\bar{x}$ und $\bar{u}$ übereinstimmen. Man braucht also bei linearen Approximationen den konstanten Term nicht um die Größe der Varianz eines externen Schocks zu erweitern.

Bemerkung 3.14 *Dies ist eine wichtige Beschränkung von Approximationen erster Ordnung. In vielen Anwendungen ist besonders der Einfluß von Unsicherheit von Interesse.* SCHMITT-GROHÉ *und* URIBE *nennen in [24] ein Beispiel auf welche Weise Unsicherheit Wohlfahrt beeinflußt. Hierbei würden die Approximationen von zwei unterschiedlich riskanten Steuerfunktionen, die auf dasselbe Gleichgewicht hinauslaufen, dasselbe Wohlfahrtsniveau erreichen. Man kann also Unterschiede in der Unsicherheit mit linearen Approximationen hier gar nicht darstellen.*

Aufgrund dieser Probleme folgt die Notwendigkeit von Approximationen höherer Ordnung. Aus Satz 3.13 ergeben sich weitere Ergebnisse für die Approximation zweiter Ordnung. Die quadratische Approximation einer Steuerungsfunktion eines stochastischem Modells unterscheidet sich demnach von ihrem deterministischen Gegenpart nur im Term $\frac{1}{2}g_{\sigma\sigma}\,\sigma^2$. Genauso unterscheiden sich die Approximation des Zustandsvektors in einem stochastischen und deterministischen Modell nur durch den Faktor $\frac{1}{2}h_{\sigma\sigma}\,\sigma^2$.
Folglich beeinflußt Unsicherheit in einem Modell bei quadratischen Approximationen nur einen konstanten Term der Entscheidungsregeln.

3.3.2 Approximationen höherer Ordnung

Um noch höhere Genauigkeit zu erzielen, bietet es sich an, über Approximationen n-ter Ordnung nachzudenken. In [23] erklären SCHMITT-GROHÉ und URIBE, daß man die Terme der linearen und quadratischen Approximation benutzen kann, um eine Approximation dritter Ordnung zu erhalten. Dies geschieht nach dem gleichen Prinzip wie bei den ersten beiden Approximationen. Auch müssten hierbei wie bisher nur lineare Gleichungen gelöst werden. Die Autoren folgern, daß man so sukzessive Approximationen

immer höherer Ordnung bilden kann, indem sowohl die Terme der niedrigeren Stufen benutzt werden als auch die Ableitungen höherer Ordnung von f an der Stelle $(\bar{x}, \bar{u})$.

3.3.3 Weitere Anwendungsmöglichkeiten

Bisher war die Basis aller Überlegungen, daß das Verfahren einen Gleichgewichtspunkt als Ausgangspunkt benutzt. Was aber passiert, wenn dieser Ausgangspunkt geändert wird? Welche Möglichkeiten ergeben sich, wenn kein Gleichgewicht existierte oder man einen anderen Punkt als Startpunkt benutzen wollte?
Tatsächlich erwähnen SCHMITT-GROHÉ und URIBE in [24], daß es grundsätzlich möglich sei, das beschriebene Verfahren nicht nur in Gleichgewichtspunkten, sondern auch an beliebigen anderen Punkten (x, σ) anzuwenden. Die Voraussetzung hierzu ist allerdings, daß die Ableitungen von $F(x, \sigma)$ an diesem Punkt bekannt sein müssen. Bisher wurde das Vorgehen dadurch vereinfacht, daß $F_{x^k \sigma^j} = 0 \quad \forall x, \sigma, k, j$ galt. Dies konnte aufgrund der Gleichgewichtsgleichung (2.1) hergeleitet werden. An einem beliebig gewählten Punkt kann diese nicht benutzt werden, weswegen die Voraussetzung eine Einschränkung darstellt. Wenn aber die Ableitungen an dem ausgesuchten Punkt bekannt sind, wird das Verfahren genauso funktionieren wie in einem Gleichgewicht. Ein weiterer Ansatz für Überlegungen ist, inwieweit hier numerische Approximationen der Ableitungen sinnvoll sein können.

4 Numerische Lösung des Verfahrens

Zur numerischen Lösung des von SCHMITT-GROHÉ und URIBE beschriebenen Verfahrens präsentieren die Autoren im Internet verschiedene Programme in Matlab, die die Methode implementieren, siehe SCHMITT-GROHÉ und URIBE [25]. Der Matlab-Code besteht aus mehreren m-Files. Einige davon bilden die Basis und werden bei jeder Ausführung benötigt wie die unten beschriebenen Files `gxhx.m`, `anal_deriv.m`, `num_eval.m`, `gxx_hxx.m` und `gss_hss.m`. Diese müssen in der Regel nicht geändert werden. Andere beziehen sich speziell auf die zu untersuchenden Beispiele und sind je nach Beispiel individuell abzuändern.
Die Programme benötigen die Toolbox SYMBOLIC MATH. Diese Toolbox verwendet symbolische Objekte als Datentypen, um Variablen, Matrizen und andere Ausdrücke symbolisch darzustellen. Außerdem bietet die Toolbox direkten Zugang zu Funktionen aus MAPLE.
Im Folgenden werden die einzelnen Programme vorgestellt, sowie ihr Aufbau und ihre Funktion erläutert.

4.1 Die elementaren Programmteile

anal_deriv.m

Mit diesem Programmteil werden analytisch die ersten und zweiten Ableitungen in x- und u- Richtung der Funktion f aus der Gleichgewichtsgleichung (2.1) berechnet. Es werden hier also keine numerischen Werte berechnet, sondern die Ableitungen als symbolische Objekte behandelt. Zur Erläuterung von symbolischen Objekten sei auf die Hilfe-Funktion von MATLAB zu der SYMBOLIC MATH Toolbox verwiesen. Das Pro-

gramm `anal_deriv.m` wird von den programmierten Modell-Beispielen jeweils an entsprechender Stelle aufgerufen.

Bemerkung 4.1 *Im Programm wird der Kontrollvektor u mit y bezeichnet. Im Folgenden wird wegen der Einheitlichkeit weiter die Bezeichnung u verwendet.*

INPUT: An die Funktion müssen die Variablen x, u, x', u' sowie der Parameter *appr.* und die Funktion f übergeben werden.

- Wenn Anzahl der Input-Variablen =5,

  so setze appr.= 2
- Wenn appr.=2

  Berechne alle Ableitungen bis Ordnung 2
- Ansonsten

  Setze alle Ableitungen zweiter Ordnung auf 0
- Gebe alle Ableitungen zurück

OUTPUT: $f_x,\ f_u,\ f_{x'},\ f_{u'},\ f_{xu},\ f_{xu'},\ f_{xx},\ f_{xx'},\ f_{x'u},\ f_{x'u'},$
$f_{x'x},\ f_{x'x'}\ f_{uu},\ f_{uu'},\ f_{ux},\ f_{ux'},\ f_{u'u},\ f_{u'u'},\ f_{u'x},\ f_{u'x'}$

Bemerkung 4.2 *Die ersten und zweiten Ableitungen von f werden jeweils mit Jacobi-Matrizen berechnet. Dies funktioniert mit dem jacobian-Befehl aus der Symbolic-Math Toolbox. Bei der zweiten Ableitung wird außerdem der reshape-Befehl benutzt, um die Ergebnisse in einen Tensor mit den passenden Dimensionen zu übertragen.*

num_eval.m

Dieses Programm wertet die Ableitungen von f im Gleichgewichtspunkt $(\bar{x}, \bar{u})$ aus. Es wird vorausgesetzt, daß die benötigten Parameter schon im Workspace von Matlab vorhanden sind. Es handelt sich dabei um den Gleichgewichtspunkt, die analytischen Ableitungen von f, sowie um alle anderen Parameter, die Argumente von f darstellen. Dies

bedeutet, daß die entsprechenden Programme - insbesondere auch `anal_deriv.m` - vor `num_eval.m` aufgerufen werden müssen, damit diese Werte belegt sind.

`Num_eval.m` hat keine direkte Eingabe, da es auf alle benötigten Input-Werte automatisch zurückgreift. Die Vorgehensweise läßt sich einfach zusammenfassen:

INPUT: `keine direkte Eingabe`
`oder Übergabe von Werten an eine Funktion`

- `Für alle ersten und zweiten Ableitungen von f`
  `belege einen array von der Größe`
  `der analytischen Ableitung mit Nullen.`
- `Werte diesen array an jeder Stelle`
  `nach entweder` $\bar{x}$ `oder` $\bar{u}$ `aus`
  `und gebe diesen Wert aus. STOP`

OUTPUT: f^n_x, f^n_u, $f^n_{x'}$, $f^n_{u'}$, f^n_{xu}, $f^n_{xu'}$, f^n_{xx}, $f^n_{xx'}$, $f^n_{x'u}$, $f^n_{x'u'}$, $f^n_{x'x}$, $f^n_{x'x'}$, f^n_{uu}, $f^n_{uu'}$, f^n_{ux}, $f^n_{ux'}$, $f^n_{u'u}$, $f^n_{u'u'}$, $f^n_{u'x}$, $f^n_{u'x'}$

Bemerkung 4.3 *f^n bezeichnet die numerische Auswertung der Ableitung.*

Durch dieses File sind also die numerischen Werte aller Ableitungen bekannt und stehen insbesondere bei der Berechnung von gx, hx, gxx, hxx etc. zur Verfügung.

4.2 Programmteil für die lineare Approximation

gxhx.m

Dieser Programmteil berechnet die Matrizen g_x und h_x für die lineare Approximation des Modells über eine verallgemeinerte Eigenwertzerlegung. Der theoretische Hintergrund hierzu wird in Kapitel 3.1.2 erläutert und die dort verwendete Notation wird auch hier benutzt.

INPUT: Benötigt die Ableitungen
im Gleichgewicht $f^n_{u'}$, f^n_u, $f^n_{x'}$,f^n_x,
sowie den Parameter $stake$ als Übergabe

- Wenn Anzahl der Input-Variablen < 5, so setze stake=1
- Setze $\mathbf{A} = [-f_{x'}, -f_{u'}]$ und $\mathbf{B} = [f_x, f_u]$
- Produziere Diagonalmatrix $\mathbf{D}$
  und eine volle Matrix $\mathbf{V}$,
  so daß gilt:

$$\mathbf{A} * \mathbf{V} = \mathbf{B} * \mathbf{V} * \mathbf{D}$$

- Suche alle Diagonaleinträge von $\mathbf{D} < stake$
- Finde Anzahl dieser Einträge
  - Falls Anzahl < d,
    - so gebe Warnung aus:
      Kein lokales Gleichgewicht.STOP
  - Falls Anzahl > d,
    - so gebe Warnung aus:
      Gleichgewicht ist nicht stabil.STOP

* Ansonsten:
  - Bestimme die Matrizen $\mathbf{V_1}, \mathbf{D_{11}}$ und $\mathbf{P}$
    Falls Rang $\mathbf{P} < d$, gebe Warnung aus:
    $\mathbf{P}$ nicht invertierbar.STOP
    Ansonsten:
      Berechne g_x und h_x

OUTPUT: g_x, h_x

Einige Matrizen haben im Theorieteil andere Bezeichnungen als im Programmteil, daher hier eine Gegenüberstellung:

Name im Theorieteil	Name im Programm
$\mathbf{V_1}$	Vs
$\mathbf{D_{11}}$	la
$\mathbf{P}$	P

Tabelle 4.1: Matrizenbezeichnung im Programm `gxhx.m`

Bemerkung 4.4 *Bei den Eingabeparametern handelt es sich bereits um die ausgewerteten Ableitungen an der Stelle* $(\bar{x}, \bar{u})$ *und nicht um die Ableitung als Funktion. Um diese Auswertung zu bekommen, benötigt man vorher die Programmteile* `anal_deriv.m` *sowie* `num_eval.m`*. Diese werden nicht in* `gxhx.m` *aufgerufen, sondern im Hauptprogramm.*

Bemerkung 4.5 *Der Parameter* $stake$ *sortiert alle Eigenwerte* $\lambda < 1$ *aus. Dies ist nötig, da bei den Modell-Annahmen in Kapitel (2.3) vorausgesetzt wurde, daß alle Eigenwerte der Matrix* h_x *<1 sein sollen.*

Bemerkung 4.6 *Außer der hier beschriebenen Methode haben die Autoren auch noch eine weitere Methode implementiert, die auf einer* SCHUR*-Zerlegung basiert. Das dazugehörige Programm* `gx_hx.m` *findet sich ebenfalls auf der Homepage von* SCHMITT-GROHÉ.

4.3 Programmteil für die quadratische Approximation

gxx_hxx.m

Dieser Teil wird bei der quadratischen Approximation benötigt, da hier die dreidimensionalen Felder gxx und hxx bestimmt werden.
Die Grundlage für das Programm ist das Gleichungssystem (3.12) aus Kapitel 3.2.1.

Ziel ist es, zuerst das dreidimensionale Gleichungssystem in ein zweidimensionales zu überführen und dieses danach zu lösen. Deshalb ist das Programm in zwei Hauptteile gegliedert:

Der erste Teil formt das oben genannte Gleichungssystem um in ein Gleichungssystem der Form

$$q = Q * x. \tag{4.1}$$

Dabei ist x der Vektor, der die unbekannten Elemente gxx und hxx enthält, also die grün gefärbten Terme in (3.12). Q und q sind respektive eine Matrix und ein Vektor, die die bekannten Elemente aus dem Gleichungssystem enthalten. Dieses wird nach Termen unterteilt. Ein Term ist jeweils ein Summand der Gleichung.
Die bekannten Werte werden entweder im Vektor q gespeichert, wenn in dem entsprechenden Term der Gleichung kein unbekanntes Element gxx oder hxx auftaucht, oder im anderen Fall in der Matrix Q.
Der erste Teil arbeitet das Gleichungssystem (3.12) termweise ab; jeder Term wird einzeln betrachtet, das heißt der erste Programmteil ist nochmal in acht kleinere Abschnitte unterteilt.

Im zweiten Teil wird die Gleichung (4.1) gelöst. Dazu wird eine Unterfunktion `temp` aufgerufen, die das Verfahren beschleunigt. Die Funktion kreiert eine Matrix A, so daß gilt:

$$x = A * \tilde{x}.$$

Der Vektor x enthält die unbekannten Elemente von gxx und hxx. Bei dem Vektor $\tilde{x}$ wird die Symmetrie von gxx und hxx ausgenutzt, d. h. bestimmte Elemente werden nur einmal gespeichert. Dies führt dazu, daß hinterher nicht die Matrix Q invertiert wird, sondern eine Matrix $\tilde{Q}$ mit dim $\tilde{Q}$ < dim Q, was Rechenzeit einspart.

Bemerkung 4.7 *Zur Verdeutlichung: Im folgenden Algorithmus bezeichnet gxx zwei verschiedene Dinge. Einmal wird der Name des Feldes gxx so bezeichnet und einmal ist damit ein Wert gemeint, der zum Abzählen des Speicherplatzes benutzt wird.*

INPUT: Alle ersten und zweiten Ableitungen von f im Gleichgewichtspunkt sowie die zur linearen Approximation nötigen Terme g_x und h_x müssen an die Funktion übergeben werden:
f^n_x, f^n_u, $f^n_{x'}$, $f^n_{u'}$, f^n_{xu}, $f^n_{xu'}$, f^n_{xx}, $f^n_{xx'}$, $f^n_{x'u}$, $f^n_{x'u'}$,
$f^n_{x'x}$, $f^n_{x'x'}$, f^n_{uu}, $f^n_{uu'}$, f^n_{ux}, $f^n_{ux'}$, $f^n_{u'u}$, $f^n_{u'u'}$, $f^n_{u'x}$, $f^n_{u'x'}$, g_x, h_x

- Belege die dreidimensionalen Felder gxx und hxx der Größe $m \times d \times d$ bzw. $d \times d \times d$ mit Nullen, belege Speicherplatz für q und Q mit Nullen, Setze m=0.
- Für alle Zeilen i=1:n
  für alle Spalten j=1:d
  für alle Seiten k=1:j
  m=m+1
  - Speichere in Zeile m von q alle bekannten Werte aus den Termen 1 bis 8 des Gleichungssystems (3.12), in denen gxx und hxx nicht vorkommen.
    Ist $q(m) \neq 0$, so setze
    $q(m) := q(m) + neu$.

* Speichere in Zeile m von Q
  alle bekannten Werte aus den Termen 1 bis 8
  des Gleichungssystems (3.12), in denen gxx und
  hxx vorkommen.
  Speichere dabei die zu gxx gehörigen Werte
  in Zeile m bis Spalte Nummer gxx,
  speichere die zu hxx gehörigen Werte
  in Zeile m ab Spalte Nummer gxx+1.
  Ist $Q(m) \neq 0$, so setze
  $Q(m) := Q(m) + neu$.

STOP k, STOP j, STOP i

- Löse das Gleichungssystem $q = Q * x$
  * Rufe die Unterfunktion temp auf $\Rightarrow x = A * \tilde{x}$
  * Setze $\tilde{Q} = Q * A$
  * Berechne $\tilde{x} = (-\tilde{Q})^{-1} * q$
  * Berechne $x = A * \tilde{x}$
  * Speichere die Elemente aus x
    bis Nummer gxx in dem Feld gxx,
    speichere die restlichen Elemente aus x
    in dem Feld hxx.

OUTPUT: g_{xx}, h_{xx}

gss_hss.m

In diesem Teil werden die Vektoren gss und hss berechnet, die neben gxx und hxx bei der quadratischen Approximation benötigt werden. Das 's' bezieht sich auf den griechischen Buchstaben σ.

Dieses Programm besteht hier nur aus einem Teil. Die Grundlage ist das Gleichungssystem (3.13) aus Abschnitt 3.2.1. Der Aufbau ist ähnlich wie bei dem Programm gxx_hxx.m. Auch hier ist es das Ziel, alle Elemente in ein lineares Gleichungssystem der Form $q = Q * x$ zu übertragen und dieses zu lösen. Die Routine arbeitet das betrachtete System termweise über neun kleinere Unterabschnitte ab. Der Aufwand ist insgesamt geringer als bei gxx_hxx.m, da es sich bei $[F_{\sigma\sigma}(\bar{x}, 0)]^i$ um einen Vektor handelt, und nicht um einen Tensor, das heißt es gibt nur eine for-Schleife.

INPUT: Alle ersten und zweiten Ableitungen
von f im Gleichgewichtspunkt,
sowie die lineare Approximation von g_x und h_x,
die quadratische Approximation von g_{xx} sowie η
müssen an die Funktion übergeben werden:
f^n_x, f^n_u, $f^n_{x'}$, $f^n_{u'}$, f^n_{xu}, $f^n_{xu'}$, f^n_{xx}, $f^n_{xx'}$, $f^n_{x'u}$, $f^n_{x'u'}$,
$f^n_{x'x}$, $f^n_{x'x'}$, f^n_{uu}, $f^n_{uu'}$, f^n_{ux}, $f^n_{ux'}$, $f^n_{u'u}$, $f^n_{u'u'}$, $f^n_{u'x}$, $f^n_{u'x'}$, g_x, h_x, g_{xx}, η

- Für alle Zeilen i=1:n
  - Speichere in Zeile i von q
    alle Werte aus den Termen 1 bis 9
    des Gleichungssystems,in denen
    gss und hss nicht vorkommen.
    Ist $q(i) \neq 0$, so setze
    $q(i) := q(i) + neu$.

* Speichere in Zeile i von Q
  alle bekannten Werte aus Term 1 bis 9,
  in denen gxx und hxx vorkommen.
  Speichere die zu gxx gehörigen Werte in Qg,
  speichere die zu hxx gehörigen Werte in Qh.
  Ist $Qg(i) \neq 0$, so setze
  $Qg(i) := Qg(i) + neu$.
  Ist $Qh(i) \neq 0$, so setze
  $Qh(i) := Qh(i) + neu$.

STOP i

- Lösen des Gleichungssystem
  $q = Q * x$ mit $Q = [Qg\, Qh]$
  * Löse $x = -(Q)^{-1} * q$
  * Speichere die Elemente aus x
    bis Nummer n_y in dem Vektor gss,
    speichere die restlichen Elemente aus x
    in dem Vektor hss.

OUTPUT: g_{ss}, h_{ss}

Bemerkung 4.8 *Im letzten Schritt wird die Matrix* **Q** *invertiert. Ist dies nicht möglich, so gibt es keine eindeutige Lösung, vergleiche dazu Bemerkung 3.11.*

Bemerkung 4.9 *Diese beiden Programmteile können aufgrund ihrer Konstruktion erst ganz am Schluß aufgerufen werden, da sie als Eingabe die Ergebnisse aller anderen Routinen benötigen.*

5 Diskussion der praktischen Ergebnisse

Dieses Kapitel ist in drei Abschnitte unterteilt und beschäftigt sich mit der praktischen Anwendung des Verfahrens am Beispiel eines neoklassischen Modells.
Im ersten Abschnitt wird das Beispiel vorgestellt. Außerdem werden die Optimalitätsbedingungen aus Satz 2.12 darauf angewendet und die entsprechenden Umformungen dargestellt. Dann erfolgt die Berechnung des Gleichgewichts und schließlich eine Beschreibung, wie das Beispiel in Matlab implementiert wurde. Die Ergebnisse des Verfahrens werden im zweiten Abschnitt vorgestellt und anhand von Graphiken veranschaulicht und diskutiert. Der dritte Abschnitt untersucht eine abgeänderte Version des Beispiels, um zu zeigen, wie sich Änderungen in der Zielfunktion auf Koeffizienten der Approximation auswirken.

5.1 Ein neoklassisches Modell, Beispiel 1

5.1.1 Einführung

Das hier betrachtete Beispiel ist ein stochastisches volkswirtschaftliches Wachstumsmodell aus GRÜNE [11]. Das Modell ist zweidimensional im Zustand x und hat eine eindimensionale Kontrolle u.
Hierfür soll die folgende Zielfunktion betrachtet werden:

$$\max \sum_{t=0}^{\infty} \beta^t \ \log u_t \tag{5.1}$$

Das Kontrollsystem lautet:

$$\varphi(x,u,z) \ = \ \begin{pmatrix} A\, e^{x_2}\, x_1^{\alpha} - u \\ \rho x_2 + z \end{pmatrix} \tag{5.2}$$

Dieses Problem ist aus zwei Gründen sehr gut als Testproblem geeignet:

1. Die Lösung ist genügend glatt für $x_1 \neq 0$.

2. Die exakte Lösung ist für die optimale Wertefunktion bekannt.

Sie ist gegeben durch die Gleichung

$$V(x) = B \; + \; C \; \ln x_1 \; + \; D \, x_2 \tag{5.3}$$

mit

$$\begin{aligned} B &= \frac{\ln((1-\beta\alpha)\,A) + \frac{\beta\alpha}{1-\beta\alpha}\ln(\beta\alpha A)}{(1-\beta)} \\ C &= \frac{\alpha}{1-\alpha\beta} \\ D &= \frac{1}{(1-\alpha\beta)(1-\rho\beta)}. \end{aligned}$$

Die exakte Lösung für die Steuerung u ist gegeben durch

$$u(x) = (1-\alpha\beta)\,A\,e^{x_2}\,x_1^{\alpha}. \tag{5.4}$$

Die folgenden Berechnungen werden durchgeführt mit
$A = 5$, $\alpha = 0.34$, $\beta = 0.95$ und $\rho = 0.9$.
Die Zufallsvariable z ist Gauß-verteilt mit Mittelwert Null und Standard-Abweichung $\sigma = 0.008$. Diese Variable kann durch ihre Varianz σ^2 so normalisiert werden, daß $\frac{z}{\sigma} \sim N(0,1)$ gilt. Daraus folgt $z = \sigma \cdot \varepsilon$. Hierbei ist ε unabhängig von x und eine $N(0,1)$-Zufallsvariable. Zur Umformung siehe auch ÅSTRÖM [1].
Durch diese Umformung erhält man die Form aus dem Modell von SCHMITT-GROHÉ und URIBE. Die Variable x_1 folgt einer deterministischen Dynamik, die Variable x_2 folgt einer stochastischen Dynamik. Hierbei werden Variablen aus der Zeitperiode $(t+1)$ aus Notationsgründen mit einem $'$ gekennzeichnet, die Variablen aus der Zeitperiode t erhalten keine Kennzeichnung:

$$\begin{aligned} x_1^{'} &= A\,e^{x_2}x_1^{\alpha} - u \\ x_2^{'} &= \rho x_2 + \eta\sigma\varepsilon, \quad \eta = 1 \end{aligned}$$

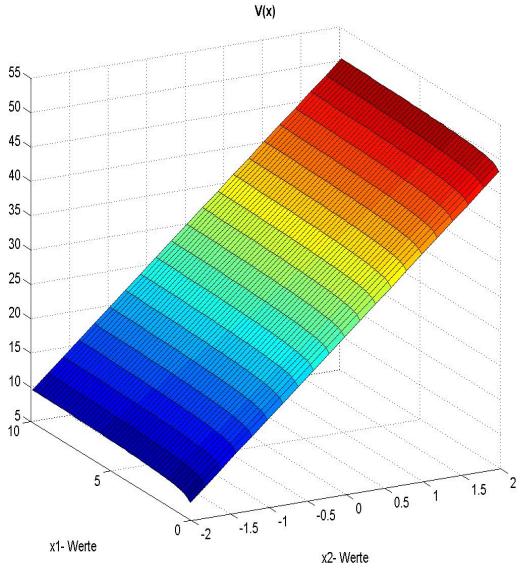

Abbildung 5.1: Die exakte optimale Wertefunktion auf dem Intervall $\Omega = [0.1, 10] \times [-2, 2]$ zur Schrittweite $h = 0.1$.

Durch die Eigenschaften dieses Beispiels ist nicht nur eine exakte Berechnung der optimalen Wertefunktion möglich, sondern auch eine exakte Fehlerbestimmung. So können die Ergebnisse des Verfahrens einfach verglichen werden. Um das Verfahren anwenden zu können, muß das Beispiel in eine Gleichgewichtsgleichung (2.1) überführt werden. Hierzu benötigt man die HAMILTON-Funktion und die Bedingungen aus Satz 2.12:

$$\begin{aligned} x_{t+1} &= \left(\frac{\partial H^t}{\partial \lambda_{t+1}}\right)^T && \text{Zustandsgleichung} \\ \lambda_t &= \left(\frac{\partial H^t}{\partial x_t}\right)^T && \text{Kozustandsbedingung} \\ 0 &= \left(\frac{\partial H^t}{\partial u_t}\right)^T && \text{Maximumbedingung} \end{aligned}$$

Die HAMILTON-Funktion für das betrachtete Beispiel lautet:

$$H^t(x, u, \lambda) = \beta \log u + \lambda_1' \cdot (A e^{x_2} x_1^\alpha - u) + \lambda_2' \cdot \rho x_2.$$

Bemerkung 5.1 *Der stochastische Anteil aus der zweiten Nebenbedingung wurde in der* HAMILTON*-Funktion weggelassen. Die* HAMILTON*-Funktion wird für das Gleichgewicht aufgestellt und hier spielt dieser Anteil nach Voraussetzung keine Rolle.*

Die notwendigen Optimalitätsbedingungen ergeben sich wie folgt:
Zustandsbedingungen:

$$\begin{aligned} x_1^{'} &= A\, e^{x_2}\, x_1^{\alpha} \\ \Longrightarrow 0 &= x_1^{'} - A\, e^{x_2}\, x_1^{\alpha} + u \end{aligned} \tag{5.5}$$

und

$$\begin{aligned} x_2^{'} &= \rho x_2 \\ \Longrightarrow 0 &= x_2^{'} - \rho x_2 \end{aligned} \tag{5.6}$$

Kozustandsbedingung:

$$\lambda_1 = \lambda_1^{'} \alpha A\, e^{x_2}\, x_1^{\alpha-1} \tag{5.7}$$

$$\lambda_2 = \lambda_2^{'} \rho + \lambda_1^{'} (A e^{x_2} x_1^{\alpha}) \tag{5.8}$$

Maximumbedingung:

$$\begin{aligned} \beta \frac{1}{u} - \lambda_1^{'} &= 0 \\ \Longrightarrow \frac{\beta}{u} &= \lambda_1^{'} \end{aligned} \tag{5.9}$$

Wenn man nun die Maximumbedingung in die Kozustandsbedingung einsetzt, so ergibt sich nach Umformung folgende Gleichung:

$$0 = \beta \frac{1}{u^{'}}\, \alpha\, A\, e^{x_2^{'}}\, x_1^{'(\alpha-1)} - \frac{1}{u} \tag{5.10}$$

Bemerkung 5.2 *Durch die Umformungen erhält man nicht exakt die Funktion f, so wie sie in Abschnitt 2.2.2 definiert wurde. Aber es wird eine dazu äquivalente Funktion $\widetilde{f}$ aufgestellt, die alle benötigten Optimalitätsbedingungen enthält.*

Also ergibt sich als Gleichgewichtsbedingung:

$$\widetilde{f} := \begin{pmatrix} x_1^{'} - A\, e^{x_2}\, x_1^{\alpha} + u \\ x_2^{'} - \rho\, x_2 \\ \beta\, \frac{1}{u^{'}}\, \alpha\, A\, e^{x_2^{'}}\, x_1^{'(\alpha-1)} - \frac{1}{u} \end{pmatrix} = 0 \tag{5.11}$$

Bemerkung 5.3 *$\widetilde{f}$ enthält nicht die zweite Kozustandsbedingung 5.8. Es kann eine Funktion $\widetilde{f}'$ aufgestellt werden, die auch diese Bedingung enthält. Allerdings ergeben sich für das Kontrollproblem dadurch keine neuen Informationen, da Gleichung (5.8) die Dynamik für λ_2 bestimmt. Diese wird für den weiteren Verlauf nicht benötigt. $\widetilde{f}$ und $\widetilde{f}'$ haben die gleiche Nullstellenmenge, deshalb wird $\widetilde{f}$ verwendet.*

Das nächste Ziel ist es, das Gleichgewicht für dieses Beispiel zu berechnen, weil es als Eingabe für die Programme benötigt wird und den Ausgangspunkt für das Verfahren bildet. Dazu wird Gleichung (5.11) benutzt und die Überlegung, daß im Gleichgewicht

$$\begin{aligned} x_1^{'} &= x_1, \\ x_2^{'} &= x_2, \\ u^{'} &= u \end{aligned}$$

gilt. Aus

$$x_2^{'} - \rho\, x_2 = 0$$

folgt

$$\bar{x}_2 = 0.$$

Dieses Ergebnis eingesetzt in Zeile eins von (5.11) ergibt die Gleichung

$$u = A \cdot 1 \cdot x_1^{\alpha} - x_1. \tag{5.12}$$

Verwendet man dies in Zeile drei von (5.11) so erhält man

$$\begin{aligned} & \beta\,\alpha\,A\,x_1^{(\alpha-1)}\,\frac{1}{A\,x_1^{\alpha}-x_1} = \frac{1}{A\,x_1^{\alpha}-x_1} \\ \Longleftrightarrow \quad & \beta\,\alpha\,A\,x_1^{(\alpha-1)} = 1 \\ \Longleftrightarrow \quad & \beta\,\alpha\,A = x_1^{(1-\alpha)} \\ \Longleftrightarrow \quad & \Big(\ \beta\,\alpha\,A\ \Big)^{\frac{1}{1-\alpha}} = x_1 \end{aligned}$$

In der letzten Zeile sind alle Werte bekannt und x_1 kann berechnet werden:

$$\begin{aligned} \Big(\ 0.95 \cdot 0.34 \cdot 5\ \Big)^{\frac{1}{0.66}} &= x_1 \\ x_1 &\approx 2.067344815 \end{aligned}$$

Mit diesem Wert eingesetzt in (5.12) kann u berechnet werden.

$$\begin{aligned} 5 \cdot (2.067344815)^{0.34} - 2.067344815 &= u \\ u &\approx 4.333103529 \end{aligned}$$

Als Ausgangspunkt für das Verfahren ergibt sich das Gleichgewicht

$$\Big[\ \bar{x};\ \ \bar{u}\ \Big] \ = \ \Big[\ (2.067344815, 0);\ \ 4.333103529\ \Big]$$

in der zugrunde gelegten Modell-Volkswirtschaft.

In SCHMITT-GROHÉ und URIBE [23] wird der Steuerungsvariablen u das Konsumverhalten der Marktteilnehmer zugeordnet. Die Variable x_1 ist der Kapitalstock der Volkswirtschaft und mit x_2 wird die technologische Veränderung bezeichnet.
Überträgt man diese Interpretation auf die obigen Werte, so bedeutet dies, daß ein Gleichgewicht nur bei einem Konsum von ca. 4.33 Einheiten und einer Kapitalausstattung von ca. 2.07 Einheiten entsteht. Eine technologische Änderung darf nicht stattfinden.

5.1.2 Implementierung in Matlab

Das Modell-Beispiel wurde in Anlehnung an die Beispiel-Routinen von SCHMITT-GROHÉ UND URIBE in drei MATLAB-Routinen implementiert:
`Beispiel.m`, `Beispiel_Zahlen.m` und `Beispiel_Run.m`, die im Anhang B.1 bis B.3 aufgeführt sind.
Die letzte Routine stellt das auszuführende Programm dar. Die Routinen können je nach dem zu betrachtenden Beispiel angepaßt werden.

Bemerkung 5.4 *Auch wenn nur die Routine* `Beipiel_Run.m` *ausgeführt wird, so müssen trotzdem die anderen beiden Routinen für das Beispiel im gleichen Verzeichnis gespeichert sein. Dasselbe gilt auch für alle anderen aufgerufenen Routinen, da* MATLAB *sonst nicht auf die benötigten Werte und Ableitungen zugreifen kann.*

Beispiel.m

Die Hauptaufgabe dieser Routine ist die Berechnung der analytischen Ableitungen. Die Routine benötigt keine Eingabe.

INPUT: keine Übergabe von Werten aus anderen Funktionen und keine direkten Eingaben

- Definiere die Parameter
  α, β, ρ, σ, η, ε, A als symbolische Variablen
- Definiere die Variablen
  $x_1, x_1^{'}, x_2, x_2^{'}, u, u^{'}$ als symbolische Variablen
- Definiere die Gleichgewichtsfunktion f zeilenweise
- Definiere die Zustands- und Kontrollvektoren
- Rufe die Routine anal_deriv.m auf

OUTPUT: f_x, f_u, $f_{x'}$, $f_{u'}$, f_{xu}, $f_{xu'}$, f_{xx}, $f_{xx'}$, $f_{x'u}$, $f_{x'u'}$,
$f_{x'x}$, $f_{x'x'}$ f_{uu}, $f_{uu'}$, f_{ux}, $f_{ux'}$, $f_{u'u}$, $f_{u'u'}$, $f_{u'x}$, $f_{u'x'}$

Die Variablen haben bei der Implementierung des Beispiels in MATLAB andere Bezeichnungen als bisher:

Name in der Theorie	Name im Programm
x_1	x1
$x_1^{'}$	x1p
x_2	x2
$x_2^{'}$	x2p
u	u
$u^{'}$	up

Tabelle 5.1: Zusammenstellung der MATLAB-Bezeichnungen der Variablen in den Beispiel-Programmen.

Beispiel_Zahlen.m

In dieser Routine müssen die ensprechenden Werte für die Parameter eingegeben werden, sowie die Gleichgewichtswerte für u, x_1 und x_2. Außerdem wird festgelegt, daß sich im Gleichgewicht die Werte der Variablen nicht mehr ändern.

INPUT: `keine Übergabe von Werten aus anderen Funktionen, aber Eingabe aller Parameter- und Variablenwerte`

- `Gebe die Werte für die Parameter ein`
- `Gebe die Gleichgewichtswerte von x und u ein`
- `Setze`
  `x2p = x2`
  `x1p = x1`
  `up = u`

OUTPUT: `Werte für` $A, \alpha, \beta, \rho, \sigma, \eta$ `sowie für` $(\bar{x}, \bar{u})$

Durch `Beispiel_Zahlen.m` können alle anderen Routinen auf die Werte der Variablen und Parameter zurückgreifen.

Bemerkung 5.5 *Die Eingabe der Werte ist fest implementiert und muß erfolgen, bevor das auszuführende Programm gestartet wird. Die Werte können und sollen zur Laufzeit nicht geändert werden.*

Beispiel_Run.m

Diese Routine stellt das auszuführende Programm dar und berechnet die Terme für die quadratische TAYLOR-Approximation.

INPUT: `keine direkte Eingabe, bekommt im Verlauf aber die ausgewerteten Ableitungen von num_eval.m übergeben`

- `Rufe Beispiel.m auf, berechne damit die analytischen Ableitungen`
- `Rufe Beispiel_Zahlen.m, belege so die Variablen und Parameter mit Werten`
- `Bestimme die Ordnung der Approximation`
- `Rufe num_eval.m um die Ableitungen im Gleichgewicht zu berechnen`
- `Rufe gxhx.m und übergebe diese Werte der Ableitungen an die Funktion. Ausgabe der linearen Approximationsterme.`
- `Rufe gxx_hxx.m und gss_hss.m und übergebe die benötigten Ableitungen und Werte an die Funktionen. Ausgabe der quadratischen Approximationsterme.`

OUTPUT: `Die Matrizen gx und hx und die Felder gxx,hxx,gss und hss.`

5.2 Ergebnisse

5.2.1 Werte aus dem Verfahren

Für die praktische Anwendung wird nun das Modell-Beispiel mit dem im vorherigen Abschnitt ermittelten Gleichgewicht

$$\left[\ \bar{x};\ \ \bar{u}\ \right] \ = \ \left[\ (2.067344815, 0);\ \ 4.333103529\ \right]$$

in dem Verfahren getestet und anschließend mit der exakten Lösung verglichen.
Dabei berechnet MATLAB für die Approximation die folgenden Ergebnisse:

- Koeffizienten der linearen Terme:

$$g_x = \begin{pmatrix} 0.7126 & 4.3331 \end{pmatrix}$$

$$h_x = \begin{pmatrix} 0.3400 & 2.0673 \\ 0 & 0.9 \end{pmatrix}.$$

- Koeffizienten der in x quadratischen Terme:

$$g_{xx}(:,:,1) = \begin{pmatrix} -0.2275 & 0.7126 \end{pmatrix}$$

$$g_{xx}(:,:,2) = \begin{pmatrix} 0.7126 & 4.3331 \end{pmatrix}$$

$$h_{xx}(:,:,1) = \begin{pmatrix} -0.1085 & 0.3400 \\ 0 & 0 \end{pmatrix}$$

$$h_{xx}(:,:,2) = \begin{pmatrix} 0.3400 & 2.0673 \\ 0 & 0 \end{pmatrix}.$$

- Koeffizienten der in σ quadratischen Terme:

$$g_{\sigma\sigma} = \begin{pmatrix} 1.5677e-010 \end{pmatrix}$$

$$h_{\sigma\sigma} = \begin{pmatrix} -0.1568e-009 \\ 0 \end{pmatrix}.$$

Man erkennt, daß die letzten beiden Terme ungefähr Null sind. Dies bedeutet, daß der stochastische Anteil bei der Approximation nahezu keine Rolle spielt.

Mit diesen Werten läßt sich nun die Approximation für x und u vollständig beschreiben.
Für die Kontrolle $u = g(x, \sigma)$ ergibt sich gemäß (3.12) die folgende Approximation:

$$\begin{aligned}
\widehat{u} &= 4.333103529 + 0.7126 \cdot [(x_1 - 2.067344815)] + 4.3331 \cdot [(x_2 - 0)] \\
&\quad + \frac{1}{2} \cdot (-0.2275) \cdot [(x_1 - 2.067344815)] \cdot [(x_1 - 2.067344815)] \\
&\quad + \frac{1}{2} \cdot 0.7126 \cdot [(x_2 - 0)] \cdot [(x_1 - 2.067344815)] \\
&\quad + \frac{1}{2} \cdot 0.7126 \cdot [(x_1 - 2.067344815) \cdot [(x_2 - 0)] \\
&\quad + \frac{1}{2} \cdot 4.3331 \cdot [(x_2 - 0)] \cdot [(x_2 - 0)] \\
&\quad + \frac{1}{2} \cdot 0 \cdot [0.008] \cdot [0.008].
\end{aligned}$$

Die Approximation der Dynamik liefert gemäß (3.12)

$$\begin{aligned}
\widehat{x}_1^{'} &= 2.067344815 + 0.3400 \cdot [(x_1 - 2.067344815)] + 2.0673 \cdot [(x_2 - 0)] \\
&\quad + \frac{1}{2} \cdot (-0.1085) \cdot [(x_1 - 2.067344815)] \cdot [(x_1 - 2.067344815)] \\
&\quad + \frac{1}{2} \cdot 0.34 \cdot [(x_2 - 0)] \cdot [(x_1 - 2.067344815)] \\
&\quad + \frac{1}{2} \cdot 0.34 \cdot [(x_1 - 2.067344815) \cdot [(x_2 - 0)] \\
&\quad + \frac{1}{2} \cdot 2.0673 \cdot [(x_2 - 0)] \cdot [(x_2 - 0)] \\
&\quad + \frac{1}{2} \cdot 0 \cdot [0.008] \cdot [0.008],
\end{aligned}$$

$$
\begin{aligned}
\widehat{x}_2^{'} \;=\; & 0 + 0 \cdot [(x_1 - 2.067344815)] + 0.9 \cdot [(x_2 - 0)] \\
& + \frac{1}{2} \cdot 0 \cdot [(x_1 - 2.067344815)] \cdot [(x_1 - 2.067344815)] \\
& + \frac{1}{2} \cdot 0 \cdot [(x_2 - 0)] \cdot [(x_1 - 2.067344815)] \\
& + \frac{1}{2} \cdot 0 \cdot [(x_1 - 2.067344815) \cdot [(x_2 - 0)] \\
& + \frac{1}{2} \cdot 0 \cdot [(x_2 - 0)] \cdot [(x_2 - 0)] \\
& + \frac{1}{2} \cdot 0 \cdot [0.008] \cdot [0.008] \\
& + \eta\sigma\varepsilon^{'}.
\end{aligned}
$$

Bemerkung 5.6

(i) Bei der Approximation von $x_2^{'}$ kann man sehr gut erkennen, daß die eigentliche Approximation ungefähr Null ist und der hintere stochastische Teil die größte Rolle spielt.

(ii) Bei der Approximation von u und $x_1^{'}$ stellt man fest, daß jeweils der erste Term das meiste Gewicht in die Berechnung einbringt, vorausgesetzt x_1 und u liegen genügend nahe beim Gleichgewicht. Daraus läßt sich als Vermutung folgern, daß die Approximation in einer genügend kleinen Umgebung um das Gleichgewicht gute Ergebnisse liefern müsste, falls die Funktion hinreichend glatt ist.

Für die praktische Anwendung ist die Approximation der Kontrolle besonders interessant, da mit u die optimale Wertefunktion $V(x)$ berechnet werden kann, siehe Definition 2.8. Deshalb beziehen sich die folgenden Auswertungen ausschließlich auf die Approximation der Kontrolle. Die Approximation der Dynamik wird vernachlässigt, da sie für die Bestimmung von $V(x)$ nicht benötigt wird.

5.2.2 Auswertung der praktischen Ergebnisse

Vorgehen bei der Auswertung

Um die Ergebnisse sinnvoll diskutieren und vergleichen zu können, wurde die MATLAB-Routine `Auswertung1.m` geschrieben, die im Anhang B.4 aufgeführt ist. Ziel dieser Routine ist es, Matrizen zu erstellen, die die zu vergleichenden Werte beinhalten. Außerdem werden diese Ergebnisse geplottet, um eine graphische Veranschaulichung zu bekommen.

Die Anzahl der Zeilen einer Matrix richtet sich dabei nach der Anzahl der x_1-Werte, die Spaltenanzahl nach der Anzahl der x_2-Werte, so daß sich ein Gitter zur Approximation der Wertefunktion $V(.)$ in der Umgebung des Gleichgewichtes $(\bar{x}, \bar{u})$ ergibt.
Die Anzahl der Knoten ist abhängig von der Größe des Intervalls und der gewählten Schrittweite. Diese beiden Parameter müssen in Auswertung1.m eingegeben werden. Danach werden zwei Vektoren x_1 und x_2 mit den entsprechenden Werten belegt.
Die Routine enthält die oben genannte Formel (5.3) zur exakten Berechnung der optimalen Wertefunktion $V(x)$. Diese Formel wird an den gewünschten x_1- und x_2-Stellen berechnet und die Matrix V_exakt mit den jeweiligen Ergebnissen belegt.
Anschließend wird mit den Koeffizienten aus der Methode von SCHMITT-GROHÉ und URIBE der approximierte Wert für die Kontrolle u berechnet und auf dieser Basis eine approximierte Wertefunktion $\widehat{V}$ berechnet. Hierbei entstehen:

1. Die Matrix T, die an der Stelle (i,j) die Kontrolle $\widehat{u}$ enthält, die sich mit x_{1_i} und x_{2_j} ergibt, $i, j = 1, 2, ...$

2. Eine Matrix V_approx, die die entsprechenden Werte der approximierten Wertefunktion $\widehat{V}$ enthält.

Danach ist es möglich, den Approximationsfehler zu berechnen indem die Differenz zwischen V_exakt(i,j) und V_approx(i,j) bestimmt wird. Es wird dabei nur der absolute Fehler betrachtet. Die Matrix, die hierbei entsteht, heißt Fehlerabs.
Auf die gleiche Weise wird noch eine weitere Matrix erstellt, die den direkten Fehler bei

der Kontrolle enthält. Auch für die Kontrolle existiert eine exakte Formel, siehe (5.4), so daß dies möglich ist.
Am Schluß der Routine werden drei Plots erstellt:

1. Die erste Graphik zeigt den betragsmäßigen Fehler zwischen den beiden Wertefunktionen.

2. Die zweite bildet die beiden Wertefunktionen ab.

3. Das dritte Bild zeigt den betragsmäßigen Fehler bei der Kontrolle $\widehat{u}$.

Das Intervall $\Omega = [2.055, 2.075] \times [-0.01, 0.01]$

Zuerst wird eine Auswertung in einer sehr kleinen Umgebung um das Gleichgewicht betrachtet. Als Intervall wird $\Omega = [2.055, 2.075] \times [-0.01, 0.01]$ gewählt, es weicht vom Gleichgewichtspunkt in jede Richtung um ca. 0.01 ab. Als Schrittweite h wird $h = 0.001$ verwendet, wodurch sich bei der Routine `Auswertung1.m` jeweils Matrizen der Größe 21×21, respektive 441 Knoten ergeben.

In Abbildung 5.2 wird der Fehler zwischen $V(x)$ und $\widehat{V}(x)$ gezeigt. Hier läßt sich erkennen, daß der Fehler nahe dem Gleichgewicht mit < 0.02 sehr gering ist. Zu den Intervallrändern hin steigt er an.
Anhand dieser Graphik sieht man, daß der Approximationsfehler in einer kleinen Umgebung relativ gering ist, was die Vermutung von oben stützt.
Auch Abbildung 5.3 unterstützt diese Aussage. Da der Fehler nicht groß ist, weichen die beiden Wertefunktionen nicht stark voneinander ab. Der Fehler entwickelt sich hier passend zu Abbildung 5.2 in beide Richtungen gleich, wobei $\widehat{V}(x)$ zuerst ober- und dann unterhalb von $V(x)$ liegt.

Weiterhin bestätigt Abbildung 5.4 die theoretischen Ergebnisse. Es wird der Fehler bei der Kontrolle direkt dargestellt. Auf den ersten Blick scheint der Fehler am Intervallrand sehr stark anzusteigen. Hier muß allerdings die Größenordnung beachtet werden. Der Fehler am Rand ist von der Größe $\leq 14 \cdot 10^{-7}$, also verschwindend gering.

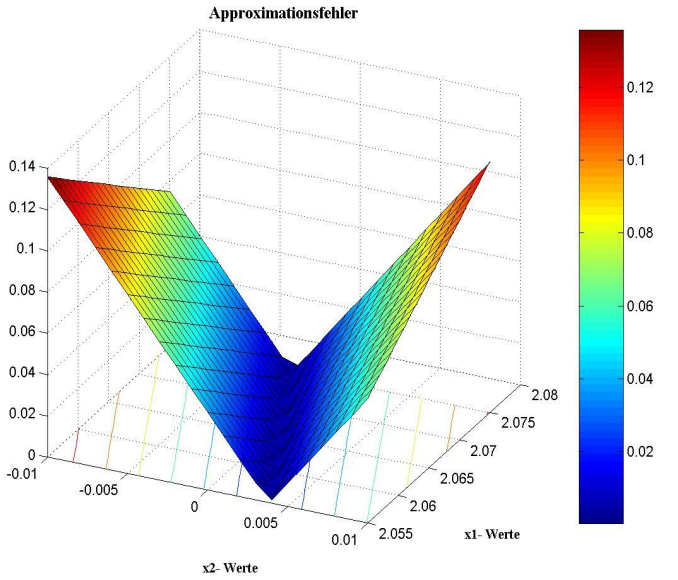

Abbildung 5.2: Die Graphik zeigt den Fehler zwischen der approximierten und der exakten Wertefunktion auf dem Intervall $\Omega = [2.055, 2.075] \times [-0.01, 0.01]$ zur Schrittweite $h = 0.001$.

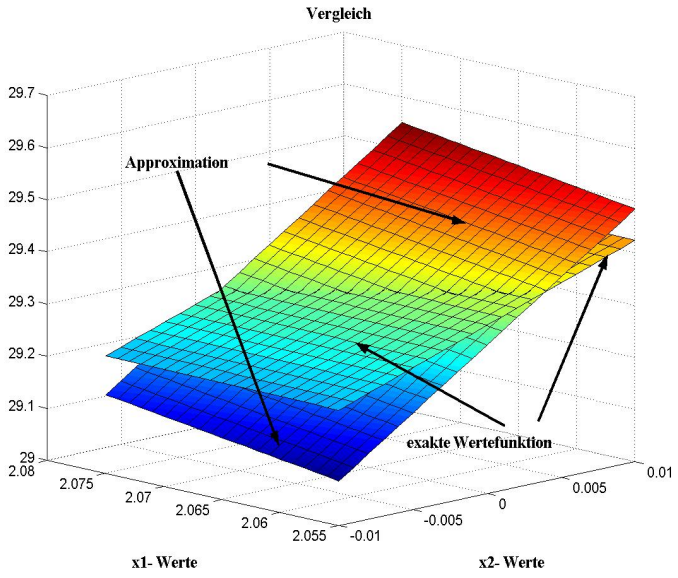

Abbildung 5.3: Das Bild zeigt die approximierte und die exakte Wertefunktion auf dem Intervall $\Omega = [2.055, 2.075] \times [-0.01, 0.01]$ zur Schrittweite $h = 0.001$.

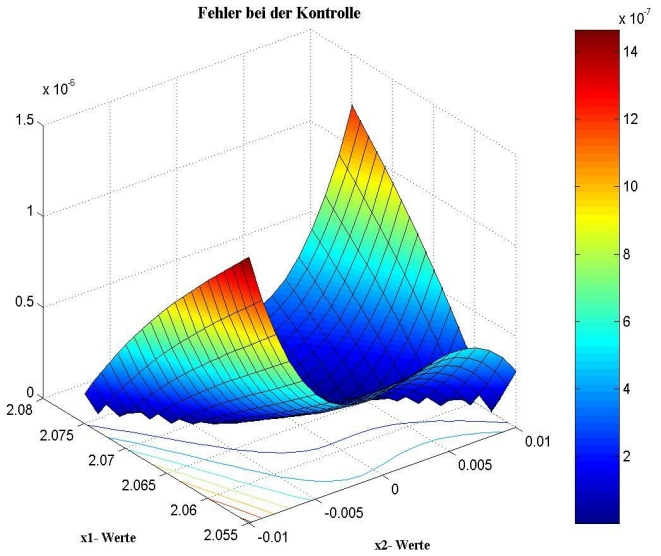

Abbildung 5.4: Der Approximationsfehler bei der Kontrolle u auf dem Intervall $\Omega = [2.055, 2.075] \times [-0.01, 0.01]$ zur Schrittweite $h = 0.001$.

Dies liegt, wie weiter oben vermutet, daran daß bei der Approximation der Kontrolle u der Term $\bar{u}$ das meiste Gewicht hat und alle anderen Terme kaum eine Rolle spielen. Da davon auszugehen ist, daß in der Nähe des Gleichgewichts die Kontrolle nicht stark abweicht, ergeben sich so niedrige Fehler in der Berechnung von $\widehat{u}$.

Es bleibt noch die Frage zu klären, warum der Approximationsfehler bei $\widehat{V}$ ausgeprägter ist als bei $\widehat{u}$. Dies ist zum einen damit zu erklären, daß $\widehat{V}$ nicht direkt approximiert wird. Dazu muß die folgende Formel betrachtet werden:

$$\widehat{V} = \frac{1}{1-\beta} \log(\widehat{u}) \tag{5.13}$$

Dieser Ausdruck wird aus der Zielfunktion (5.1) für $t \to \infty$ mit $\beta < 1$ hergeleitet und gilt im Gleichgewicht.

Da $\widehat{u}$ in diese Formel eingesetzt wird, vergrößert sich der schon vorhandene Fehler.

Die zweite Erklärung ist, daß Formel (5.13) zwar gelten kann $\forall\ (x, u) \in \Omega$, jedoch kann dies nur im Gleichgewicht garantiert werden. Da sie hier zur Approximation der

Umgebung benutzt wird, muß sie also das Ergebnis nicht korrekt wiedergeben, wodurch der größte Anteil des Fehlers bei der Bestimmung von $\widehat{V}(x)$ entsteht. Dies bedeutet, daß sie ein eher schlechtes Werkzeug zur Approximation von V darstellt und gar keine Chance auf einen akzeptablen Fehler läßt. Mangels ähnlich einfacher Alternativen wird die Formel aber trotzdem verwendet und der große Fehler wird toleriert.

Das Intervall $\Omega = [1, 4] \times [-0.3, 0.3]$

Als nächstes wird eine größere Umgebung ausgewertet. Dazu wird das Intervall $\Omega = [1, 4] \times [-0.3, 0.3]$ ausgewählt. Die Schrittgröße ist $h = 0.075$ in x_1-Richtung und $h = 0.015$ in x_2-Richtung. Die Routine Auswertung1.m bildet Matrizen der Größe 41×41, was einer Knotenanzahl von 1681 entspricht.
Bei dieser Auswertung ergibt sich ein anderes Bild als vorher. Abbildung 5.5 zeigt den Approximationsfehler auf dem vergrößerten Intervall: In der Nähe des Gleichgewichts ist er immer noch sehr niedrig, aber am Rand rechts ist er ≥ 7.

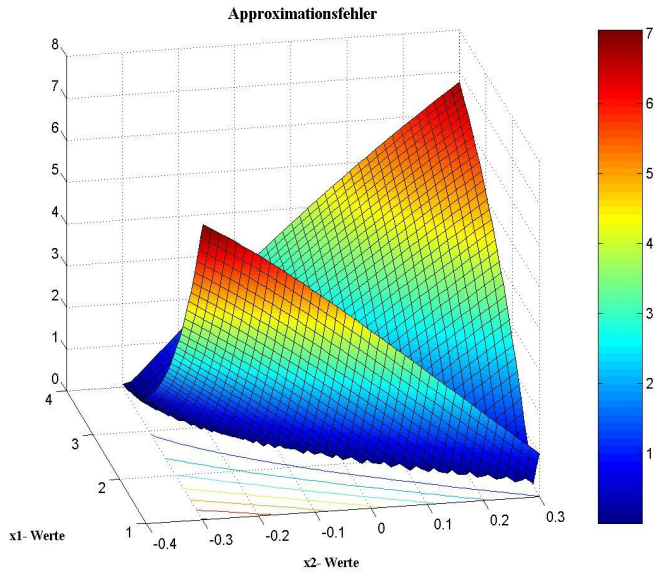

Abbildung 5.5: Die Graphik zeigt den Fehler zwischen der approximierten und der exakten Wertefunktion auf dem Intervall $\Omega = [1, 4] \times [-0.3, 0.3]$ zur Schrittweite $h = 0.075$ in x_1- und $h = 0.015$ in x_2-Richtung.

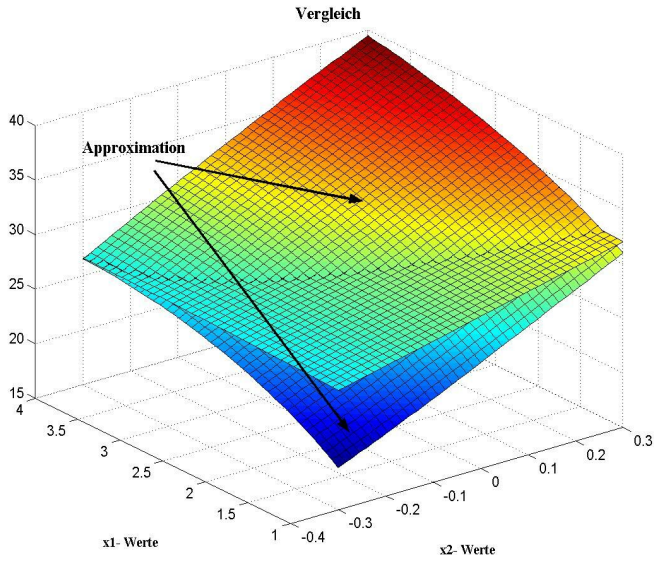

Abbildung 5.6: Das Bild zeigt die approximierte und die exakte Wertefunktion auf dem Intervall $\Omega = [1, 4] \times [-0.3, 0.3]$ zur Schrittweite $h = 0.075$ in x_1- und $h = 0.015$ in x_2-Richtung.

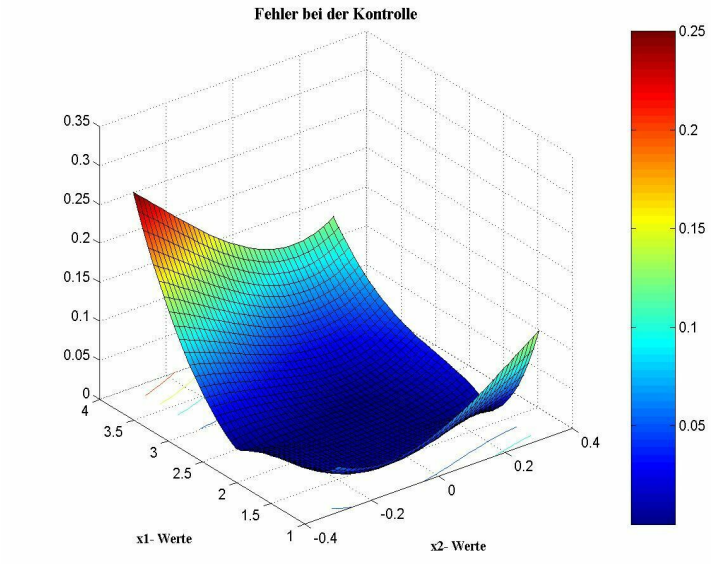

Abbildung 5.7: Der Approximationsfehler bei der Kontrolle u auf dem Intervall $\Omega = [1, 4] \times [-0.3, 0.3]$, zur Schrittweite $h = 0.075$ in x_1- und $h = 0.015$ in x_2-Richtung.

In Abbildung 5.6 ist entsprechend die Entwicklung von V und $\widehat{V}$ abgebildet.
Interessant ist in Abbildung 5.7, daß die Graphik in ihrer Form sehr der Graphik in Abbildung 5.4 ähnelt. Allerdings hat sich die Größenordnung des Fehlers stark verschlechtert und erreicht Werte ≥ 0.25.
Wenn das Intervall weiter vergrößert wird, ergibt sich noch ein interessantes Bild, wie Abbildung 5.8 zeigt. Für die x_2-Werte in negativer Richtung steigt der Fehler zuerst auch an, sinkt dann aber wieder ab, um danach erneut anzusteigen. Der Fehler entwickelt sich also nicht symmetrisch. Der zweite Knick kann nicht damit zusammenhängen, daß das Verfahren evtl. in dieser Richtung wieder genauer wird. Dies würde genauere Werte für $\widehat{u}$ voraussetzten. Wie man aber in Auswertungen für $\widehat{u}$ in diesem Intervall erkennen kann, steigt gerade in dieser Richtung der Fehler an. Des weiteren zeigen Auswertungen noch größerer Intervalle, daß der Fehler in dieser Richtung insgesamt steigt. Deshalb kann der Knick hier als Zufall interpretiert werden.

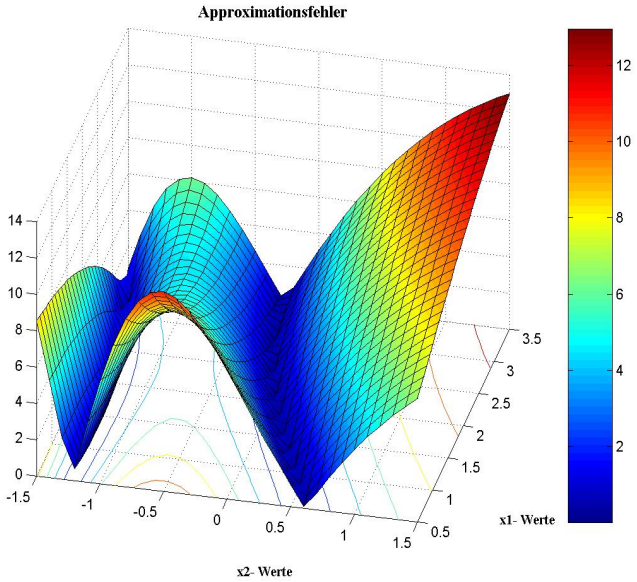

Abbildung 5.8: Die Graphik zeigt den Fehler zwischen der approximierten und der exakten Wertefunktion auf dem Intervall $\Omega = [0.5, 3.5] \times [-1.5, 1.5]$ zur Schrittweite$h = 0.1$.

Insgesamt kann an dieser Stelle festgehalten werden, daß die Fehler sowohl bei $\widehat{V}$ als auch bei $\widehat{u}$ mit größer werdenden Intervallen immer weiter ansteigen. Dies bedeutet, daß

die Entfernung vom Gleichgewicht für die Anwendung des Verfahrens eine essentielle Rolle spielt. In einer sehr kleinen Umgebung um das Gleichgewicht werden sinnvolle Werte mit niedrigen Fehlern geliefert, inbesondere für die Kontrolle $\widehat{u}$. Die Anwendung für $\widehat{V}(x)$ ist allerdings auch schon auf einem relativ kleinen Intervall fraglich. Dabei ist anzumerken, daß SCHMITT-GROHÉ und URIBE ihr Verfahren ausschließlich für die Approximation einer Entscheidungsregel verwenden und die Anwendung bei Wertefunktionen nicht diskutieren.
Mit ansteigender Entfernung vom Gleichgewicht steigen die Fehler so stark an, daß man das Verfahren hier nicht benutzen kann. Ein Vorteil ist die Schnelligkeit der Routine. Da nur ein Rechenschritt ausgeführt wird, erhält man die Werte für die Koeffizienten der TAYLOR-Approximation in Sekundenbruchteilen.

5.3 Ein neoklassisches Modell, Beispiel 2

Da in der Zielfunktion des Modell-Beispiels aus Kapitel 5.1 die Variable x_2 keine Rolle spielt, hat auch der Unsicherheitsfaktor keinen Einfluß auf die Zielfunktion. Dies spiegelt sich in Werten von g_{ss} und $h_{ss} \leq 10^{-9}$ wieder. Deshalb wurde das Modell-Beispiel abgeändert, um der stochastischen Dynamik mehr Gewicht zu verleihen. Des weiteren soll untersucht werden, wie diese Änderung sich auf die Koeffizienten auswirkt.

Das geänderte Beispiel wird mit Beispiel 2 bezeichnet und sieht aus wie folgt:
Als Zielfunktion verwendet man:

$$\max \sum_{t=0}^{\infty} \beta^t \ \log u_t \cdot e^{\kappa \cdot x_2} \tag{5.14}$$

und für die Dynamik:

$$\varphi(x,u,z) \ = \ \begin{pmatrix} A\, e^{x_2}\, x_1^{\alpha} - u \\ \rho x_2 + z \end{pmatrix}. \tag{5.15}$$

Es wurde also lediglich ein Multiplikator in der Zielfunktion ergänzt, der x_2 enthält. Die Nebenbedingungen bleiben gleich. Die Parameter behalten dieselben Werte wie vorher: $A = 5$, $\alpha = 0.34$, $\beta = 0.95$ und $\rho = 0.9$. Es wurde noch ein neuer Parameter κ

eingeführt, der für die Auswertung auf 2 gesetzt wird. Auch dieses Beispiel muß in eine Gleichgewichtsgleichung überführt werden. Dazu werden die Optimalitätsbedingungen aufgestellt und umgeformt. Es ergibt sich die Gleichgewichtsgleichung

$$\widetilde{f} := \begin{pmatrix} x_1^{'} - A\, e^{x_2}\, x_1^{\alpha} + u \\ x_2^{'} - \rho\, x_2 \\ \beta \cdot \frac{1}{u^{'}} \cdot e^{\kappa x_2^{'}} \cdot \alpha \cdot A \cdot e^{x_2^{'}} \cdot x_1^{'(\alpha-1)} - \frac{1}{u} \cdot e^{\kappa x_2^{'}} \end{pmatrix} = 0. \tag{5.16}$$

Durch die Modifikation des Beispiels hat sich die dritte Zeile der Gleichgewichtsgleichung verändert. Allerdings bleibt das Gleichgewicht unverändert und unabhängig von κ, da sich die gleichen Schritte wie oben ergeben:
Man kann einfach erkennen, daß wie oben $\bar{x}_2 = 0$ gelten muß. Dieses Ergebnis wird in die erste Zeile der Gleichgewichtsgleichung (5.16) eingesetzt und man erhält wie in Gleichung (5.12)

$$u = A \cdot 1 \cdot x_1^{\alpha} - x_1. \tag{5.17}$$

Das Ergebnis für die Kontrolle u läßt sich wie oben in Zeile drei von (5.16) verwenden. Daraus ergibt sich

$$\begin{aligned} & \beta\, \alpha\, A\, x_1^{(\alpha-1)}\, \frac{e^{\kappa x_2}}{A\, x_1^{\alpha} - x_1} = \frac{e^{\kappa x_2}}{A\, x_1^{\alpha} - x_1} \\ \Longleftrightarrow \quad & \beta\, \alpha\, A\, x_1^{(\alpha-1)} = 1. \end{aligned}$$

Die restlichen Schritte sind äquivalent zu Beispiel 1 und man erhält wieder das Gleichgewicht

$$\left[\ \bar{x};\ \ \bar{u}\ \right] \ = \ \left[\ (2.067344815, 0);\ \ 4.333103529\ \right].$$

Bei diesem Modell erwartet man, daß die optimale Wertefunktion in ihrem Verlauf leicht gebogen ist. Mit $\kappa = 2$ und nach Anwendung der Perturbations-Methode erhält man das folgende Ergebnis:

- Koeffizienten der linearen Terme sind:

$$g_x = \begin{pmatrix} 0.7126 & 4.7277 \end{pmatrix}$$

$$h_x = \begin{pmatrix} 0.3400 & 1.6727 \\ 0 & 0.9000 \end{pmatrix}$$

- Koeffizienten der in x quadratischen Terme:

$$g_{xx}(:,:,1) = \begin{pmatrix} -0.2275 & 0.7775 \end{pmatrix}$$

$$g_{xx}(:,:,2) = \begin{pmatrix} 0.7775 & 5.0563 \end{pmatrix}$$

$$h_{xx}(:,:,1) = \begin{pmatrix} -0.1085 & 0.2751 \\ 0 & 0 \end{pmatrix}$$

$$h_{xx}(:,:,2) = \begin{pmatrix} 0.2751 & 1.3441 \\ 0 & 0 \end{pmatrix}$$

- Koeffizienten der in σ quadratischen Terme:

$$g_{\sigma\sigma} = \begin{pmatrix} -7.5821 \end{pmatrix}$$

und

$$h_{\sigma\sigma} = \begin{pmatrix} 7.5821 \\ 0 \end{pmatrix}.$$

Interessanterweise sind nun die Terme g_{ss} und h_{ss} um ein Vielfaches im Vergleich zu den Werten aus Abschnitt 5.2.1 gestiegen. Dies zeigt, daß nun der stochastische Anteil eine größere Rolle spielt. Setzt man die Terme zu einer TAYLOR-Approximation zusammen und stellt den Verlauf der Wertefunktion graphisch dar, ergeben sich die Abbildungen 5.9 und 5.10.
Die leichte Krümmung der Wertefunktion läßt sich besonders gut in Abbildung 5.10 erkennen.

Bemerkung 5.7 *Bei diesem Beispiel müssen die Werte für $\widehat{u}$ in die Formel*

$$\widehat{V} = \frac{1}{1-\beta} \cdot \log(\widehat{u}) \cdot e^{\kappa x_2} \tag{5.18}$$

eingesetzt werden. Dieser Ausdruck liefert mit der gleichen Begründung eine ebenso schlechte Approximation von V wie Formel 5.13.

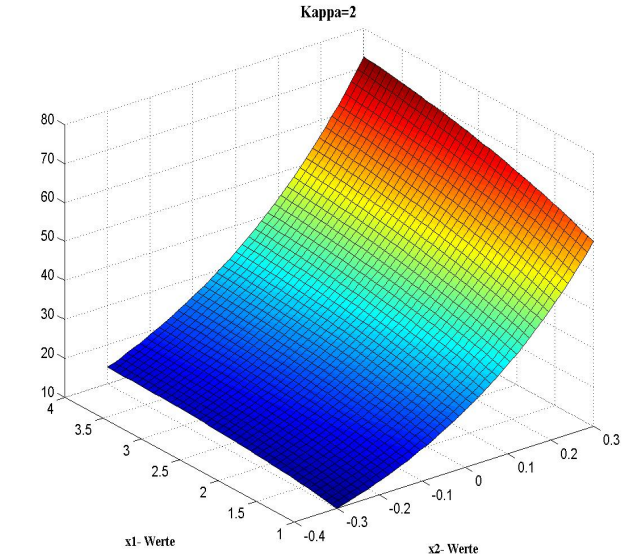

Abbildung 5.9: Die Graphik zeigt die Wertefunktion für $\kappa = 2$ auf dem Intervall $\Omega = [1, 4] \times [-0.3, 0.3]$ zur Schrittweite $h = 0.075$ in x_1-Richtung und $h = 0.015$ in x_2-Richtung.

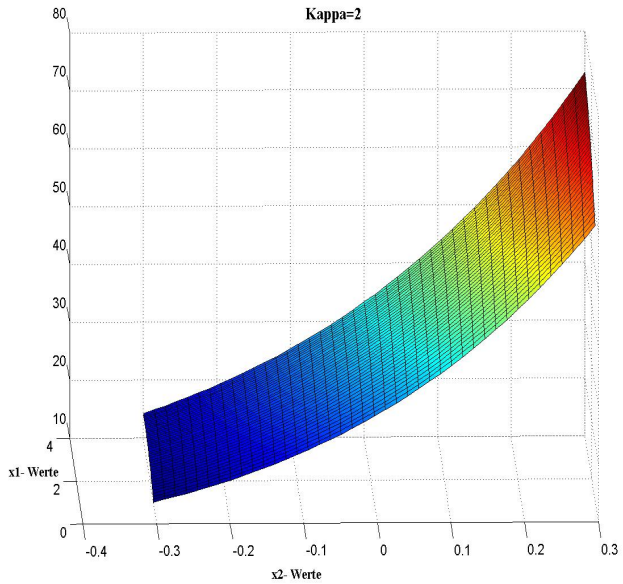

Abbildung 5.10: Die Graphik zeigt die Wertefunktion aus Abbildung 5.9 in der Seitenansicht.

Da für dieses Modell keine exakte Lösung existiert, ist es nicht möglich die Richtigkeit der Ergebnisse zu überprüfen. Von den vorhergehenden Untersuchungen kann man schließen, daß der Verlauf der Wertefunktion annähernd richtig auf diesem Intervall ist, daß die Werte an sich allerdings einem großen Fehler unterliegen.
Mit diesem veränderten Beispiel hat man deutlich gezeigt, wie eine Veränderung der Zielfunktion Auswirkungen auf die Koeffizienten bei der Approximation haben kann.

6 Vergleich mit der dynamischen Programmierung

Um die Ergebnisse aus der Anwendung der Methode von SCHMITT-GROHÉ und URIBE einordnen zu können, ist es nötig, diese einem Vergleich zu unterziehen. Dazu wird ein von GRÜNE entwickelter Algorithmus benutzt, der in [10] hergeleitet und in [12] beschrieben und angewendet wird. Da die Basis dafür bestimmten Eigenschaften der optimalen Wertefunktion $V(x)$ sind, werden diese im ersten Abschnitt eingeführt und ihr Hintergrund erläutert. Außerdem werden die Grundlagen und der Aufbau des Verfahrens kurz erklärt. Im zweiten Teil werden die beiden Beispiele aus dem vorangegangenen Kapitel mit diesem Algorithmus ausgewertet und mit den bisherigen Ergebnissen verglichen.

6.1 Theoretischer Hintergrund von Bellmann-Verfahren

Der betrachtete Algorithmus von GRÜNE basiert auf dem BELLMANNschen Optimalitätsprinzip. Es ist auch bekannt unter dem Begriff Dynamische Programmierung. BELLMANN beschreibt das Prinzip in [2] mit den Worten:

> An optimal policy has the property that whatever the initial state and initial decision are the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

Das Prinzip besagt also, daß Teilstücke optimaler Trajektorien wieder optimale Trajektorien sind. Das BELLMANNsche Optimalitätsprinzip ist eine Eigenschaft der optimalen Wertefunktion. Diese läßt sich mathematisch mit dem nächsten Satz beschreiben.

Satz 6.1 ***(BELLMANNsches Optimalitätsprinzip)***

Betrachte das optimale Steuerungsproblem aus Definition 2.8. Dann erfüllt die optimale Wertefunktion $V_h(x)$ für jedes $x \in \mathbb{R}^d$ und für jedes $k \in \mathbb{N}$ die Gleichung

$$V_h(x) \;=\; \sup_{u_t \in \mathbf{U}} \left\{ \sum_{t=0}^{k} \beta^t \psi(\Phi_h(t, x_t, u_t), u_t) + \; \beta^{k+1} V_h(\Phi_h((k+1), x_t, u_t)) \right\}$$

wobei $h = 1$ gilt.

Beweis:

(i) "$\leq$": Seien $x \in \mathbb{R}^d$, $k \in \mathbb{N}$ und $u_t \in \mathbf{U}$ beliebig. Dann gilt

$$\begin{aligned}
J_h(x_t, u_t) &= \sum_{t=0}^{\infty} \beta^t \psi(\Phi_h(t, x_t, u_t), u_t) \\
&= \sum_{t=0}^{k} \beta^t \psi(\Phi_h(t, x_t, u_t), u_t) + \sum_{t=k+1}^{\infty} \beta^t \psi(\Phi_h(t, x_t, u_t), u_t) \\
&= \sum_{t=0}^{k} \beta^t \psi(\Phi_h(t, x_t, u_t), u_t) \\
&\quad + \sum_{t=0}^{\infty} \beta^{(t+k+1)} \psi(\Phi_h(t, \Phi_h((k+1), x_t, u_t), u_t), u_{t+k+1}) \\
&= \sum_{t=0}^{k} \beta^t \psi(\Phi_h(t, x_t, u_t), u_t) \\
&\quad + \beta^{k+1} \sum_{t=0}^{\infty} \beta^t \psi(\Phi_h(t, \Phi_h((k+1), x_t, u_t), u_t), u_{k+1+.}) \\
&= \sum_{t=0}^{k} \beta^t \psi(\Phi_h(t, x_t, u_t), u_t) + \beta^{k+1} J_h(\Phi_h((k+1), x_t, u_t), u_{k+1+.}) \\
&\leq \sum_{t=0}^{k} \beta^t \psi(\Phi_h(t, x_t, u_t), u_t) + \beta^{k+1} V_h(\Phi_h((k+1), x_t, u_t)).
\end{aligned}$$

Da $u_t \in \mathbf{U}$ beliebig war, gilt die Ungleichung auch für

$$\sup J_h(x_t, u_t) = V_h(x_t, u_t).$$

(ii) "$\geq$": Seien $x \in \mathbb{R}^d$, $k \in \mathbb{N}$ und $\varepsilon > 0$ beliebig. Wähle ein $\breve{u}_t \in \mathbf{U}$ so daß

$$\begin{aligned}
&\sup_{u_t \in \mathbf{U}} \left\{ \sum_{t=0}^{k} \beta^t \psi(\Phi_h(t, x_t, u_t), u_t) + \; \beta^{k+1} V_h(\Phi_h((k+1), x_t, u_t)) \right\} \qquad (6.1) \\
\leq \; &\sum_{t=0}^{k} \beta^t \psi(\Phi_h(t, x_t, \breve{u}_t), \breve{u}_t) + \beta^{k+1} V_h(\Phi_h((k+1), x_t, \breve{u}_t)) + \varepsilon
\end{aligned}$$

Dadurch ist $\check{u}_t$ auf [0,k+1] festgelegt. Wähle nun $\check{u}_t|_{]k+1,\infty[}$ so, daß

$$J_h(\Phi_h((k+1), x_t, \check{u}_t), \check{u}_{k+1+.}) \geq V_h(\Phi_h((k+1), x_t, \check{u}_t)) - \varepsilon. \tag{6.2}$$

(6.2) wird in (6.1) eingesetzt. Damit ergibt sich

$$\begin{aligned} & \sup_{u_t \in \mathbf{U}} \left\{ \sum_{t=0}^{k} \beta^t \psi(\Phi_h(t, x_t, u_t), u_t) + \beta^{k+1} V_h(\Phi_h((k+1), x_t, u_t)) \right\} \qquad (6.3) \\ \leq & \sum_{t=0}^{k} \beta^t \psi(\Phi_h(t, x_t, \check{u}_t), \check{u}_t) + \beta^{k+1}(J_h(\Phi_h((k+1), x_t, \check{u}_t), \check{u}_{k+1+.}) + \varepsilon) + \varepsilon \\ = & \sum_{t=0}^{k} \beta^t \psi(\Phi_h(t, x_t, \check{u}_t), \check{u}_t) + \beta^{k+1}(J_h(\Phi_h((k+1), x_t, \check{u}_t), \check{u}_{k+1+.}) \\ & + (1 + \beta^{k+1})\varepsilon \end{aligned}$$

Nun werden wie in Teil (i) die Grenzen verschoben, aber dieses Mal in die andere Richtung:

$$J_h(\Phi_h((k+1), x_t, \check{u}_t), \check{u}_{k+1+.}) = \frac{1}{\beta^{k+1}} \sum_{t=k+1}^{\infty} \beta^t \psi(\Phi_h(t, x_t, \check{u}_t), \check{u}_t)$$

Dies wird in (6.3) eingesetzt:

$$\begin{aligned} & \sup_{u_t \in \mathbf{U}} \left\{ \sum_{t=0}^{k} \beta^t \psi(\Phi_h(t, x_t, u_t), u_t) + \beta^{k+1} V_h(\Phi_h((k+1), x_t, u_t)) \right\} \\ \leq & \sum_{t=0}^{k} \beta^t \psi(\Phi_h(t, x_t, \check{u}_t), \check{u}_t) + \beta^{k+1}(J_h(\Phi_h((k+1), x_t, \check{u}_t), \check{u}_{k+1+.}) \\ & + (1 + \beta^{k+1})\varepsilon \\ = & \sum_{t=0}^{k} \beta^t \psi(\Phi_h(t, x_t, \check{u}_t), \check{u}_t) + \sum_{t=k+1}^{\infty} \beta^t \psi(\Phi_h(t, x_t, \check{u}_t), \check{u}_t) \\ = & J_h(x_t, \check{u}_t) + (1 + \beta^{k+1})\varepsilon \leq V_h(x_t) + (1 + \beta^{k+1})\varepsilon. \end{aligned}$$

Da ε beliebig war, folgt so die Behauptung.

□

Eine weitere wichtige Eigenschaft der optimalen Wertefunktion ist, daß $V(x)$ durch das Optimalitätsprinzip eindeutig bestimmt ist.

Satz 6.2 *Betrachte das optimale Steuerungsproblem aus Definition 2.8 mit optimaler Wertefunktion $V_h(x)$. Sei ein $k \in \mathbb{N}$ gegeben und sei $W_h : \mathbb{R}^d \to \mathbb{R}$ eine beschränkte Funktion, die das Optimalitätsprinzip*

$$W_h(x) = \sup_{u_t \in \mathbf{U}} \left\{ \sum_{t=0}^{k} \beta^t \psi(\Phi_h(t, x_t, u_t), u_t) + \beta^{k+1} W_h(\Phi_h((k+1), x_t, u_t)) \right\}$$

für alle $x \in \mathbb{R}^d$ erfüllt. Dann ist $V_h = W_h$.

Für den Beweis auf GRÜNE [10] verwiesen.

Die Hauptidee von BELLMANN ist es, daß ein komplexes System in einfachere Subprobleme aufgeteilt werden kann, wobei die Optimierung des Systems durch die Optimierung der Subprobleme stattfindet. Die Lösung der einfacheren Probleme führt also auf die Lösung des Original-Problems.
Damit das BELLMANNsche Optimalitätsprinzip angewendet werden kann, ist es unabdingbar, daß das Optimierungsproblem in einzelne Stufen getrennt werden kann. Dies ist z.B. bei diskreten Problemen durch die einzelnen Zeitstufen gegeben. Die Trennung kann aber auch künstlich erfolgen. Für Beispiele von typischen Problemen siehe MORLOCK und NEUMANN [18], Kapitel 5.
Es wird also immer ein mehrstufiger Entscheidungsprozess betrachtet, wobei in jeder Stufe die Kontrollvariable u_t in Abhängigkeit vom aktuellen Zustand x_t und der Anzahl der noch verbleibenden Stufen gewählt werden muß. Dies bedeutet für Verfahren, die auf dem Prinzip der dynamischen Programmierung basieren, daß sie iterativ vorgehen, d.h. jede Stufe wird einzeln nacheinander abgearbeitet. Daraus folgt, daß diese Art von Verfahren immer mehrere Schritte machen, um zu einer Lösung zu kommen. Sie benötigen eventuell dementsprechend mehr Zeit für das Optimierungsproblem als andere Methoden. Dies hängt natürlich grundsätzlich vom Problem selbst ab und davon, wieviele Stufen es gibt.

Das hier betrachtete numerische Verfahren besteht aus zwei Schritten. Der erste Schritt ist die Diskretisierung in der Zeit, der bereits in Abschnitt 2.2.3 erläutert wurde. Es liegt

also eine diskrete Wertefunktion V_h vor. Da V_h das BELLMANNsche Optimalitätsprinzip erfüllt, ist es möglich, dieses als Basis für ein Iterationsverfahren zu benutzen.

Definition 6.3 *Es werden iterativ Funktionen* $v_h^i : \mathbb{R}^d \to \mathbb{R}^d$, $i = 0, 1, ...$ *definiert mittels* $v_h^0 = 0$ *und* $v_h^{i+1} = \mathbf{T}_h(v_h^i)(x)$ *für alle* $x \in \mathbb{R}^d$, *wobei der Operator* $\mathbf{T}_h : \mathcal{C}(\mathbb{R}^d, \mathbb{R}) \to \mathcal{C}(\mathbb{R}^d, \mathbb{R})$ *gegeben ist durch*

$$\mathbf{T}_h(w)(x) := \max_{u \in \mathbf{U}} \{h\psi(x, u) + \beta \cdot w(\varphi_h(x, u))\}$$

Nachdem die Basis für die Iteration definiert wurde, folgt der Schritt der räumlichen Diskretisierung. Hierfür ist die Überlegung notwendig, daß die Funktionen v_h^i in dieser Form nicht implementiert werden können, da man sie dafür an unendlich vielen Stellen auswerten müßte. Daher verwendet man Funktionen, die nur an endlich vielen Punkten zu berechnen sind.
Es wird angenommen, daß die kompakte Menge $\Omega \subset \mathbb{R}^2$ ein Rechteck ist. Die Methode läßt sich auch für den Fall $d > 2$ erweitern. Zuerst muß ein passender Funktionenraum definiert werden.

Definition 6.4 *(i) Sei* $A \subset \mathbb{R}^2$. *Eine Funktion* $w : A \to \mathbb{R}$ *heißt affin bilinear, falls es Konstanten* $\alpha_0, ..., \alpha_3, \alpha_i \in \mathbb{R}, i = 1, ...3$ *gibt, so daß für alle* $x \in A$ *die Identität* $w(x) = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_1 x_2$ *gilt.*

(ii) Betrachte eine rechteckförmige Menge $\Omega \subset \mathbb{R}^2$. *Die Menge der stetigen und stückweise affin bilinearen Funktionen auf* Ω *bezüglich* Γ *ist definiert als*
$\mathcal{W} := \{w : \Omega \to \mathbb{R} |$ *w ist stetig und* $w\,|_{R_i}$ *ist affin bilinear für jedes i=0,...,P-1* $\}$.

Bemerkung 6.5 *Jede Funktion* $w \in \mathcal{W}$ *läßt sich mit ihren Werten* $w(E_i)$ *in den Eckpunkten* E_i *des Gitters identifizieren. Für alle* x, *die nicht auf einem Eckpunkt liegen, gibt es Vorfaktoren, um den Wert* $w(x)$ *zu interpolieren.*

Bemerkung 6.6 *Der Operator* $\mathbf{T}$ *muß dann nur noch an den Eckpunkten des Gitters berechnet werden. Es ergibt sich eine Folge von Funktionen* $\widehat{v}_h^j \in \mathcal{W}$ *durch*

$$\widehat{v}_h^{j+1}(E_i) = \max_{u \in \mathbf{U}} \left\{ h\psi(E_i, u) + \beta \cdot \widehat{v}_h^j(\varphi_h(E_i, u)) \right\}.$$

Bemerkung 6.7 *Der Term $\widehat{v}_h^j(\varphi_h(E_i,u))$ aus der obigen Funktionenfolge kann nicht direkt berechnet werden, da $\widehat{v}_h^j$ nicht an der Stelle $\varphi_h(E_i)$ bekannt ist. Daher muß an dieser Stelle interpoliert werden.*

Schließlich wird $V^j = \left(V_1^j, ..., V_N^j\right) \in \mathbb{R}^N$ gesetzt, mit $V_i^j = \widehat{v}_h^j(E_i)$. Zu einem gegebenen Gitter können nun sukzessive Vektoren $V^j \in \mathbb{R}^N$ nach der folgenden Vorschrift berechnet werden.

Definition 6.8 *Betrachte ein zeitdiskretes optimales Steuerungsproblem und ein Rechteckgitter $\Gamma \in \mathbb{R}^2$ mit P Rechtecken und N Eckpunkten. Zu jedem $u \in \mathbf{U} \subset \mathbb{R}^m$ und jedem $i = 0, ...N-1$ sei B(i,u) der N-dimensionale Zeilenvektor, für den für jedes $w \in \mathcal{W}$ und $W = (w(E_0), ..., w(E_{N-1})) \in \mathbb{R}^N$*

$$w(\varphi_h(E_i,u)) = B(i,u) \cdot W$$

gilt.
Des weiteren sei $G(i,u) = h\psi(E_i,u)$. Dann werden die Vektoren V^j iterativ berechnet mit $V^0 := (0, ...0)$ und dem Gesamtschrittverfahren

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

oder mit dem Gesamtschrittverfahren

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

Es kann gezeigt werden, daß die Vekoren V^j gegen einen eindeutigen Vektor V konvergieren. Für den Beweis, für die Betrachtung der verschiedenen Diskretisierungsfehler und die Angabe von Abbruchkriterien sei auf GRÜNE [10] verwiesen.

Bemerkung 6.9 *Das in Definiton 6.8 beschriebene Iterationsverfahren konvergiert bei Problemen mit kleinen $\delta > 0$ nicht sehr schnell. Es sei angemerkt, daß es auch schnellere Methoden für die Iteration gibt wie z.B. das kontrollierte* GAUSS-SEIDEL*-Verfahren und die Strategie-Iteration, siehe* GRÜNE *[10].*

Ein hervorzuhebender Punkt bei der Vorgehensweise des Algorithmus ist es, daß er mit adaptiven Gittern arbeitet. Dies bedeutet, daß bei Überschreiten einer bestimmten Fehlerschranke das gewählte Gitter an diesen Stellen verfeinert wird. So ist garantiert, daß die Ergebnisse an jedem Gitterpunkt einer bestimmten Genauigkeit entsprechen.

6.2 Praktische Auswertungen

6.2.1 Beispiel 1

Für den Vergleich wird das neoklassische Beispiel 1 aus Kapitel 5 betrachtet und mit dem oben erläuterten Algorithmus berechnet. Die vom Algorithmus erzeugten Werte der optimalen Wertefunktion $\widehat{V}$ und der Kontrolle $\widehat{u}$ werden von den MATLAB-Routinen `Auswertung2a.m` und `-2b.m` eingelesen und ausgewertet.
Zuerst wird wieder der Approximations-Fehler auf einem kleinen Intervall betrachtet. Man wählt wie in Kapitel 5 das Intervall $\Omega = [2.055, 2.075] \times [-0.01, 0.01]$.
Für $\widehat{V}$ liefert Abbildung 6.1 eine ähnliche Graphik wie Abbildung 5.2, allerdings ist nun der Fehler von der Größenordnung 10^{-4}. Das bedeutet, daß das BELLMANN-Verfahren in diesem kleinen Bereich genauer ist.

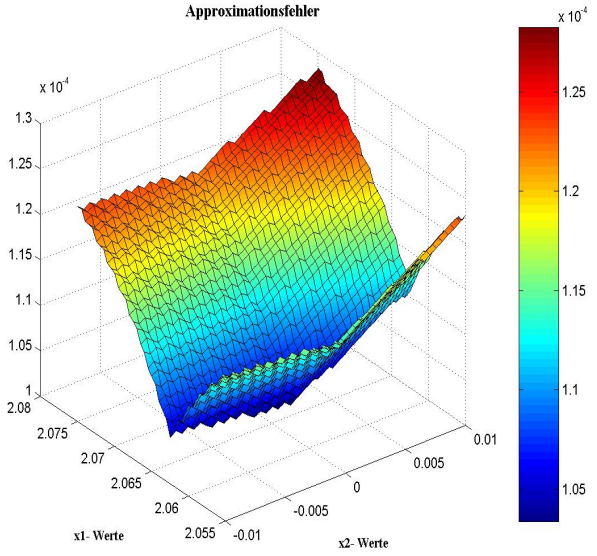

Abbildung 6.1: Der Approximationsfehler bei V auf dem Intervall $\Omega = [2.055, 2.075] \times [-0.01, 0.01]$ zur Schrittweite $h = 0.0005$.

Den umgekehrten Fall liefert der Vergleich von $\widehat{u}$. In Abbildung 6.2 liegt der Fehler bei einer Größenordnung von 10^{-3}. Die entsprechende Abbildung 5.4 aus der Perturbations-Methode zeigt einen Fehler der Größenordnung 10^{-7} bei $\widehat{u}$.

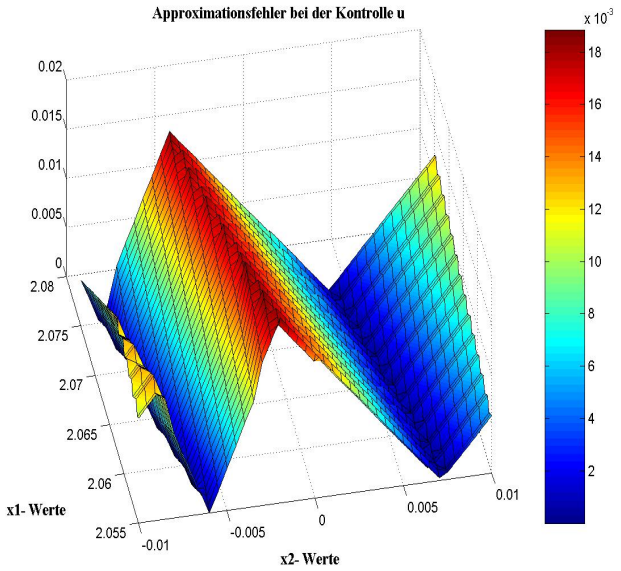

Abbildung 6.2: Der Approximationsfehler bei u auf dem Intervall $\Omega = [2.055, 2.075] \times [-0.01, 0.01]$ zur Schrittweite $h = 0.0005$.

Folglich liefert hier die Perturbations-Methode genauere Werte für die Kontrolle u.

Als vergrößertes Intervall wird ebenfalls wie in Kapitel 5 der Bereich $\Omega = [1, 4] \times [-0.3, 0.3]$ untersucht. Bei der Perturbations-Methode in Abbildung 5.5 zeigte sich auch nach der Vergößerung ein ähnliches Bild wie auf dem kleinen Intervall: Der Fehler steigt vom Gleichgewicht aus zu beiden Seiten hin an.
Im Gegensatz dazu ergibt sich nun ein ganz anderes Bild: In Abbildung 6.3 ist ein uneinheitlicher Verlauf des Approximationsfehlers von $\widehat{V}$ dargestellt. Der Fehler ist überall von der Größenordung 10^{-4}. Daran läßt sich sehr gut erkennen, daß das BELLMANN-Verfahren anders vorgeht als die Perturbations-Methode.
Abbildung 6.4 zeigt ebenfalls einen uneinheitlichen Verlauf des Fehlers bei $\widehat{u}$ in diesem Bereich. Bei der Perturbations-Methode stieg der Fehler zum Rand hin stark an bis auf Werte > 0.2 wie in Abbildung 5.7 zu erkennen ist. Hier ist er überall von der Größenordnung 10^{-3} oder kleiner.

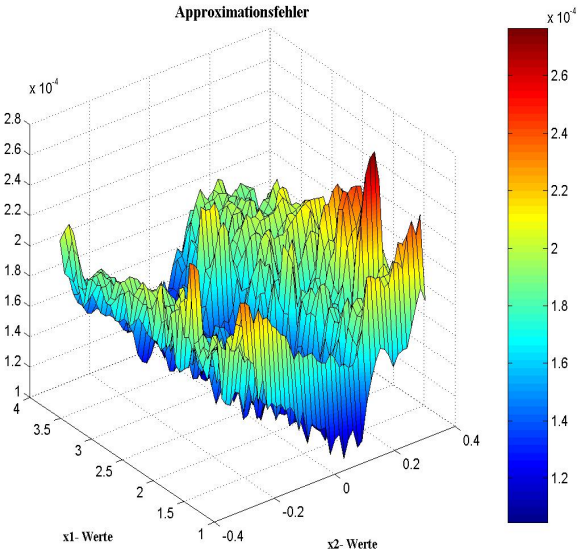

Abbildung 6.3: Der Approximationsfehler bei V auf dem Intervall $\Omega = [1,4] \times [-0.3, 0.3]$ zur Schrittweite $h = 0.075$ in x_1-Richtung und $h = 0.015$ in x_2-Richtung.

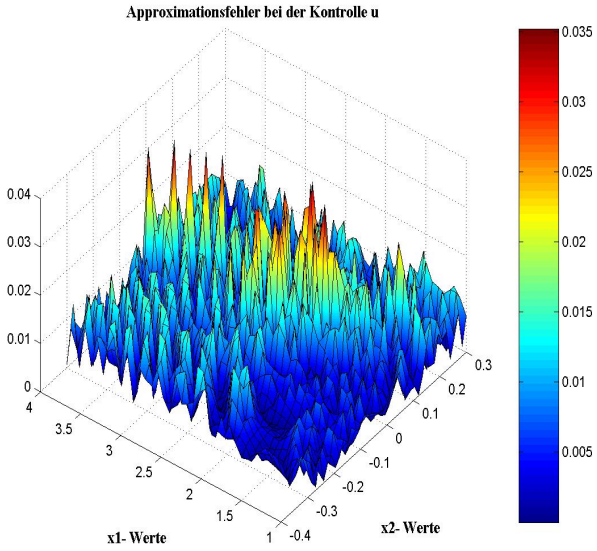

Abbildung 6.4: Der Approximationsfehler bei der Kontrolle u auf dem Intervall $\Omega = [1,4] \times [-0.3, 0.3]$ zur Schrittweite $h = 0.075$ in x_1-Richtung und $h = 0.015$ in x_2-Richtung.

Folglich ist auf dem größeren Intervall die BELLMANN-Methode für die Berechnung in beiden Fällen genauer. Durch die Gitter-Adaption ist es möglich, überall auf dem betrachteten Bereich eine gute Genauigkeit zu erzielen. Bei der zuerst untersuchten Methode hat man eine relativ gute Genauigkeit nur in der Nähe des Gleichgewichts und auch nur für die Kontrolle u.

Um den Vergleich noch besser zu veranschaulichen, werden Intervalle von verschiedener Größe auf den maximalen Approximationsfehler bei $\widehat{V}$ und $\widehat{u}$ mit beiden Methoden untersucht und verglichen. Die Intervalle werden ausgehend vom Gleichgewicht über sieben Stufen sukzessive vergrößert, wie in Tabelle 6.1 gezeigt.

Intervall	**Bereich**
1.	Im Gleichgewicht
2.	$[2.06, 2.07] \times [-0.001, 0.001]$
3.	$[2.055, 2.075] \times [-0.01, 0.01]$
4.	$[1.8, 2.5] \times [-0.05, 0.05]$
5.	$[1.8, 2.5] \times [-0.1, 0.1]$
6.	$[1.5, 3, 5] \times [-0.1, 0.1]$
7.	$[1.5, 3.5] \times [-0.3, 0.3]$
8.	$[1, 4] \times [-0.3, 0.3]$

Tabelle 6.1: Die mit beiden Methoden untersuchten Intervalle.

Die Ergebnisse sind in Tabelle 6.2 zusammengefaßt. Das Verfahren von SCHMITT-GROHÉ und URIBE [23] wird in der Gegenüberstellung mit Verfahren 1 bezeichnet, der Algorithmus von GRÜNE mit Verfahren 2.
Die Tabelle zeigt die Entwicklung des maximalen Approximationsfehlers bei der Berechnung von $\widehat{V}(x)$ und $\widehat{u}$. Wenn man nun zuerst die Fehlerentwicklung bei $\widehat{V}$ vergleicht, wird deutlich, was die Abbildungen oben schon vermuten ließen:
Bei Verfahren 1 steigt der Fehler mit steigender Intervallgröße rapide an. Im Gleichgewicht ist der Fehler nahe bei Null, dies liegt an der Konstruktion des Verfahrens: Der

Intervall	**Verfahren 1**		**Verfahren 2**	
	Fehler bei $\widehat{V}$	Fehler bei $\widehat{u}$	Fehler bei $\widehat{V}$	Fehler bei $\widehat{u}$
1.	$< 1.2 \cdot 10^{-9}$	$< 2.9 \cdot 10^{-10}$	$4.2797 \cdot 10^{-4}$	$1 \cdot 10^{-2}$
2.	0.0322	$2.4927 \cdot 10^{-7}$	$4.4651 \cdot 10^{-4}$	$7.8 \cdot 10^{-3}$
3.	0.1358	$1.4651 \cdot 10^{-6}$	$4.9812 \cdot 10^{-4}$	$12.7 \cdot 10^{-3}$
4.	1.6805	0.0034	$8.3723 \cdot 10^{-4}$	$12.5 \cdot 10^{-3}$
5.	2.1682	0.0048	$8.6718 \cdot 10^{-4}$	$23.3 \cdot 10^{-3}$
6.	4.1218	0.085	$8.4346 \cdot 10^{-4}$	$23.3 \cdot 10^{-3}$
7.	6.0472	0.13	$8.4346 \cdot 10^{-4}$	$24.2 \cdot 10^{-3}$
8.	7.0566	0.2503	$8.4346 \cdot 10^{-4}$	$24.0 \cdot 10^{-3}$

Tabelle 6.2: Vergleich des Approximationsfehlers in beiden Verfahren über sukzessive vergrößerte Intervalle.

Gleichgewichtspunkt $(\bar{x}, \bar{u})$ ist ja schon vorher bekannt und dient als Ausgangspunkt, also muß der Fehler hier Null sein. In den ersten beiden Intervallen nach dem Gleichgewicht kann der Fehler noch als annehmbar gelten, aber alle Werte danach sind nicht mehr akzeptabel. Im letzten untersuchten Bereich beläuft sich dieser Wert auf > 7.
Bei Verfahren 2 hingegen ist der Fehler in drei Bereichen sogar gleich. Ansonsten schwankt er und verdoppelt sich sogar an einer Stelle, aber diese Veränderungen bewegen sich alle in einem sehr kleinen Bereich. Der Fehler liegt immer in der Größenordnung 10^{-4}, außer im letzten Intervall, wo er geringfügig höher liegt. Dieses Ergebnis belegt also die Folgerungen aus Abbildung 6.3 mit den ensprechenden Werten.
Nun muß noch die Fehlerentwicklung von $\widehat{u}$ diskutiert werden. Zuerst wird Verfahren 1 betrachtet. Wie schon in der Auswertung in Abschnitt 5.2.2 zu erkennen war, ist die

Differenz zwischen u und $\widehat{u}$ sehr gering im Vergleich zu V und $\widehat{V}$. Der Fehler bei $\widehat{u}$ liegt im Gleichgewicht bei Null aufgrund der Konstruktion. In den sehr kleinen Umgebungen liegt er bei der Größenordnung 10^{-7} bzw. 10^{-6}. Der Wert steigt zwar mit steigender Intervallgröße an bis auf ca. 0.25, aber er bewegt sich sehr langsam im Vergleich zu den Werten bei $\widehat{V}$.

Bei Verfahren 2 liegt der Fehler ungefähr immer auf dem selben Niveau von 10^{-3}, im Gleichgewicht ist er geringfügig höher. Interessant ist hier, daß der Fehler bei $\widehat{u}$ immer etwas schlechter ausfällt als bei $\widehat{V}$. Dies liegt an der Vorgehensweise der Methode: Es wird zuerst $\widehat{V}$ berechnet. Rückwirkend als Feedback wird dann die Kontrolle auf Basis von $\widehat{V}$ ermittelt, so daß sich der Fehler fortpflanzt.

Tabelle 6.2 verdeutlicht noch einmal, daß die Approximation von $\widehat{u}$ mit Verfahren 1 in kleinen Umgebungen um das Gleichgewicht sehr gut funktioniert, während es bei $\widehat{V}$ problematischer ist. Wie in Abschnitt 5.2.2 erwähnt wurde, entsteht der große Fehler bei $\widehat{V}$ dadurch, daß $\widehat{u}$ in Formel (5.13) eingesetzt wird. Diese gilt aber nur für $t \to \infty$, also im Gleichgewicht, und liefert deshalb außerhalb nur eine schlechte Näherung. Die Werte aus Tabelle 6.2 stützen diese Aussage.

Verfahren 2 ist für beide Approximationen problemlos anwendbar. Bei der Näherung von $\widehat{V}$ ist es außerhalb des Gleichgewichts immer der Perturbations-Methode überlegen. Bei der Berechnung von $\widehat{u}$ liefert es ab dem 3. Intervall $\Omega = [2.055, 2.075] \times [-0.01, 0.01]$ kleinere Fehler.

Bemerkung 6.10 *Beim Vergleich der Werte aus Tabelle 6.2 muß beachtet werden, daß hier leichte Ungenauigkeiten vorliegen können. Da sich in den verschiedenen Intervallen die Schrittweite ändert, werden bestimmte Punkte manchmal durch das Gitter nicht mit abgedeckt. Dies bedeutet, daß der maximale Fehler eventuell an einer anderen Stelle liegen kann, wenn die Schrittweite anders gewählt wird.*

Bemerkung 6.11 *Die Abbildungen 6.2 und 6.4 wurden durch ein Gitter mit jeweils 40 Knoten pro Koordinatenrichtung erstellt. Die maximalen Fehlerwerte für $\widehat{u}$ mit Verfahren 2 wurden durch Datenauswertung von jeweils 100 Knoten gewonnen, um die Genauigkeit zu erhöhen.*

6.2.2 Beispiel 2

In Kapitel 5.3 wurde eine veränderte Version des Modell-Beispiels betrachtet. Es wurde gezeigt, wie sich diese Veränderung auf die berechneten Koeffizienten der Perturbations-Methode auswirkt und wie sich der Verlauf der Wertefunktion verändert. Allerdings gab es für Beispiel 2 keine exakte Lösung, so daß ein Vergleich nicht möglich war. Aus dem in diesem Kapitel bisher betrachteten Vergleich der Perturbations-Methode mit dem BELLMANN-Verfahren läßt sich die Vermutung aufstellen, daß das BELLMANN-Verfahren auch bei Beispiel 2 sowohl bei $\widehat{V}$ als auch bei $\widehat{u}$ eine größere konstante Genauigkeit liefert. Deshalb soll in diesem Abschnitt das Ergebnis der Perturbations-Methode bei dem veränderten Beispiel mit den Ergebnissen aus der dynamischen Programmierung verglichen werden.

Es liegen die Zielfunktion (5.14) und die Dynamik (5.15) zu Grunde. Mit dem Verfahren der dynamischen Programmierung wurde die Wertefunktion für $\kappa = 1, ...4$ berechnet. Der jeweilige Verlauf ist in Abbildung 6.5 dargestellt.

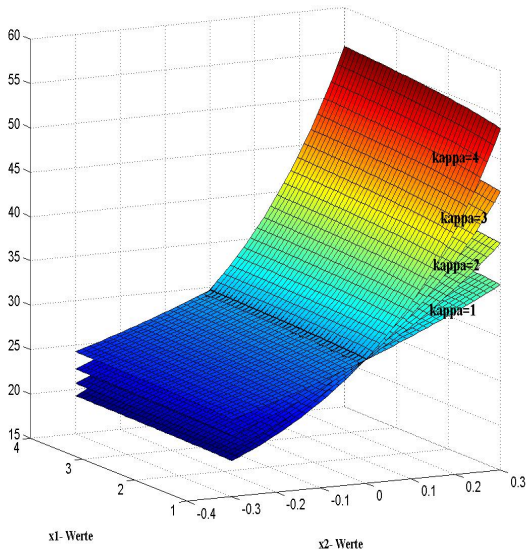

Abbildung 6.5: Verlauf der Wertefunktion des veränderten Beispiels mit $\kappa = 1, ..., 4$.

Bemerkung 6.12 *Alle Abbildungen dieses Abschnittes beziehen sich auf das Intervall* $\Omega = [1, 4] \times [-0.3, 0.3]$ *zur Schrittweite* $h = 0.075$ *in* x_1*-Richtung und* $h = 0.015$ *in* x_2*-Richtung.*

Es ist erkennbar, daß die Krümmung mit steigendem Wert von κ immer stärker ausgeprägt ist. Deshalb kann zunächst festgehalten werden, daß das Verfahren von SCHMITT-GROHÉ und URIBE den Verlauf der Wertefunktion tatsächlich richtig wiedergegeben hat.

Abbildung 6.6 zeigt den Verlauf beider Wertefunktionen. Hierbei ist zu erkennen, daß die mit der Perturbations-Methode berechnete Wertefunktion viel steiler verläuft als die mit dem BELLMANN-Verfahren berechnete Wertefunktion. Anscheinend ist also der Fehler nicht unerheblich.

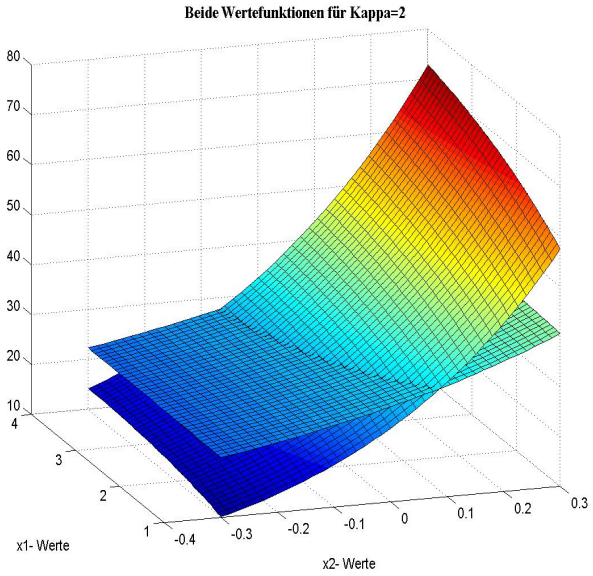

Abbildung 6.6: Das Bild zeigt jeweils den Verlauf der Wertefunktion für die neue Version des Beispiels mit $\kappa = 2$, die mit beiden Methoden berechnet wurde.

Nun wird die Differenz zwischen den Wertefunktionen aus beiden Verfahren an den Knoten berechnet und der absolute Wert betrachtet. Das Ergebnis ist in Abbildung 6.7 dargestellt. Es ist zu erkennen, daß sich der Unterschied am Rand auf Werte > 30 steigert. Dies bedeutet, daß das Verfahren von SCHMITT-GROHÉ und URIBE hier sehr ungenaue Werte

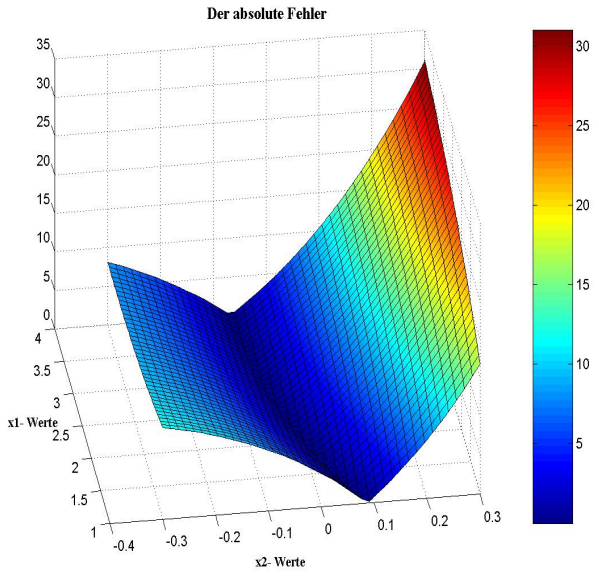

Abbildung 6.7: Hier wird die Differenz zwischen beiden Wertefunktionen betrachtet, wobei der absolute Wert gezeigt wird.

liefert.Deshalb ist es an dieser Stelle interessant herauszufinden, wo diese Methode relativ genaue Werte liefert. Dazu werden alle Punkte herausgefiltert, die einer bestimmten Genauigkeitsgrenze entsprechen. Dies bedeutet, daß alle Stellen gekennzeichnet werden, an denen die Differenz der beiden Wertefunktionen unter einer bestimmten Grenze liegt.

Zuerst werden die Knoten gesucht, an denen der Abstand $\leq 10^{-4}$ ist. Leider findet man zu dieser Schranke keine Werte. Erhöht man die Zahl auf 10^{-3} gibt es auch noch nichts Nennenswertes zu erkennen. Erst bei einer Schranke von 10^{-2} ergibt sich ein Bild, das in Abbildung 6.8 zu sehen ist. Wird die Genauigkeit weiter herabgesetzt auf 10^{-1}, so erhöht sich die Anzahl der betroffenen Punkte wie Abbildung 6.9 zeigt. Es ist offensichtlich, daß die Genauigkeit in der Nähe des Gleichgewichtes am höchsten ist. Allerdings ist dieser Bereich relativ klein und die Ungenauigkeiten außerhalb sind zu groß wie auf Abbildung 6.7 gezeigt wurde. Deshalb kann an dieser Stelle gefolgert werden, daß die Methode von SCHMITT-GROHÉ und URIBE auch die Wertefunktion von Beispiel 2 nur

ungenügend approximiert.

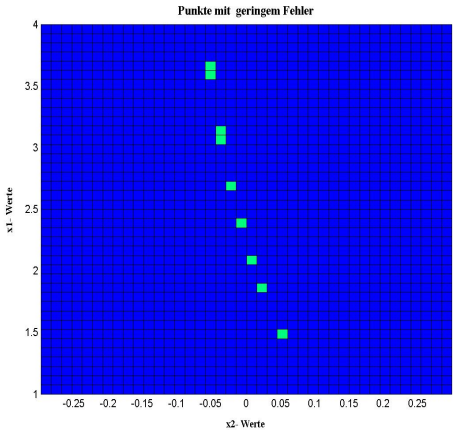

Abbildung 6.8: Punkte mit Differenz$\leq 10^{-2}$.

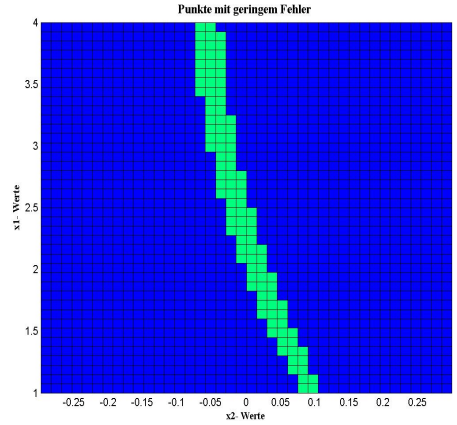

Abbildung 6.9: Punkte mit Differenz $\leq 10^{-1}$.

Für den Vergleich von $\widehat{u}$ aus beiden Methoden wird der Abstand der jeweils berechneten Werte voneinander an den Knoten bestimmt. Das Ergebnis wird in Abbildung 6.10 gezeigt. Zu den Intervallrändern steigt die Differenz auf Werte > 3. Wenn man auch hier

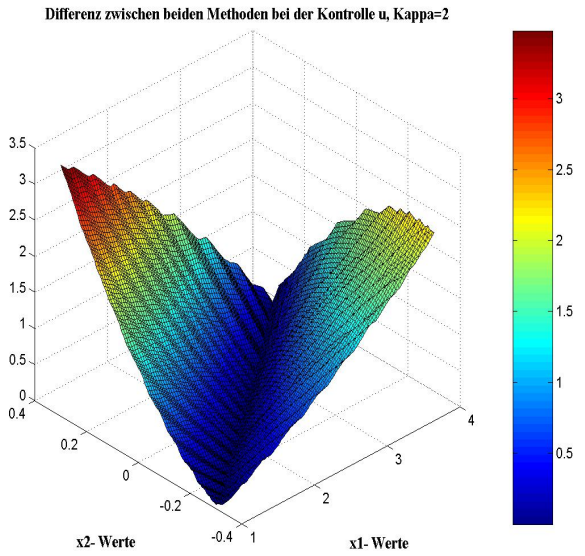

Abbildung 6.10: Hier wird die Differenz zwischen den $\widehat{u}$ dargestellt, wobei der absolute Wert gezeigt wird.

Tabelle 6.2 zu Grunde legt, dann kann man schließen, daß das BELLMANN-Verfahren auf einem Bereich dieser Größe genauer ist. Der große Unterschied wird also durch steigende Fehler bei der Perturbations-Methode verursacht.
Zusammenfassend kann festgestellt werden, daß auch Beispiel 2 diesselben Schlußfolgerungen zuläßt wie Beispiel 1. Die Perturbations-Methode ist nur in einer kleinen Umgebung um das Gleichgewicht für die Approximation von u geeignet.

7 Fazit der praktischen Ergebnisse und Ausblick

An dieser Stelle soll das Ergebnis der praktischen Auswertungen und des Vergleichs der beiden Verfahren noch einmal zusammengefaßt und ein Ausblick auf Verbesserungsmöglichkeiten gegeben werden.

Bei der Auswertung der Methode von SCHMITT-GROHÉ und URIBE in Kapitel 5 zeigt sich, daß diese mit zunehmender Entfernung vom Gleichgewicht immer schlechtere Werte für die Approximation der Wertefunktion $\widehat{V}$ und der Kontrolle $\widehat{u}$ liefert. Bei der TAYLOR-Approximation von $\widehat{u}$ hat der erste Term $\bar{u}$ das meiste Gewicht. Dies bedeutet zwar einerseits hohe Fehler für große Umgebungen, liefert aber andererseits sehr genaue Näherungen in kleinen Umgebungen, was eine Stärke der Methode darstellt. Der hohe Approximationsfehler bei $\widehat{V}$ wird einerseits dadurch verursacht, daß man $\widehat{u}$ einsetzt und sich der Fehler hier fortpflanzt. Der Hauptgrund ist allerdings, daß für die Approximation in der Umgebung um das Gleichgewicht die Formel für die Wertefunktion für $t \to \infty$ benutzt wird. Diese gilt folglich nur im Gleichgwicht selbst und bildet einen ersten Ansatzpunkt zur Verbesserung der Approximation. Hier muß überlegt werden, wie die Wertefunktion in der Umgebung passender dargestellt werden kann. Approximationen höherer Ordnung sind an dieser Stelle kein Ausweg, da die Problemstellung sich nicht ändert.

Bei dem Vergleich der beiden Verfahren in Kapitel 6 zeigt sich, daß die Methode aus der dynamischen Programmierung konstante Approximationsfehler bei $\widehat{V}$ liefert, die unabhängig vom Gleichgewicht sind. Nachteil hier ist die teilweise recht lange Rechenzeit, da die Methode mehrere Iterationen durchläuft, die abhängig vom Beispiel und von der

Schrittweite sind. Auf direkte Vergleiche der Rechenzeit der beiden Methoden wurde verzichtet, da die Perturbations-Methode nur einen Schritt ausführt und das Ergebnis unmittelbar berechnet wird. Dieses Verfahren ist also oberflächlich betrachtet immer das schnellere von beiden. Allerdings muß hier das Aufstellen der Gleichgewichtsgleichung offline erfolgen bevor man die Routine starten kann. Folglich wird hier schon bei der Vorarbeit viel Zeit investiert, die schlecht in einem direkten Vergleich gemessen werden kann.
Diese Vorarbeit bildet einen weiteren Ansatz zur Verbesserung. Es kann überlegt werden, in wie weit dieser Schritt automatisiert werden kann. Man kann untersuchen, ob z.B. ein Programmpaket wie MAPLE die Optimalitätsbedingungen berechnen und die Gleichgewichtsgleichung so schneller aufgestellt werden kann.
Ein weiterer Nachteil der Perturbations-Methode ist es, daß das zu untersuchende Beispiel einen Gleichgewichtspunkt haben muß, damit das hier dargestellte Verfahren überhaupt angewendet werden kann. Dazu kommt noch, daß dieses Gleichgewicht vorher berechnet werden muß. Dieser Nachteil trat hier nicht direkt auf, weil die Beispiele natürlich so gewählt waren, daß sie ein Gleichgewicht hatten. Dessen Berechnung war außerdem vergleichsweise einfach, aber bei anderen Modellen kann der Aufwand dafür sehr hoch sein.

Zusammenfassend läßt sich die Frage, welches der Verfahren zur Approximation der Wertefunktion $\widehat{V}$ und der Kontrolle $\widehat{u}$ besser geeignet ist, eindeutig zu Gunsten der dynamischen Programmierung beantworten. Man kann sich darauf verlassen, daß sich der Fehler in einer bestimmten akzeptablen Größenordnung bewegt. Dafür muß die etwas längere Rechenzeit in Kauf genommen werden. Ein Nachteil dieser Methode ist, daß sie für Modelle höherer Dimension nicht anwendbar ist, da die Gittererzeugung zu kompliziert wird. Allerdings kann die dynamische Programmierung bei Beispielen mit niedriger Dimension viel flexibler eingesetzt werden und auf mehr Modelle angewendet werden, da sie nicht von einem bestimmten Punkt ausgeht und nicht die Kenntnis bestimmter Ableitungen voraussetzt.
Wenn man allerdings ausschließlich an einer Approximation der Kontrolle $\widehat{u}$ interessiert

ist, kann die Perturbations-Methode hier sehr gute Approximationen liefern - zumindest in kleinen Umgebungen. Wie groß diese Umgebungen sein dürfen, bietet Platz für weitere Untersuchungen in Hinblick darauf, welche Rolle hier z.B. die Standardabweichung spielt.

A Notation

An dieser Stelle werden die wichtigsten Variablenbezeichnungen und Notationen übersichtsartig aufgeführt. Die Reihenfolge entspricht dabei dem erstmaligen Auftreten im Text.

f	Gleichgewichtsgleichung
u_t	Kontrolle zum Zeitpunkt t in diskreter Zeit
x_t	Zustand zum Zeitpunkt t in diskreter Zeit
$\mathbf{U}$	Kontrollbereich
$\mathbf{T}$	Zeitachse in diskreter Zeit
h	Schrittweite
φ_h	Kontrollsystem in diskreter Zeit
Φ_h	Trajektorie in diskreter Zeit
$\bar{x}$	Zustand im Gleichgewicht
$\bar{u}$	Kontrolle im Gleichgewicht
J_h	diskontiertes Funktional in diskreter Zeit
β^t	Diskontrate in diskreter Zeit
$V_h(x)$	Optimale Wertefunktion in diskreter Zeit
H^t	Hamiltonfunktion in diskreter Zeit
λ	Kozustandsvektor
$u(t)$	Kontrolle in kontinuierlicher Zeit
$x(t)$	Zustand in kontinuierlicher Zeit
φ	Kontrollsystem in kontinuierlicher Zeit
Φ	Trajektorie in kontinuierlicher Zeit
J	diskontiertes Funktional in kontinuierlicher Zeit
$V(x)$	Optimale Wertefunktion in kontinuierlicher Zeit
$\widetilde{\varphi}_h$	mit Euler-Verfahren zeitlich diskretisiertes Kontrollsystem
$\widetilde{\Phi}_h$	Lösung eines mit Euler diskretisierten Kontrollsystems

$\widetilde{J}_h$	zur Euler-Diskretisierung gehöriges diskontiertes Funktional
$\widetilde{V}_h$	zur Euler-Diskretisierung gehörige optimale Wertefunktion
x^1	Zustandsvariable, die deterministischer Dynamik folgt
x^2	Zustandsvariable, die stochastischer Dynamik folgt
g	Kontrollfunktion im Modell
h	Funktion für Dynamik im Modell
$F(x, \sigma)$	Definierte Hilfsfunktion, um Terme für die Approximation zu berechnen
$\widetilde{f}$	äquivalente Form zu f
$\widehat{u}$	Approximation der Kontrolle u
$\widehat{x}_1$	Approximation der Zustandsvariablen x_1
$\widehat{x}_2$	Approximation der Zustandsvariablen x_2
$\widehat{V}$	Approximation der optimalen Wertefunktion
Ω	das untersuchte Intervall
v_h^i	Optimale Wertefunktion im Laufe des Iterationsverfahrens nach der i-ten Iteration
T_h	Operator zur Berechnung der v_h^i
$\mathcal{W}$	Raum der affin bilinearen Funktionen
$\widehat{v}_h^i$	An den Eckpunkten des Gitters approximierte Wertefunktion nach der i-ten Iteration aus $\mathcal{W}$
V^j	Vektor der Werte von $\widehat{v}_h^j$ an den Knoten

B MATLAB-Quelltexte

B.1 Die Routine Beispiel.m

```
%BEISPIEL.M function
[fx,fxp,fy,fyp,fypyp,fypy,fypxp,fypx,fyyp,fyy,
fyxp,fyx,fxpyp,fxpy,fxpxp,fxpx,fxyp,fxy,fxxp,fxx,f]
= beispiel
% Dieses Programm berechnet die analytischen Ableitungen.
% Es werden keine Inputs benötigt.
% Output: Analytische erste und zweite Ableitung der
% Funktion f.
% Ruft : anal_deriv.m


% Parameter definieren
syms SIG ALFA Betta RHO A eta

% Variablen definieren
syms x1 x1p x2 x2p u up  % Das 'p' steht für prime
und bedeutet Periode (t+1)

% Gleichungen für Funktion f definieren, i=1:3
f1 = x1p - A*exp(x2) * x1^(ALFA) + u ;
f2 = Betta/up* ALFA * A *
exp(x2p) * x1p^(ALFA -1) - 1/u;
f3 = x2p - RHO*x2 ;
```

```
% Funktion f definieren
f = [f1;f2;f3];

% Zustands- und Kontrollvektoren definieren.
x = [x1 x2]; y = u; xp = [x1p x2p]; yp = up;

% anal_deriv.m aufrufen
[fx,fxp,fy,fyp,fypyp,fypy,fypxp,fypx,fyyp,fyy,fyxp,
fyx,fxpyp,fxpy,fxpxp,fxpx,fxyp,fxy,fxxp,fxx]
=anal_deriv(f,x,y,xp,yp);
```

B.2 Die Routine Beispiel_Zahlen.m

```
%Beispiel_Zahlen.M function
[SIG,ALFA,Betta,RHO,eta,A,x1,x2,u,x1p,x2p,up]
=beispiel_zahlen
% In diesem Programm müssen in die Werte aller
% Parameter eingegeben werden,
% sowie die Gleichgewichtswerte für u, x1 und x2.
% Ausserdem wird festgelegt, dass im GGW gilt
% x1(t+1)=x1(t), x2(t+1)=x2(t)
% und u(t+1)=u(t).

A=5; ALFA=0.34; Betta=0.95; RHO=0.9; SIG=0.008;
eta=[0 1]';

x2 = 0; % Gleichgewichtswert der
'stochastischen' Variablen
x1 = 2.067344815; % Gleichgewichtswert der
```

```
'deterministischen' Variablen
u = 4.333103529; %steady-state value of control

x2p = x2;
x1p = x1;
up = u;
```

B.3 Die Routine Beispiel_Run.m

```
%BEISPIEL_RUN.M
% Dieses Programm berechnet die quadratische Approximation
% für das
% Modell-Beispiel. Approximiert werden die Dynamik x
% und die Kontrolle u. Sie sind von der Form
% xp = h(x,sigma) + sigma* eta * ep
% u = g(x,sigma)
% Notation: x is x_t and xp is x_t+1
%
% hx ist nx* nx
% gx ist ny * nx
% hxx ist nx * nx * nx
% gxx ist ny * nx * nx
% eta ist nx * ne
% gss ist ny * 1
% hss ist nx * 1
% sigma ist eine positive Zahl
% Ruft: beispiel.m, num_eval.m, beispiel_zahlen.m,
% gx_hx.m, gxx_hxx.m, gss_hss.m

% Mit Beispiel.m werden die analytischen
```

```
% Ableitungen berechnet
[fx,fxp,fy,fyp,fypyp,fypy,fypxp,fypx,fyyp,
fyy,fyxp,fyx,fxpyp,fxpy,fxpxp,fxpx,fxyp,fxy,fxxp,fxx,f]
= beispiel;

% Mit Beispiel_zahlen.m werden
%alle Parameter mit Werten belegt
[SIG,ALFA,Betta,RHO,eta,A,x1,x2,u,x1p,x2p,up]
=beispiel_zahlen;

% Ordnung der Approximation
approx = 2;

% Aufruf von num_eval.m um die numerischen Werte
%zu bekommen.
num_eval

% Lineare Approximation
[gx,hx] = gxhx(nfy,nfx,nfyp,nfxp)

% Quadratische Approximation
[gxx,hxx] = gxx_hxx(nfx,nfxp,nfy,nfyp,nfypyp,
nfypy,nfypxp,nfypx,nfyyp,nfyy,nfyxp,nfyx,nfxpyp,
nfxpy,nfxpxp,nfxpx, nfxyp,nfxy,nfxxp,nfxx,hx,gx)

[gss,hss] = gss_hss(nfx,nfxp,nfy,nfyp,nfypyp,nfypy,
nfypxp,nfypx,nfyyp,nfyy,nfyxp,nfyx,nfxpyp,nfxpy,nfxpxp,
nfxpx,nfxyp,nfxy,nfxxp,nfxx,hx,gx,gxx,eta)
```

B.4 Die Routine Auswertung1.m

```
clear all;

% Parameter definieren
beta=0.95; alpha=0.34; rho=0.9; A=5; sigma=0.008;

%Festlegen des zu untersuchenden Intervalls
x1anfang=1; x1ende=4; x2anfang=-0.3; x2ende=0.3;
schrittweitex1=0.075; schrittweitex2=0.015;

laengex1=(x1ende-x1anfang)/schrittweitex1 + 1;
laengex2=(x2ende-x2anfang)/schrittweitex2 + 1;

%Belegen von x mit Werten

for i=1:laengex1
    x1(i)=x1anfang+(i-1)*schrittweitex1;
end

for i=1:laengex2
    x2(i)=x2anfang+(i-1)*schrittweitex2;
end

%Einzelteile für Formel von exaktem V(x)definieren:
%V(x)=B+C*ln(x1)+D*x2
B=
(log((1-beta*alpha)*A)+((beta*alpha)/
(1-beta*alpha))*log(beta*alpha*A))/(1-beta);
```

```
C=alpha/(1-alpha*beta);

D=1/((1-alpha*beta)*(1-rho*beta));

%Belegen der Matrix V_exakt mit den Werten aus der Formel
for i=1:laengex1
    for j=1:laengex2

        V_exakt(i,j)=B +C*log(x1(i)) +D*x2(j);

    end
end

%Festlegen der Werte für Taylor-Approx.

x1_ggw=2.067344815; x2_ggw=0; u_ggw=4.333103529; gx=[0.7126
4.3331]; gxx1=[-0.2275 0.7126]; gxx2=[0.7126 4.3331]; gss=0;

%Taylor-Approximation
for i=1:laengex1
    for j=1:laengex2
        T(i,j)=u_ggw
        +gx(1)*(x1(i)-x1_ggw)
        +gx(2)*(x2(j)-x2_ggw)
        +0.5*(gxx1(1)*(x1(i)-x1_ggw)*(x1(i)-x1_ggw)
        + gxx1(2)*(x2(j)-x2_ggw)*(x1(i)-x1_ggw))
        +0.5*(gxx2(1)*(x1(i)-x1_ggw)*(x2(j)-x2_ggw)
        +gxx2(2)*(x2(j)-x2_ggw)*(x2(j)-x2_ggw))
        +0.5*(gss*sigma*sigma);
    end
```

```
 end

%Berechnung von V_approx:

for i=1:laengex1
    for j=1:laengex2

    V_approx(i,j)=log(T(i,j))/(1-beta);
    end
end

%Berechnung des Approximationsfehlers

for i=1:laengex1
    for j=1:laengex2
        Fehler(i,j)=V_exakt(i,j)-V_approx(i,j);
    end
end

for i=1:laengex1
    for j=1:laengex2
        Fehlerabs(i,j)=abs(Fehler(i,j));
    end
end

%Berechnung des Fehlers bei der Kontrolle u.
%Für u_exakt gibt es eine
%Formel:(1-alpha*beta)*A*exp(x2)*x1^alpha.

for i=1:laengex1
```

```
    for j=1:laengex2
        U_exakt(i,j)=(1-alpha*beta)*A*exp(x2(j))
        *x1(i)^alpha;
        U_Fehler(i,j)=abs(T(i,j)-U_exakt(i,j));
    end
end



% Plotten der Fehler zwischen exakter und
%approx. Value-Function: figure
surfc(x2anfang:schrittweitex2:x2ende,
x1anfang:schrittweitex1:x1ende,Fehlerabs)
xlabel('x2- Werte');
ylabel('x1- Werte');
title('\bf{Approximationsfehler}') colorbar

% Plotten der beiden value-functions:
figure
surf(x2anfang:schrittweitex2:x2ende,
x1anfang:schrittweitex1:x1ende,
V_exakt)

xlabel('x2- Werte');
ylabel('x1- Werte');
title('\bf{Vergleich}')

hold on
surf(x2anfang:schrittweitex2:x2ende,
x1anfang:schrittweitex1:x1ende,
```

```
V_approx)

% Plotten der Fehler zwischen exaktem und approx. u:
figure
surfc(x2anfang:schrittweitex2:x2ende,
x1anfang:schrittweitex1:x1ende,
U_Fehler)

xlabel('x2- Werte'); ylabel('x1- Werte');
title('\bf{Fehler bei
der Kontrolle}')
colorbar
```

B.5 Die Routinen Auswertung2a.m und Auswertung2b.m

Für die Berechnung der Wertefunktion $\widehat{V}$:

```
clear all;

% Parameter definieren
beta=0.95; alpha=0.34; rho=0.9; A=5; sigma=0.008;

% Die gewünschten Daten über die Wertefunktion einlesen
Value=load('Intervall2.txt');

% Ale benötigten Parameter eingeben
x1anfang=1; x1ende=4; x2anfang=-0.3; x2ende=0.3;
schrittweitex1=0.075; schrittweitex2=0.015;
laengex1=(x1ende-x1anfang)/schrittweitex1 + 1;
laengex2=(x2ende-x2anfang)/schrittweitex2 + 1;
```

```
% Die Daten in eine passende Matrix übertragen,
% es wird nur die dritte Spalte benötigt.
for i=1:laengex1
    for j=1:laengex2
        Value_Matrix(i,j)=Value((i-1)*laengex1 + j, 3);
    end
end

%Belegen von x mit Werten

for i=1:laengex1
    x1(i)=x1anfang+(i-1)*schrittweitex1;
end

for i=1:laengex2
    x2(i)=x2anfang+(i-1)*schrittweitex2;
end

% Mit exakter Formel für V vergleichen
% Einzelteile für Formel von exaktem V(x)
% definieren:V(x)=B+C*ln(x1)+D*x2

B=
(log((1-beta*alpha)*A)+((beta*alpha)/
(1-beta*alpha))*log(beta*alpha*A))/(1-beta);

C=alpha/(1-alpha*beta);

D=1/((1-alpha*beta)*(1-rho*beta));
```

```
% Belegen der Matrix V mit den Werten aus der Formel
for i=1:laengex1
    for j=1:laengex2

        V(i,j)=B +C*log(x1(i)) +D*x2(j);

    end
end

figure surf(x2anfang:schrittweitex2:x2ende,
x1anfang:schrittweitex1:x1ende,Value_Matrix)
xlabel('x2- Werte');
ylabel('x1- Werte');
title('\bf{Vergleich}')
hold on

surf(x2anfang:schrittweitex2:x2ende,
x1anfang:schrittweitex1:x1ende,V)

%Berechnung des Fehlers mit der Matrix F
for i=1:laengex1
    for j=1:laengex2
        F(i,j)=V(i,j)-Value_Matrix(i,j);
    end
end
% Berechnung des absoluten Fehlers
for i=1:laengex1
    for j=1:laengex2
        F_abs(i,j)=abs(F(i,j));
```

```
    end
end

figure surf(x2anfang:schrittweitex2:x2ende,
x1anfang:schrittweitex1:x1ende,F_abs)

xlabel('x2- Werte');
ylabel('x1- Werte');
title('\bf{Approximationsfehler}')
colorbar
```

Für die Berechnung der Kontrollen $\widehat{u}$:

```
clear all;
% Für die Auswertung der Bellmann-Methode
% Parameter definieren
beta=0.95; alpha=0.34; rho=0.9; A=5; sigma=0.008;


% Die gewünschten Daten über die Kontrolle einlesen
UWerte=load('uzahlenggw.txt');

% Alle benötigten Parameter eingeben
x1anfang=2.066; x1ende=2.068; x2anfang=-0.001; x2ende=0.001;
schrittweitex1=0.00002;
schrittweitex2=0.00002;
laengex1=(x1ende-x1anfang)/schrittweitex1 + 1;
laengex2=(x2ende-x2anfang)/schrittweitex2 + 1;


% Die Daten in eine passende Matrix übertragen
```

```
for i=1:laengex1
    for j=1:laengex2
        U_Matrix(i,j)=UWerte((i-1)*laengex1 + j, 3);
    end
end

%Belegen von x mit Werten

for i=1:laengex1
    x1(i)=x1anfang+(i-1)*schrittweitex1;
end

for i=1:laengex2
    x2(i)=x2anfang+(i-1)*schrittweitex2;
end

%Berechnung des exakten u mit Formel
%Formel:(1-alpha*beta)*A*exp(x2)*x1^alpha.

for i=1:laengex1
    for j=1:laengex2
        U_exakt(i,j)=(1-alpha*beta)*A*exp(x2(j))*x1(i)^alpha;
        U_Fehler(i,j)=abs(U_Matrix(i,j)-U_exakt(i,j));
    end
end


figure surf(x2anfang:schrittweitex2:x2ende,
x1anfang:schrittweitex1:x1ende,U_Matrix)
xlabel('x2- Werte');
```

```
ylabel('x1- Werte');
title('\bf{Vergleich}')
hold on
surf(x2anfang:schrittweitex2:x2ende,
x1anfang:schrittweitex1:x1ende,U_exakt)


figure surf(x2anfang:schrittweitex2:x2ende,
x1anfang:schrittweitex1:x1ende,U_Fehler)

xlabel('x2- Werte');
ylabel('x1- Werte');
title('\bf{Approximationsfehler bei der Kontrolle u}')
colorbar
```

C Material auf der beiliegenden CD

Die beiliegende CD enthält alle benutzten und angesprochenen MATLAB-Dateien im Verzeichnis MATLAB, alle in der Arbeit verwendeten Abbildungen im Verzeichnis Bilder, alle zur Auswertung verwendeten Datensätze im Verzeichnis Datensaetze, sowie die komplette Arbeit als pdf-Datei und als Latex-Datei im Verzeichnis Diplomarbeit.

Das Verzeichnis Matlab

Es sind die folgenden Dateien gespeichert

- `anal_deriv.m`, `num_eval.m`, `gxhx.m`, `gxx_hxx.m` und `gss_hss.m` für die Methode von SCHMITT-GROHÉ und URIBE.
- `Beispiel.m`, `Beispiel_Zahlen.m`, `Beispiel_run.m` für die Implementierung von Beispiel 1 und `Beispiel2.m`, `Beispiel2_Zahlen.m` und `Beispiel2_run.m` für die Implementierung von Beispiel 2.
- `Auswertung1.m`, `Auswertung2a.m` und `Auswertung2b.m` für die praktischen Auswertungen und zur Erzeugung der Abbildungen.

Das Verzeichnis Bilder

Um die Abbildungen besser zuordnen zu können, haben die Datei-Namen die gleiche Bezeichnung wie die entsprechenden Abbildungsnummern im Text.

Das Verzeichnis Datensaetze

Die benutzten Datensätze werden in der folgenden Tabelle übersichtartig aufgeführt und ihre Verwendung beschrieben.

Name der Datensätze	**Verwendung**
v_zahlen1.txt - v_zahlen8.txt	Tabelle 6.2, Auswertung mit Verfahren 2 von $\widehat{V}$
u_zahlen1.txt - u_zahlen8.txt	Tabelle 6.2, Auswertung mit Verfahren 2 von $\widehat{u}$
intervall1.txt	Abbildung 6.1
intervall1a.txt	Abbildung 6.2
intervall2.txt	Abbildung 6.3
intervall2a.txt	Abbildung 6.4
kappa1.txt - kappa4.txt	Abbildung 6.5
kappa2.txt	Abbildung 6.6
u_zahlen_kappa2.txt	Abbildung 6.10

Literaturverzeichnis

[1] K. J. ÅSTRÖM, *Introduction to Stochastic Control Theory*, in R. Bellmann, Hrsg., *Mathematics in Science and Engineering, Volume 70*, Academic Press, 1970.

[2] R. BELLMANN, *Introduction to the Mathematical Theory of Control Processes, Volume I*, in R. Bellmann, Hrsg., *Mathematics in Science and Engineering, Volume 40*, Academic Press, 1967.

[3] C. BÜSKENS, *Diskretisierungsverfahren zur numerischen Lösung optimaler Steuerprozesse*, Ausarbeitung einer Vorlesung gehalten an der Universität Bayreuth im Wintersemester 2002/2003.

[4] C. BÜSKENS, *Optimale Steuerung und Regelung*, Ausarbeitung einer Vorlesung gehalten an der Universität Bayreuth im Wintersemester 2003/2004.

[5] M. CAPIŃSKI UND E. KOPP, *Measure, Integral and Probability*, 2. Auflage, Springer-Verlag, 2005.

[6] J. ELSTRODT, *Maß- und Integrationstheorie*, 3. Auflage, Springer-Verlag, 2002.

[7] G. FEICHTINGER UND R. F. HARTL, *Optimale Kontrolle ökonomischer Prozesse*, de Gruyter, 1986.

[8] G. FISCHER, *Lineare Algebra*, Vieweg Verlag, 1997.

[9] L. GRÜNE, *Numerik dynamischer Systeme*, Ausarbeitung einer Vorlesung gehalten an der Universität Bayreuth im Wintersemester 2003/2004.

[10] L. GRÜNE, *Numerische Dynamik von Kontrollsystemen*, Ausarbeitung einer Vorlesung gehalten an der Universität Bayreuth im Sommersemester 2004.

[11] L. GRÜNE, *Error estimation and adaptive discretization for the discrete stochastic Hamilton-Jacobi-Bellmann equation* , Numerische Mathematik 99, pp.85-112, Springer-Verlag 2004.

[12] L. GRÜNE UND W. SEMMLER, *Using dynamic programming with adaptive grid scheme for optimal control problems in economics*, Journal of Economic Dynamics and Control 28, pp. 2427-2456, 2004.

[13] H. JIN UND K. L. JUDD, *Perturbation methods for general dynamic stochastic models*, Working Paper, Stanford University, 2002, `http://bucky.stanford.edu/defaultmain.htm`, Stand April 2005.

[14] K. L. JUDD, *Approximation, perturbation, and projection methods in economic analyis*, in H. Amman, D. Kendrick, und J. Rust, Hrsg., *Handbook of Computational Economics*, Amsterdam: Elsevier, 1996.

[15] K. L. JUDD, *Numerical Methods in Economics*, The MIT Press, 1998.

[16] F. E. KYDLAND UND E. C. PRESCOTT, *Time to build and aggregate fluctations*, Econometrica 50, pp. 1345-1370, 1982.

[17] A. MAS-COLELL, *The Theory of General Economic Equilibrium - A Differentiable Approach*, Econometric Society Monograph, Cambridge University Press, 1989.

[18] M. MORLOCK UND K. NEUMANN, *Operations Research*, 2. Auflage, Hanser Verlag, 2002.

[19] J. A. MURDOCK, *Perturbations, Theory and Methods*, John Wiley & Sons, 1991.

[20] R. NECK, *Stochastic Control Theory and Operational Research*, European Journal of Operational Research 17, pp. 283-301, 1984.

[21] P. OBERENDER, *Grundbegriffe der Mikroökonomie*, Verlag P.C.O., Bayreuth, 1999.

[22] F. VAN DER PLOEG, Hrsg., *Mathematical Methods in Economics*, John Wiley & Sons, 1986.

[23] S. SCHMITT-GROHÉ UND M. URIBE, *Solving dynamic general equilibrium models using a second-order approximation to the policy function*, Journal of Economic Dynamics and Control 28, pp. 755-775, 2004.

[24] S. SCHMITT-GROHÉ, *Perturbation Methods for the Numerical Analysis of DSGE Models*, Lecture Notes, preliminary draft, Duke University, 2005, `www.econ.duke.edu/~grohe/teaching/perturbationmethods.htm`, Stand April 2005.

[25] S. SCHMITT-GROHÉ UND M. URIBE, *Matlab Codes for Second Order Approximation*, `http://www.econ.duke.edu/~uribe/2nd_order.htm`, Stand April 2005.

Erklärung

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

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

Bayreuth, den 05. Oktober 2005 .....

Stephanie Becker